GDCM

3.0.24

Generated by Doxygen 1.15.0

1 GDCM Documentation 1

2 Todo List 3

3 Deprecated List 5

4 Bug List 7

5 Directory Hierarchy 9

5.1 Directories . . . 9

6 Namespace Index 23

6.1 Namespace List . . . 23

7 Hierarchical Index 25

7.1 Class Hierarchy . . . 25

8 Class Index 35

8.1 Class List . . . 35

9 File Index 49

9.1 File List . . . 49

10 Directory Documentation 57

10.1 Common Directory Reference . . . 57

10.2 DataDictionary Directory Reference . . . 59

10.3 DataStructureAndEncodingDefinition Directory Reference . . . 60

10.4 InformationObjectDefinition Directory Reference . . . 61

10.5 MediaStorageAndFileFormat Directory Reference . . . 62

10.6 MessageExchangeDefinition Directory Reference . . . 65

10.7 Python Directory Reference . . . 67

10.8 Source Directory Reference . . . 67

10.9 Utilities Directory Reference . . . 68

10.10 VTK Directory Reference . . . 68

10.11 Wrapping Directory Reference . . . 69

11 Namespace Documentation 71

11.1 gdcm Namespace Reference . . . 71

11.1.1 Detailed Description . . . 86

11.1.2 Typedef Documentation . . . 86

11.1.2.1 AEComp . . . 86

11.1.2.2 ASComp . . . 86

11.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER . . . 86

11.1.2.4 CSComp . . . 86
11.1.2.5 DAComp . . . 87
11.1.2.6 DTComp . . . 87
11.1.2.7 FileList . . . 87
11.1.2.8 IconImage . . . 87
11.1.2.9 LOComp . . . 87
11.1.2.10 LTComp . . . 87
11.1.2.11 MacroEntry . . . 87
11.1.2.12 NestedMacroEntries . . . 87
11.1.2.13 PNComp . . . 88
11.1.2.14 SHComp . . . 88
11.1.2.15 STComp . . . 88
11.1.2.16 TMComp . . . 88
11.1.2.17 UCComp . . . 88
11.1.2.18 UIComp . . . 88
11.1.2.19 URComp . . . 88
11.1.2.20 UTComp . . . 88
11.1.3 Enumeration Type Documentation . . . 89
11.1.3.1 CompOperators . . . 89
11.1.3.2 ECharSet . . . 89
11.1.3.3 ENQueryType . . . 90
11.1.3.4 EQueryLevel . . . 90
11.1.3.5 EQueryType . . . 90
11.1.3.6 ERootType . . . 90
11.1.3.7 LodModeType . . . 91
11.1.4 Function Documentation . . . 91
11.1.4.1 add1() . . . 91
11.1.4.2 backslash() . . . 91
11.1.4.3 Clamp() . . . 91
11.1.4.4 clean() . . . 91
11.1.4.5 doround() . . . 92
11.1.4.6 GetVRFromTag() . . . 92
11.1.4.7 operator"!=() [1/2] . . . 92
11.1.4.8 operator"!=() [2/2] . . . 92
11.1.4.9 operator<<() [1/59] . . . 92
11.1.4.10 operator<<() [2/59] . . . 92
11.1.4.11 operator<<() [3/59] . . . 93
11.1.4.12 operator<<() [4/59] . . . 93
11.1.4.13 operator<<() [5/59] . . . 93

11.1.4.14 operator<<() [6/59] 93
11.1.4.15 operator<<() [7/59] 93
11.1.4.16 operator<<() [8/59] 93
11.1.4.17 operator<<() [9/59] 93
11.1.4.18 operator<<() [10/59] 94
11.1.4.19 operator<<() [11/59] 94
11.1.4.20 operator<<() [12/59] 94
11.1.4.21 operator<<() [13/59] 94
11.1.4.22 operator<<() [14/59] 94
11.1.4.23 operator<<() [15/59] 94
11.1.4.24 operator<<() [16/59] 94
11.1.4.25 operator<<() [17/59] 95
11.1.4.26 operator<<() [18/59] 95
11.1.4.27 operator<<() [19/59] 95
11.1.4.28 operator<<() [20/59] 95
11.1.4.29 operator<<() [21/59] 95
11.1.4.30 operator<<() [22/59] 95
11.1.4.31 operator<<() [23/59] 95
11.1.4.32 operator<<() [24/59] 96
11.1.4.33 operator<<() [25/59] 96
11.1.4.34 operator<<() [26/59] 96
11.1.4.35 operator<<() [27/59] 96
11.1.4.36 operator<<() [28/59] 96
11.1.4.37 operator<<() [29/59] 96
11.1.4.38 operator<<() [30/59] 96
11.1.4.39 operator<<() [31/59] 97
11.1.4.40 operator<<() [32/59] 97
11.1.4.41 operator<<() [33/59] 97
11.1.4.42 operator<<() [34/59] 97
11.1.4.43 operator<<() [35/59] 97
11.1.4.44 operator<<() [36/59] 97
11.1.4.45 operator<<() [37/59] 97
11.1.4.46 operator<<() [38/59] 98
11.1.4.47 operator<<() [39/59] 98
11.1.4.48 operator<<() [40/59] 98
11.1.4.49 operator<<() [41/59] 98
11.1.4.50 operator<<() [42/59] 98
11.1.4.51 operator<<() [43/59] 98
11.1.4.52 operator<<() [44/59] 98

11.1.4.53 operator<<() [45/59] . . . 99
11.1.4.54 operator<<() [46/59] . . . 99
11.1.4.55 operator<<() [47/59] . . . 99
11.1.4.56 operator<<() [48/59] . . . 99
11.1.4.57 operator<<() [49/59] . . . 99
11.1.4.58 operator<<() [50/59] . . . 99
11.1.4.59 operator<<() [51/59] . . . 99
11.1.4.60 operator<<() [52/59] . . . 100
11.1.4.61 operator<<() [53/59] . . . 100
11.1.4.62 operator<<() [54/59] . . . 100
11.1.4.63 operator<<() [55/59] . . . 100
11.1.4.64 operator<<() [56/59] . . . 100
11.1.4.65 operator<<() [57/59] . . . 100
11.1.4.66 operator<<() [58/59] . . . 100
11.1.4.67 operator<<() [59/59] . . . 101
11.1.4.68 operator==() . . . 101
11.1.4.69 operator>>() [1/3] . . . 101
11.1.4.70 operator>>() [2/3] . . . 101
11.1.4.71 operator>>() [3/3] . . . 101
11.1.4.72 Round() . . . 101
11.1.4.73 roundat() . . . 102
11.1.4.74 x16printf() . . . 102
11.1.5 Variable Documentation . . . 102
11.1.5.1 GlobalInstance . . . 102
11.2 gdcm::network Namespace Reference . . . 102
11.2.1 Enumeration Type Documentation . . . 107
11.2.1.1 EEventID . . . 107
11.2.1.2 EStateID . . . 107
11.2.2 Function Documentation . . . 108
11.2.2.1 GetStateIndex() . . . 108
11.2.3 Variable Documentation . . . 108
11.2.3.1 cMaxEventID . . . 108
11.2.3.2 cMaxStateID . . . 108
11.3 gdcm::SegmentHelper Namespace Reference . . . 109
11.4 gdcm::terminal Namespace Reference . . . 109
11.4.1 Detailed Description . . . 109
11.4.2 Enumeration Type Documentation . . . 110
11.4.2.1 Attribute . . . 110
11.4.2.2 Color . . . 110

11.4.2.3 Mode . . . 110
11.4.3 Function Documentation . . . 111
11.4.3.1 setattribute() . . . 111
11.4.3.2 setbgcolor() . . . 111
11.4.3.3 setfgcolor() . . . 111
11.4.3.4 setmode() . . . 111

12 Class Documentation **113**
12.1 gdcm::network::AAbortPDU Class Reference . . . 113
12.1.1 Detailed Description . . . 114
12.1.2 Constructor & Destructor Documentation . . . 114
12.1.2.1 AAbortPDU() . . . 114
12.1.3 Member Function Documentation . . . 114
12.1.3.1 IsLastFragment() . . . 114
12.1.3.2 Print() . . . 114
12.1.3.3 Read() . . . 115
12.1.3.4 SetReason() . . . 115
12.1.3.5 SetSource() . . . 115
12.1.3.6 Size() . . . 115
12.1.3.7 Write() . . . 115
12.2 gdcm::network::AAssociateACPDU Class Reference . . . 116
12.2.1 Detailed Description . . . 117
12.2.2 Member Typedef Documentation . . . 117
12.2.2.1 SizeType . . . 117
12.2.3 Constructor & Destructor Documentation . . . 117
12.2.3.1 AAssociateACPDU() . . . 117
12.2.4 Member Function Documentation . . . 118
12.2.4.1 AddPresentationContextAC() . . . 118
12.2.4.2 GetNumberOfPresentationContextAC() . . . 118
12.2.4.3 GetPresentationContextAC() . . . 118
12.2.4.4 GetUserInformation() . . . 118
12.2.4.5 InitFromRQ() . . . 118
12.2.4.6 IsLastFragment() . . . 118
12.2.4.7 Print() . . . 118
12.2.4.8 Read() . . . 119
12.2.4.9 SetCalledAETitle() . . . 119
12.2.4.10 SetCallingAETitle() . . . 119
12.2.4.11 Size() . . . 119
12.2.4.12 Write() . . . 119

12.2.5 Friends And Related Symbol Documentation . . . 119
12.2.5.1 AAssociateRQPDU . . . 119
12.3 gdcm::network::AAssociateRJPDU Class Reference . . . 120
12.3.1 Detailed Description . . . 121
12.3.2 Constructor & Destructor Documentation . . . 121
12.3.2.1 AAssociateRJPDU() . . . 121
12.3.3 Member Function Documentation . . . 121
12.3.3.1 IsLastFragment() . . . 121
12.3.3.2 Print() . . . 121
12.3.3.3 Read() . . . 121
12.3.3.4 Size() . . . 121
12.3.3.5 Write() . . . 122
12.4 gdcm::network::AAssociateRQPDU Class Reference . . . 122
12.4.1 Detailed Description . . . 124
12.4.2 Member Typedef Documentation . . . 124
12.4.2.1 PresentationContextArrayType . . . 124
12.4.2.2 SizeType . . . 124
12.4.3 Constructor & Destructor Documentation . . . 124
12.4.3.1 AAssociateRQPDU() [1/2] . . . 124
12.4.3.2 AAssociateRQPDU() [2/2] . . . 124
12.4.4 Member Function Documentation . . . 124
12.4.4.1 AddPresentationContext() . . . 124
12.4.4.2 GetCalledAETitle() . . . 124
12.4.4.3 GetCallingAETitle() . . . 125
12.4.4.4 GetNumberOfPresentationContext() . . . 125
12.4.4.5 GetPresentationContext() . . . 125
12.4.4.6 GetPresentationContextByAbstractSyntax() . . . 125
12.4.4.7 GetPresentationContextByID() . . . 125
12.4.4.8 GetPresentationContexts() . . . 125
12.4.4.9 GetReserved43_74() . . . 125
12.4.4.10 GetUserInformation() . . . 125
12.4.4.11 IsAETitleValid() . . . 126
12.4.4.12 IsLastFragment() . . . 126
12.4.4.13 Print() . . . 126
12.4.4.14 Read() . . . 126
12.4.4.15 SetCalledAETitle() . . . 126
12.4.4.16 SetCallingAETitle() . . . 126
12.4.4.17 SetUserInformation() . . . 127
12.4.4.18 Size() . . . 127

12.4.4.19 Write() . . . 127
12.4.5 Friends And Related Symbol Documentation . . . 127
12.4.5.1 AAssociateACPDU . . . 127
12.5 gdcm::AbortEvent Class Reference . . . 128
12.6 gdcm::network::AbstractSyntax Class Reference . . . 129
12.6.1 Detailed Description . . . 129
12.6.2 Constructor & Destructor Documentation . . . 129
12.6.2.1 AbstractSyntax() . . . 129
12.6.3 Member Function Documentation . . . 130
12.6.3.1 GetAsDataElement() . . . 130
12.6.3.2 GetName() . . . 130
12.6.3.3 operator==() . . . 130
12.6.3.4 Print() . . . 130
12.6.3.5 Read() . . . 130
12.6.3.6 SetName() . . . 130
12.6.3.7 SetNameFromUID() . . . 130
12.6.3.8 Size() . . . 130
12.6.3.9 Write() . . . 131
12.7 gdcm::AnonymizeEvent Class Reference . . . 131
12.7.1 Detailed Description . . . 133
12.7.2 Member Typedef Documentation . . . 133
12.7.2.1 Self . . . 133
12.7.2.2 Superclass . . . 133
12.7.3 Constructor & Destructor Documentation . . . 133
12.7.3.1 AnonymizeEvent() [1/2] . . . 133
12.7.3.2 ∼AnonymizeEvent() . . . 133
12.7.3.3 AnonymizeEvent() [2/2] . . . 133
12.7.4 Member Function Documentation . . . 134
12.7.4.1 CheckEvent() . . . 134
12.7.4.2 GetEventName() . . . 134
12.7.4.3 GetTag() . . . 134
12.7.4.4 MakeObject() . . . 134
12.7.4.5 operator=() . . . 134
12.7.4.6 SetTag() . . . 134
12.8 gdcm::Anonymizer Class Reference . . . 135
12.8.1 Detailed Description . . . 137
12.8.2 Constructor & Destructor Documentation . . . 138
12.8.2.1 Anonymizer() . . . 138
12.8.2.2 ∼Anonymizer() . . . 139

12.8.3 Member Function Documentation . . . 139
12.8.3.1 BALCPProtect() . . . 139
12.8.3.2 BasicApplicationLevelConfidentialityProfile() . . . 139
12.8.3.3 CanEmptyTag() . . . 139
12.8.3.4 Clear() [1/2] . . . 139
12.8.3.5 Clear() [2/2] . . . 139
12.8.3.6 ClearInternalUIDs() . . . 140
12.8.3.7 Empty() [1/2] . . . 140
12.8.3.8 Empty() [2/2] . . . 140
12.8.3.9 GetBasicApplicationLevelConfidentialityProfileAttributes() . . . 140
12.8.3.10 GetCryptographicMessageSyntax() . . . 140
12.8.3.11 GetFile() . . . 141
12.8.3.12 New() . . . 141
12.8.3.13 RecurseDataSet() . . . 141
12.8.3.14 Remove() [1/2] . . . 141
12.8.3.15 Remove() [2/2] . . . 141
12.8.3.16 RemoveGroupLength() . . . 141
12.8.3.17 RemovePrivateTags() . . . 142
12.8.3.18 RemoveRetired() . . . 142
12.8.3.19 Replace() [1/4] . . . 142
12.8.3.20 Replace() [2/4] . . . 142
12.8.3.21 Replace() [3/4] . . . 142
12.8.3.22 Replace() [4/4] . . . 143
12.8.3.23 SetCryptographicMessageSyntax() . . . 143
12.8.3.24 SetFile() . . . 143
12.9 gdcm::AnyEvent Class Reference . . . 144
12.10 gdcm::network::ApplicationContext Class Reference . . . 145
12.10.1 Detailed Description . . . 146
12.10.2 Constructor & Destructor Documentation . . . 146
12.10.2.1 ApplicationContext() . . . 146
12.10.3 Member Function Documentation . . . 146
12.10.3.1 GetName() . . . 146
12.10.3.2 Print() . . . 146
12.10.3.3 Read() . . . 146
12.10.3.4 SetName() . . . 146
12.10.3.5 Size() . . . 146
12.10.3.6 Write() . . . 147
12.11 gdcm::ApplicationEntity Class Reference . . . 147
12.11.1 Detailed Description . . . 148

12.11.2 Member Function Documentation . . . 148
12.11.2.1 IsValid() . . . 148
12.11.2.2 Print() . . . 148
12.11.2.3 SetBlob() . . . 148
12.11.2.4 Squeeze() . . . 148
12.11.3 Member Data Documentation . . . 149
12.11.3.1 Internal . . . 149
12.11.3.2 MaxLength . . . 149
12.11.3.3 MaxNumberOfComponents . . . 149
12.11.3.4 Padding . . . 149
12.11.3.5 Separator . . . 149
12.12 gdcm::network::AReleaseRPPDU Class Reference . . . 149
12.12.1 Detailed Description . . . 150
12.12.2 Constructor & Destructor Documentation . . . 150
12.12.2.1 AReleaseRPPDU() . . . 150
12.12.3 Member Function Documentation . . . 150
12.12.3.1 IsLastFragment() . . . 150
12.12.3.2 Print() . . . 151
12.12.3.3 Read() . . . 151
12.12.3.4 Size() . . . 151
12.12.3.5 Write() . . . 151
12.13 gdcm::network::AReleaseRQPDU Class Reference . . . 151
12.13.1 Detailed Description . . . 152
12.13.2 Constructor & Destructor Documentation . . . 153
12.13.2.1 AReleaseRQPDU() . . . 153
12.13.3 Member Function Documentation . . . 153
12.13.3.1 IsLastFragment() . . . 153
12.13.3.2 Print() . . . 153
12.13.3.3 Read() . . . 153
12.13.3.4 Size() . . . 153
12.13.3.5 Write() . . . 153
12.14 gdcm::network::ARTIMTimer Class Reference . . . 154
12.14.1 Detailed Description . . . 154
12.14.2 Constructor & Destructor Documentation . . . 154
12.14.2.1 ARTIMTimer() . . . 154
12.14.3 Member Function Documentation . . . 154
12.14.3.1 GetElapsedTime() . . . 154
12.14.3.2 GetHasExpired() . . . 155
12.14.3.3 GetTimeout() . . . 155

12.14.3.4 SetTimeout() . . . 155
12.14.3.5 Start() . . . 155
12.14.3.6 Stop() . . . 155
12.15 gdcm::ASN1 Class Reference . . . 155
12.15.1 Detailed Description . . . 156
12.15.2 Constructor & Destructor Documentation . . . 156
12.15.2.1 ASN1() [1/2] . . . 156
12.15.2.2 ∼ASN1() . . . 156
12.15.2.3 ASN1() [2/2] . . . 156
12.15.3 Member Function Documentation . . . 156
12.15.3.1 operator=() . . . 156
12.15.3.2 ParseDump() . . . 156
12.15.3.3 ParseDumpFile() . . . 157
12.15.3.4 TestPBKDF2() . . . 157
12.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference . . . 157
12.16.1 Detailed Description . . . 157
12.16.2 Constructor & Destructor Documentation . . . 157
12.16.2.1 AsynchronousOperationsWindowSub() . . . 157
12.16.3 Member Function Documentation . . . 158
12.16.3.1 Print() . . . 158
12.16.3.2 Read() . . . 158
12.16.3.3 Size() . . . 158
12.16.3.4 Write() . . . 158
12.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference . . . 158
12.17.1 Detailed Description . . . 160
12.17.2 Member Typedef Documentation . . . 160
12.17.2.1 ArrayType . . . 160
12.17.3 Member Enumeration Documentation . . . 161
12.17.3.1 anonymous enum . . . 161
12.17.4 Member Function Documentation . . . 161
12.17.4.1 GDCM_STATIC_ASSERT() [1/3] . . . 161
12.17.4.2 GDCM_STATIC_ASSERT() [2/3] . . . 161
12.17.4.3 GDCM_STATIC_ASSERT() [3/3] . . . 161
12.17.4.4 GetAsDataElement() . . . 162
12.17.4.5 GetDictVM() . . . 162
12.17.4.6 GetDictVR() . . . 162
12.17.4.7 GetNumberOfValues() . . . 162
12.17.4.8 GetTag() . . . 163
12.17.4.9 GetValue() [1/2] . . . 163

12.17.4.10 GetValue() [2/2] . . . 163
12.17.4.11 GetValues() . . . 163
12.17.4.12 GetVM() . . . 164
12.17.4.13 GetVR() . . . 164
12.17.4.14 operator"!=() . . . 164
12.17.4.15 operator<() . . . 164
12.17.4.16 operator==() . . . 164
12.17.4.17 operator[]() [1/2] . . . 165
12.17.4.18 operator[]() [2/2] . . . 165
12.17.4.19 Print() . . . 165
12.17.4.20 Set() . . . 165
12.17.4.21 SetByteValue() . . . 165
12.17.4.22 SetByteValueNoSwap() . . . 166
12.17.4.23 SetFromDataElement() . . . 166
12.17.4.24 SetFromDataSet() . . . 166
12.17.4.25 SetValue() . . . 167
12.17.4.26 SetValues() . . . 167
12.17.5 Member Data Documentation . . . 167
12.17.5.1 Internal . . . 167
12.18 gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference . . . 168
12.18.1 Member Typedef Documentation . . . 170
12.18.1.1 ArrayType . . . 170
12.18.2 Member Enumeration Documentation . . . 170
12.18.2.1 anonymous enum . . . 170
12.18.2.2 anonymous enum . . . 171
12.18.3 Member Function Documentation . . . 171
12.18.3.1 GDCM_STATIC_ASSERT() [1/4] . . . 171
12.18.3.2 GDCM_STATIC_ASSERT() [2/4] . . . 171
12.18.3.3 GDCM_STATIC_ASSERT() [3/4] . . . 171
12.18.3.4 GDCM_STATIC_ASSERT() [4/4] . . . 171
12.18.3.5 GetAsDataElement() . . . 171
12.18.3.6 GetDictVM() . . . 172
12.18.3.7 GetDictVR() . . . 172
12.18.3.8 GetNumberOfValues() . . . 172
12.18.3.9 GetTag() . . . 172
12.18.3.10 GetValue() [1/2] . . . 172
12.18.3.11 GetValue() [2/2] . . . 172
12.18.3.12 GetValues() . . . 172
12.18.3.13 GetVM() . . . 173

12.18.3.14 GetVR() . . . 173
12.18.3.15 operator"!=() . . . 173
12.18.3.16 operator<() . . . 173
12.18.3.17 operator==() . . . 173
12.18.3.18 operator[]() . . . 173
12.18.3.19 Print() . . . 174
12.18.3.20 Set() . . . 174
12.18.3.21 SetByteValue() . . . 174
12.18.3.22 SetByteValueNoSwap() . . . 174
12.18.3.23 SetFromDataElement() . . . 174
12.18.3.24 SetFromDataSet() . . . 175
12.18.3.25 SetValue() . . . 175
12.18.3.26 SetValues() . . . 175
12.18.4 Member Data Documentation . . . 175
12.18.4.1 Internal . . . 175
12.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference . . . 176
12.19.1 Member Typedef Documentation . . . 178
12.19.1.1 ArrayType . . . 178
12.19.2 Member Enumeration Documentation . . . 178
12.19.2.1 anonymous enum . . . 178
12.19.3 Member Function Documentation . . . 178
12.19.3.1 GDCM_STATIC_ASSERT() . . . 178
12.19.3.2 GetAsDataElement() . . . 178
12.19.3.3 GetDictVM() . . . 178
12.19.3.4 GetDictVR() . . . 179
12.19.3.5 GetNumberOfValues() . . . 179
12.19.3.6 GetTag() . . . 179
12.19.3.7 GetValue() . . . 179
12.19.3.8 GetValues() . . . 179
12.19.3.9 GetVM() . . . 179
12.19.3.10 GetVR() . . . 179
12.19.3.11 operator"!=() . . . 179
12.19.3.12 operator<() . . . 179
12.19.3.13 operator==() . . . 180
12.19.3.14 operator[]() . . . 180
12.19.3.15 Print() . . . 180
12.19.3.16 Set() . . . 180
12.19.3.17 SetByteValue() . . . 180
12.19.3.18 SetByteValueNoSwap() . . . 180

12.19.3.19 SetFromDataElement() . . . 180
12.19.3.20 SetFromDataSet() . . . 180
12.19.3.21 SetValue() . . . 181
12.19.3.22 SetValues() . . . 181
12.19.4 Member Data Documentation . . . 181
12.19.4.1 Internal . . . 181
12.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference . . . 181
12.20.1 Member Typedef Documentation . . . 183
12.20.1.1 ArrayType . . . 183
12.20.2 Member Enumeration Documentation . . . 184
12.20.2.1 anonymous enum . . . 184
12.20.3 Member Function Documentation . . . 184
12.20.3.1 GDCM_STATIC_ASSERT() . . . 184
12.20.3.2 GetAsDataElement() . . . 184
12.20.3.3 GetDictVM() . . . 184
12.20.3.4 GetDictVR() . . . 184
12.20.3.5 GetNumberOfValues() . . . 184
12.20.3.6 GetTag() . . . 184
12.20.3.7 GetValue() . . . 184
12.20.3.8 GetValues() . . . 185
12.20.3.9 GetVM() . . . 185
12.20.3.10 GetVR() . . . 185
12.20.3.11 operator"!=() . . . 185
12.20.3.12 operator<() . . . 185
12.20.3.13 operator==() . . . 185
12.20.3.14 operator[]() . . . 185
12.20.3.15 Print() . . . 185
12.20.3.16 Set() . . . 186
12.20.3.17 SetByteValue() . . . 186
12.20.3.18 SetByteValueNoSwap() . . . 186
12.20.3.19 SetFromDataElement() . . . 186
12.20.3.20 SetFromDataSet() . . . 186
12.20.3.21 SetValue() . . . 186
12.20.3.22 SetValues() . . . 186
12.20.4 Member Data Documentation . . . 187
12.20.4.1 Internal . . . 187
12.21 gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference . . . 187
12.21.1 Member Typedef Documentation . . . 189
12.21.1.1 ArrayType . . . 189

12.21.2 Member Enumeration Documentation . . . 189
12.21.2.1 anonymous enum . . . 189
12.21.3 Constructor & Destructor Documentation . . . 190
12.21.3.1 Attribute() . . . 190
12.21.3.2 ∼Attribute() . . . 190
12.21.4 Member Function Documentation . . . 190
12.21.4.1 GDCM_STATIC_ASSERT() [1/3] . . . 190
12.21.4.2 GDCM_STATIC_ASSERT() [2/3] . . . 190
12.21.4.3 GDCM_STATIC_ASSERT() [3/3] . . . 190
12.21.4.4 GetAsDataElement() . . . 190
12.21.4.5 GetDictVM() . . . 191
12.21.4.6 GetDictVR() . . . 191
12.21.4.7 GetNumberOfValues() . . . 191
12.21.4.8 GetTag() . . . 191
12.21.4.9 GetValue() [1/2] . . . 191
12.21.4.10 GetValue() [2/2] . . . 191
12.21.4.11 GetValues() . . . 192
12.21.4.12 GetVM() . . . 192
12.21.4.13 GetVR() . . . 192
12.21.4.14 operator"!=() . . . 192
12.21.4.15 operator<() . . . 192
12.21.4.16 operator==() . . . 192
12.21.4.17 operator[]() [1/2] . . . 192
12.21.4.18 operator[]() [2/2] . . . 193
12.21.4.19 Print() . . . 193
12.21.4.20 Set() . . . 193
12.21.4.21 SetByteValue() . . . 193
12.21.4.22 SetByteValueNoSwap() . . . 193
12.21.4.23 SetFromDataElement() . . . 193
12.21.4.24 SetFromDataSet() . . . 194
12.21.4.25 SetNumberOfValues() . . . 194
12.21.4.26 SetValue() [1/2] . . . 194
12.21.4.27 SetValue() [2/2] . . . 194
12.21.4.28 SetValues() . . . 194
12.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference . . . 195
12.22.1 Member Typedef Documentation . . . 198
12.22.1.1 ArrayType . . . 198
12.22.2 Member Enumeration Documentation . . . 198
12.22.2.1 anonymous enum . . . 198

12.22.3 Member Function Documentation . . . 198
12.22.3.1 GDCM_STATIC_ASSERT() . . . 198
12.22.3.2 GetAsDataElement() . . . 199
12.22.3.3 GetDictVM() . . . 199
12.22.3.4 GetDictVR() . . . 199
12.22.3.5 GetNumberOfValues() . . . 199
12.22.3.6 GetTag() . . . 199
12.22.3.7 GetValue() . . . 199
12.22.3.8 GetValues() . . . 199
12.22.3.9 GetVM() . . . 199
12.22.3.10 GetVR() . . . 199
12.22.3.11 operator"!=() . . . 200
12.22.3.12 operator<() . . . 200
12.22.3.13 operator==() . . . 200
12.22.3.14 operator[]() . . . 200
12.22.3.15 Print() . . . 200
12.22.3.16 Set() . . . 200
12.22.3.17 SetByteValue() . . . 200
12.22.3.18 SetByteValueNoSwap() . . . 200
12.22.3.19 SetFromDataElement() . . . 201
12.22.3.20 SetFromDataSet() . . . 201
12.22.3.21 SetValue() . . . 201
12.22.3.22 SetValues() . . . 201
12.22.4 Member Data Documentation . . . 201
12.22.4.1 Internal . . . 201
12.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference . . . 202
12.23.1 Member Typedef Documentation . . . 204
12.23.1.1 ArrayType . . . 204
12.23.2 Member Enumeration Documentation . . . 204
12.23.2.1 anonymous enum . . . 204
12.23.3 Member Function Documentation . . . 204
12.23.3.1 GDCM_STATIC_ASSERT() . . . 204
12.23.3.2 GetAsDataElement() . . . 204
12.23.3.3 GetDictVM() . . . 204
12.23.3.4 GetDictVR() . . . 205
12.23.3.5 GetNumberOfValues() . . . 205
12.23.3.6 GetTag() . . . 205
12.23.3.7 GetValue() . . . 205
12.23.3.8 GetValues() . . . 205

12.23.3.9 GetVM() . . . 205
12.23.3.10 GetVR() . . . 205
12.23.3.11 operator"!=() . . . 205
12.23.3.12 operator<() . . . 205
12.23.3.13 operator==() . . . 206
12.23.3.14 operator[]() . . . 206
12.23.3.15 Print() . . . 206
12.23.3.16 Set() . . . 206
12.23.3.17 SetByteValue() . . . 206
12.23.3.18 SetByteValueNoSwap() . . . 206
12.23.3.19 SetFromDataElement() . . . 206
12.23.3.20 SetFromDataSet() . . . 206
12.23.3.21 SetValue() . . . 207
12.23.3.22 SetValues() . . . 207
12.23.4 Member Data Documentation . . . 207
12.23.4.1 Internal . . . 207
12.24 gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference . . . 207
12.24.1 Member Typedef Documentation . . . 210
12.24.1.1 ArrayType . . . 210
12.24.2 Member Enumeration Documentation . . . 211
12.24.2.1 anonymous enum . . . 211
12.24.3 Member Function Documentation . . . 211
12.24.3.1 GDCM_STATIC_ASSERT() . . . 211
12.24.3.2 GetAsDataElement() . . . 211
12.24.3.3 GetDictVM() . . . 211
12.24.3.4 GetDictVR() . . . 211
12.24.3.5 GetNumberOfValues() . . . 211
12.24.3.6 GetTag() . . . 211
12.24.3.7 GetValue() . . . 211
12.24.3.8 GetValues() . . . 212
12.24.3.9 GetVM() . . . 212
12.24.3.10 GetVR() . . . 212
12.24.3.11 operator"!=() . . . 212
12.24.3.12 operator<() . . . 212
12.24.3.13 operator==() . . . 212
12.24.3.14 operator[]() . . . 212
12.24.3.15 Print() . . . 212
12.24.3.16 Set() . . . 213
12.24.3.17 SetByteValue() . . . 213

12.24.3.18 SetByteValueNoSwap() . . . 213
12.24.3.19 SetFromDataElement() . . . 213
12.24.3.20 SetFromDataSet() . . . 213
12.24.3.21 SetValue() . . . 213
12.24.3.22 SetValues() . . . 213
12.24.4 Member Data Documentation . . . 214
12.24.4.1 Internal . . . 214
12.25 gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference . . . 214
12.25.1 Member Typedef Documentation . . . 216
12.25.1.1 ArrayType . . . 216
12.25.2 Member Enumeration Documentation . . . 216
12.25.2.1 anonymous enum . . . 216
12.25.3 Member Function Documentation . . . 217
12.25.3.1 GDCM_STATIC_ASSERT() . . . 217
12.25.3.2 GetAsDataElement() . . . 217
12.25.3.3 GetDictVM() . . . 217
12.25.3.4 GetDictVR() . . . 217
12.25.3.5 GetNumberOfValues() . . . 217
12.25.3.6 GetTag() . . . 217
12.25.3.7 GetValue() . . . 217
12.25.3.8 GetValues() . . . 217
12.25.3.9 GetVM() . . . 218
12.25.3.10 GetVR() . . . 218
12.25.3.11 operator"!=() . . . 218
12.25.3.12 operator<() . . . 218
12.25.3.13 operator==() . . . 218
12.25.3.14 operator[]() . . . 218
12.25.3.15 Print() . . . 218
12.25.3.16 Set() . . . 218
12.25.3.17 SetByteValue() . . . 219
12.25.3.18 SetByteValueNoSwap() . . . 219
12.25.3.19 SetFromDataElement() . . . 219
12.25.3.20 SetFromDataSet() . . . 219
12.25.3.21 SetValue() . . . 219
12.25.3.22 SetValues() . . . 219
12.25.4 Member Data Documentation . . . 219
12.25.4.1 Internal . . . 219
12.26 gdcm::AudioCodec Class Reference . . . 220
12.26.1 Detailed Description . . . 221

12.26.2 Constructor & Destructor Documentation . . . 221
12.26.2.1 AudioCodec() . . . 221
12.26.2.2 ∼AudioCodec() . . . 222
12.26.3 Member Function Documentation . . . 222
12.26.3.1 CanCode() . . . 222
12.26.3.2 CanDecode() . . . 222
12.26.3.3 Decode() . . . 222
12.27 gdcm::Base64 Class Reference . . . 222
12.27.1 Detailed Description . . . 223
12.27.2 Constructor & Destructor Documentation . . . 223
12.27.2.1 Base64() . . . 223
12.27.3 Member Function Documentation . . . 223
12.27.3.1 Decode() . . . 223
12.27.3.2 Encode() . . . 224
12.27.3.3 GetDecodeLength() . . . 224
12.27.3.4 GetEncodeLength() . . . 225
12.27.3.5 operator=() . . . 225
12.28 gdcm::network::BaseCompositeMessage Class Reference . . . 225
12.28.1 Detailed Description . . . 226
12.28.2 Constructor & Destructor Documentation . . . 227
12.28.2.1 ∼BaseCompositeMessage() . . . 227
12.28.3 Member Function Documentation . . . 227
12.28.3.1 ConstructPDV() . . . 227
12.29 gdcm::network::BaseNormalizedMessage Class Reference . . . 227
12.29.1 Detailed Description . . . 228
12.29.2 Constructor & Destructor Documentation . . . 229
12.29.2.1 ∼BaseNormalizedMessage() . . . 229
12.29.3 Member Function Documentation . . . 229
12.29.3.1 ConstructPDV() . . . 229
12.30 gdcm::network::BasePDU Class Reference . . . 230
12.30.1 Detailed Description . . . 230
12.30.2 Constructor & Destructor Documentation . . . 231
12.30.2.1 ∼BasePDU() . . . 231
12.30.3 Member Function Documentation . . . 231
12.30.3.1 IsLastFragment() . . . 231
12.30.3.2 Print() . . . 231
12.30.3.3 Read() . . . 231
12.30.3.4 Size() . . . 232
12.30.3.5 Write() . . . 232

12.31 gdcm::BaseQuery Class Reference . . . 232
12.31.1 Detailed Description . . . 234
12.31.2 Constructor & Destructor Documentation . . . 234
12.31.2.1 BaseQuery() . . . 234
12.31.2.2 ~BaseQuery() . . . 234
12.31.3 Member Function Documentation . . . 234
12.31.3.1 AddQueryDataSet() . . . 234
12.31.3.2 GetAbstractSyntaxUID() . . . 235
12.31.3.3 GetQueryDataSet() [1/2] . . . 235
12.31.3.4 GetQueryDataSet() [2/2] . . . 235
12.31.3.5 GetSOPInstanceUID() . . . 235
12.31.3.6 Print() . . . 235
12.31.3.7 SetSearchParameter() [1/3] . . . 235
12.31.3.8 SetSearchParameter() [2/3] . . . 235
12.31.3.9 SetSearchParameter() [3/3] . . . 236
12.31.3.10 SetSOPInstanceUID() . . . 236
12.31.3.11 ValidateQuery() . . . 236
12.31.3.12 ValidDataSet() . . . 236
12.31.3.13 WriteHelpFile() . . . 236
12.31.3.14 WriteQuery() . . . 236
12.31.4 Friends And Related Symbol Documentation . . . 236
12.31.4.1 QueryFactory . . . 236
12.31.5 Member Data Documentation . . . 237
12.31.5.1 mDataSet . . . 237
12.31.5.2 mSopInstanceUID . . . 237
12.32 gdcm::BaseRootQuery Class Reference . . . 237
12.32.1 Detailed Description . . . 239
12.32.2 Constructor & Destructor Documentation . . . 239
12.32.2.1 BaseRootQuery() . . . 239
12.32.2.2 ~BaseRootQuery() . . . 240
12.32.3 Member Function Documentation . . . 240
12.32.3.1 Construct() . . . 240
12.32.3.2 GetQueryLevelFromQueryRoot() . . . 240
12.32.3.3 GetQueryLevelFromString() . . . 240
12.32.3.4 GetQueryLevelString() . . . 240
12.32.3.5 GetTagListByLevel() . . . 240
12.32.3.6 InitializeDataSet() . . . 241
12.32.3.7 ValidateQuery() . . . 241
12.32.4 Friends And Related Symbol Documentation . . . 241

12.32.4.1 QueryFactory . . . 241
12.32.5 Member Data Documentation . . . 241
12.32.5.1 mHelpDescription . . . 241
12.32.5.2 mImage . . . 242
12.32.5.3 mPatient . . . 242
12.32.5.4 mRootType . . . 242
12.32.5.5 mSeries . . . 242
12.32.5.6 mStudy . . . 242
12.33 gdcm::SegmentHelper::BasicCodedEntry Struct Reference . . . 242
12.33.1 Detailed Description . . . 244
12.33.2 Constructor & Destructor Documentation . . . 244
12.33.2.1 BasicCodedEntry() [1/3] . . . 244
12.33.2.2 BasicCodedEntry() [2/3] . . . 244
12.33.2.3 BasicCodedEntry() [3/3] . . . 244
12.33.3 Member Function Documentation . . . 245
12.33.3.1 IsEmpty() . . . 245
12.33.4 Member Data Documentation . . . 245
12.33.4.1 CM . . . 245
12.33.4.2 CSD . . . 245
12.33.4.3 CSV . . . 245
12.33.4.4 CV . . . 245
12.34 gdcm::BasicOffsetTable Class Reference . . . 246
12.34.1 Detailed Description . . . 249
12.34.2 Constructor & Destructor Documentation . . . 249
12.34.2.1 BasicOffsetTable() . . . 249
12.34.3 Member Function Documentation . . . 249
12.34.3.1 Read() . . . 249
12.34.4 Friends And Related Symbol Documentation . . . 249
12.34.4.1 operator<< . . . 249
12.35 gdcm::Bitmap Class Reference . . . 250
12.35.1 Detailed Description . . . 253
12.35.2 Member Typedef Documentation . . . 253
12.35.2.1 LUTPtr . . . 253
12.35.3 Constructor & Destructor Documentation . . . 253
12.35.3.1 Bitmap() . . . 253
12.35.3.2 ~Bitmap() . . . 253
12.35.4 Member Function Documentation . . . 253
12.35.4.1 AreOverlaysInPixelData() . . . 253
12.35.4.2 Clear() . . . 253

12.35.4.3 ComputeLossyFlag() . . . 254
12.35.4.4 GetBuffer() . . . 254
12.35.4.5 GetBuffer2() . . . 254
12.35.4.6 GetBufferLength() . . . 254
12.35.4.7 GetColumns() . . . 254
12.35.4.8 GetDataElement() [1/2] . . . 254
12.35.4.9 GetDataElement() [2/2] . . . 255
12.35.4.10 GetDimension() . . . 255
12.35.4.11 GetDimensions() . . . 255
12.35.4.12 GetLUT() [1/2] . . . 255
12.35.4.13 GetLUT() [2/2] . . . 255
12.35.4.14 GetNeedByteSwap() . . . 256
12.35.4.15 GetNumberOfDimensions() . . . 256
12.35.4.16 GetPhotometricInterpretation() . . . 256
12.35.4.17 GetPixelFormat() [1/2] . . . 256
12.35.4.18 GetPixelFormat() [2/2] . . . 256
12.35.4.19 GetPlanarConfiguration() . . . 257
12.35.4.20 GetRows() . . . 257
12.35.4.21 GetTransferSyntax() . . . 257
12.35.4.22 IsEmpty() . . . 257
12.35.4.23 IsLossy() . . . 257
12.35.4.24 IsTransferSyntaxCompatible() . . . 257
12.35.4.25 Print() . . . 258
12.35.4.26 SetColumns() . . . 258
12.35.4.27 SetDataElement() . . . 258
12.35.4.28 SetDimension() . . . 258
12.35.4.29 SetDimensions() . . . 259
12.35.4.30 SetLossyFlag() . . . 259
12.35.4.31 SetLUT() . . . 259
12.35.4.32 SetNeedByteSwap() . . . 259
12.35.4.33 SetNumberOfDimensions() . . . 259
12.35.4.34 SetPhotometricInterpretation() . . . 260
12.35.4.35 SetPixelFormat() . . . 260
12.35.4.36 SetPlanarConfiguration() . . . 260
12.35.4.37 SetRows() . . . 260
12.35.4.38 SetTransferSyntax() . . . 261
12.35.4.39 TryJPEG2000Codec() . . . 261
12.35.4.40 TryJPEG2000Codec2() . . . 261
12.35.4.41 TryJPEGCodec() . . . 261

12.35.4.42 TryJPEGCodec2() . . . 261
12.35.4.43 TryJPEGLSCodec() . . . 261
12.35.4.44 TryKAKADUCodec() . . . 262
12.35.4.45 TryPVRGCodec() . . . 262
12.35.4.46 TryRAWCodec() . . . 262
12.35.4.47 TryRLECodec() . . . 262
12.35.4.48 UnusedBitsPresentInPixelData() . . . 262
12.35.5 Friends And Related Symbol Documentation . . . 262
12.35.5.1 ImageChangeTransferSyntax . . . 262
12.35.5.2 PixmapReader . . . 263
12.35.6 Member Data Documentation . . . 263
12.35.6.1 Dimensions . . . 263
12.35.6.2 LossyFlag . . . 263
12.35.6.3 LUT . . . 263
12.35.6.4 NeedByteSwap . . . 263
12.35.6.5 NumberOfDimensions . . . 263
12.35.6.6 PF . . . 263
12.35.6.7 PI . . . 264
12.35.6.8 PixelData . . . 264
12.35.6.9 PlanarConfiguration . . . 264
12.35.6.10 TS . . . 264
12.36 gdcm::BitmapToBitmapFilter Class Reference . . . 264
12.36.1 Detailed Description . . . 265
12.36.2 Constructor & Destructor Documentation . . . 266
12.36.2.1 BitmapToBitmapFilter() . . . 266
12.36.2.2 ~BitmapToBitmapFilter() . . . 266
12.36.3 Member Function Documentation . . . 266
12.36.3.1 GetOutput() . . . 266
12.36.3.2 GetOutputAsBitmap() . . . 266
12.36.3.3 SetInput() . . . 266
12.36.4 Member Data Documentation . . . 266
12.36.4.1 Input . . . 266
12.36.4.2 Output . . . 267
12.37 gdcm::BoxRegion Class Reference . . . 267
12.37.1 Detailed Description . . . 268
12.37.2 Constructor & Destructor Documentation . . . 269
12.37.2.1 BoxRegion() [1/2] . . . 269
12.37.2.2 ~BoxRegion() . . . 269
12.37.2.3 BoxRegion() [2/2] . . . 269

12.37.3 Member Function Documentation . . . 269
12.37.3.1 Area() . . . 269
12.37.3.2 BoundingBox() . . . 269
12.37.3.3 Clone() . . . 270
12.37.3.4 ComputeBoundingBox() . . . 270
12.37.3.5 Empty() . . . 270
12.37.3.6 GetXMax() . . . 270
12.37.3.7 GetXMin() . . . 270
12.37.3.8 GetYMax() . . . 270
12.37.3.9 GetYMin() . . . 270
12.37.3.10 GetZMax() . . . 271
12.37.3.11 GetZMin() . . . 271
12.37.3.12 IsValid() . . . 271
12.37.3.13 operator=() . . . 271
12.37.3.14 Print() . . . 271
12.37.3.15 SetDomain() . . . 271
12.38 gdcm::ByteBuffer Class Reference . . . 272
12.38.1 Detailed Description . . . 272
12.38.2 Constructor & Destructor Documentation . . . 272
12.38.2.1 ByteBuffer() . . . 272
12.38.3 Member Function Documentation . . . 272
12.38.3.1 Get() . . . 272
12.38.3.2 GetStart() . . . 272
12.38.3.3 ShiftEnd() . . . 273
12.38.3.4 UpdatePosition() . . . 273
12.39 gdcm::ByteSwap< T > Class Template Reference . . . 273
12.39.1 Detailed Description . . . 273
12.39.2 Member Function Documentation . . . 273
12.39.2.1 Swap() . . . 273
12.39.2.2 SwapFromSwapCodeIntoSystem() . . . 274
12.39.2.3 SwapRange() . . . 274
12.39.2.4 SwapRangeFromSwapCodeIntoSystem() . . . 274
12.39.2.5 SystemIsBigEndian() . . . 274
12.39.2.6 SystemIsLittleEndian() . . . 275
12.40 gdcm::ByteSwapFilter Class Reference . . . 275
12.40.1 Detailed Description . . . 275
12.40.2 Constructor & Destructor Documentation . . . 275
12.40.2.1 ByteSwapFilter() [1/2] . . . 275
12.40.2.2 ~ByteSwapFilter() . . . 276

12.40.2.3 ByteSwapFilter() [2/2] . . . 276
12.40.3 Member Function Documentation . . . 276
12.40.3.1 ByteSwap() . . . 276
12.40.3.2 operator=() . . . 276
12.40.3.3 SetByteSwapTag() . . . 276
12.41 gdcm::ByteValue Class Reference . . . 277
12.41.1 Detailed Description . . . 279
12.41.2 Constructor & Destructor Documentation . . . 279
12.41.2.1 ByteValue() [1/2] . . . 279
12.41.2.2 ByteValue() [2/2] . . . 279
12.41.2.3 ∼ByteValue() . . . 279
12.41.3 Member Function Documentation . . . 280
12.41.3.1 Append() . . . 280
12.41.3.2 Clear() . . . 280
12.41.3.3 ComputeLength() . . . 280
12.41.3.4 Fill() . . . 280
12.41.3.5 GetBuffer() . . . 280
12.41.3.6 GetLength() . . . 281
12.41.3.7 GetPointer() . . . 281
12.41.3.8 GetVoidPointer() [1/2] . . . 281
12.41.3.9 GetVoidPointer() [2/2] . . . 281
12.41.3.10 IsEmpty() . . . 282
12.41.3.11 IsPrintable() . . . 282
12.41.3.12 operator const std::vector< char > &() . . . 282
12.41.3.13 operator=() . . . 282
12.41.3.14 operator==() [1/2] . . . 282
12.41.3.15 operator==() [2/2] . . . 282
12.41.3.16 Print() . . . 283
12.41.3.17 PrintASCII() . . . 283
12.41.3.18 PrintASCIIXML() . . . 283
12.41.3.19 PrintGroupLength() . . . 283
12.41.3.20 PrintHex() . . . 283
12.41.3.21 PrintHexXML() . . . 283
12.41.3.22 PrintPNXML() . . . 283
12.41.3.23 Read() [1/2] . . . 284
12.41.3.24 Read() [2/2] . . . 284
12.41.3.25 SetLength() . . . 284
12.41.3.26 SetLengthOnly() . . . 284
12.41.3.27 Write() [1/2] . . . 284

12.41.3.28 Write() [2/2] . . . 284
12.41.3.29 WriteBuffer() . . . 285
12.42 gdcm::CAPICryptoFactory Class Reference . . . 285
12.42.1 Constructor & Destructor Documentation . . . 286
12.42.1.1 CAPICryptoFactory() . . . 286
12.42.2 Member Function Documentation . . . 286
12.42.2.1 CreateCMSProvider() . . . 286
12.43 gdcm::CAPICryptographicMessageSyntax Class Reference . . . 287
12.43.1 Constructor & Destructor Documentation . . . 288
12.43.1.1 CAPICryptographicMessageSyntax() . . . 288
12.43.1.2 ∼CAPICryptographicMessageSyntax() . . . 288
12.43.2 Member Function Documentation . . . 288
12.43.2.1 Decrypt() . . . 288
12.43.2.2 Encrypt() . . . 289
12.43.2.3 GetCipherType() . . . 289
12.43.2.4 GetInitialized() . . . 289
12.43.2.5 ParseCertificateFile() . . . 289
12.43.2.6 ParseKeyFile() . . . 289
12.43.2.7 SetCipherType() . . . 289
12.43.2.8 SetPassword() . . . 290
12.44 gdcm::network::CEchoRQ Class Reference . . . 290
12.44.1 Detailed Description . . . 291
12.44.2 Member Function Documentation . . . 292
12.44.2.1 ConstructPDV() . . . 292
12.44.3 Member Data Documentation . . . 292
12.44.3.1 AffectedSOPClassUID . . . 292
12.44.3.2 MessageID . . . 292
12.45 gdcm::network::CEchoRSP Class Reference . . . 292
12.45.1 Detailed Description . . . 293
12.45.2 Member Function Documentation . . . 293
12.45.2.1 ConstructPDVByDataSet() . . . 293
12.46 gdcm::network::CFind Class Reference . . . 293
12.46.1 Detailed Description . . . 294
12.47 gdcm::network::CFindCancelRQ Class Reference . . . 294
12.47.1 Detailed Description . . . 295
12.47.2 Member Function Documentation . . . 295
12.47.2.1 ConstructPDVByDataSet() . . . 295
12.48 gdcm::network::CFindRQ Class Reference . . . 295
12.48.1 Detailed Description . . . 296

12.48.2 Member Function Documentation . . . 296
12.48.2.1 ConstructPDV() . . . 296
12.49 gdcm::network::CFindRSP Class Reference . . . 297
12.49.1 Detailed Description . . . 298
12.49.2 Member Function Documentation . . . 298
12.49.2.1 ConstructPDVByDataSet() . . . 298
12.50 gdcm::Cleaner Class Reference . . . 298
12.50.1 Detailed Description . . . 300
12.50.2 Constructor & Destructor Documentation . . . 301
12.50.2.1 Cleaner() . . . 301
12.50.2.2 ∼Cleaner() . . . 301
12.50.3 Member Function Documentation . . . 301
12.50.3.1 Clean() . . . 301
12.50.3.2 Empty() [1/4] . . . 301
12.50.3.3 Empty() [2/4] . . . 301
12.50.3.4 Empty() [3/4] . . . 301
12.50.3.5 Empty() [4/4] . . . 302
12.50.3.6 EmptyWhenScrubFails() . . . 302
12.50.3.7 GetFile() . . . 302
12.50.3.8 New() . . . 302
12.50.3.9 Preserve() . . . 302
12.50.3.10 Remove() [1/4] . . . 302
12.50.3.11 Remove() [2/4] . . . 303
12.50.3.12 Remove() [3/4] . . . 303
12.50.3.13 Remove() [4/4] . . . 303
12.50.3.14 RemoveAllGroupLength() . . . 303
12.50.3.15 RemoveAllIllegal() . . . 303
12.50.3.16 RemoveAllMissingPrivateCreator() . . . 303
12.50.3.17 RemoveMissingPrivateCreator() . . . 304
12.50.3.18 Scrub() [1/4] . . . 304
12.50.3.19 Scrub() [2/4] . . . 304
12.50.3.20 Scrub() [3/4] . . . 304
12.50.3.21 Scrub() [4/4] . . . 304
12.50.3.22 SetFile() . . . 304
12.51 gdcm::network::CMoveCancelRq Class Reference . . . 305
12.51.1 Member Function Documentation . . . 306
12.51.1.1 ConstructPDVByDataSet() . . . 306
12.52 gdcm::network::CMoveRQ Class Reference . . . 306
12.52.1 Detailed Description . . . 307

12.52.2 Member Function Documentation . . . 307
12.52.2.1 ConstructPDV() . . . 307
12.53 gdcm::network::CMoveRSP Class Reference . . . 307
12.53.1 Detailed Description . . . 308
12.53.2 Member Function Documentation . . . 308
12.53.2.1 ConstructPDVByDataSet() . . . 308
12.54 gdcm::Codec Class Reference . . . 309
12.54.1 Detailed Description . . . 310
12.55 gdcm::Coder Class Reference . . . 310
12.55.1 Detailed Description . . . 311
12.55.2 Constructor & Destructor Documentation . . . 311
12.55.2.1 ∼Coder() . . . 311
12.55.3 Member Function Documentation . . . 311
12.55.3.1 CanCode() . . . 311
12.55.3.2 Code() . . . 311
12.55.3.3 InternalCode() . . . 312
12.56 gdcm::CodeString Class Reference . . . 312
12.56.1 Detailed Description . . . 313
12.56.2 Member Typedef Documentation . . . 313
12.56.2.1 const_iterator . . . 313
12.56.2.2 const_reference . . . 313
12.56.2.3 const_reverse_iterator . . . 313
12.56.2.4 difference_type . . . 314
12.56.2.5 iterator . . . 314
12.56.2.6 pointer . . . 314
12.56.2.7 reference . . . 314
12.56.2.8 reverse_iterator . . . 314
12.56.2.9 size_type . . . 314
12.56.2.10 value_type . . . 314
12.56.3 Constructor & Destructor Documentation . . . 314
12.56.3.1 CodeString() [1/4] . . . 314
12.56.3.2 CodeString() [2/4] . . . 315
12.56.3.3 CodeString() [3/4] . . . 315
12.56.3.4 CodeString() [4/4] . . . 315
12.56.4 Member Function Documentation . . . 315
12.56.4.1 GetAsString() . . . 315
12.56.4.2 IsValid() . . . 315
12.56.4.3 Size() . . . 315
12.56.4.4 TrimInternal() . . . 315

12.56.5 Friends And Related Symbol Documentation . . . 316
12.56.5.1 operator"!= . . . 316
12.56.5.2 operator<< . . . 316
12.56.5.3 operator== . . . 316
12.57 gdcm::Command Class Reference . . . 316
12.57.1 Detailed Description . . . 318
12.57.2 Constructor & Destructor Documentation . . . 318
12.57.2.1 Command() [1/2] . . . 318
12.57.2.2 Command() [2/2] . . . 318
12.57.2.3 ~Command() . . . 318
12.57.3 Member Function Documentation . . . 319
12.57.3.1 Execute() [1/2] . . . 319
12.57.3.2 Execute() [2/2] . . . 319
12.57.3.3 operator=() . . . 319
12.58 gdcm::CommandDataSet Class Reference . . . 320
12.58.1 Detailed Description . . . 322
12.58.2 Constructor & Destructor Documentation . . . 322
12.58.2.1 CommandDataSet() . . . 322
12.58.2.2 ~CommandDataSet() . . . 323
12.58.3 Member Function Documentation . . . 323
12.58.3.1 Insert() . . . 323
12.58.3.2 Read() . . . 323
12.58.3.3 Replace() . . . 323
12.58.3.4 Write() . . . 323
12.58.4 Friends And Related Symbol Documentation . . . 324
12.58.4.1 operator<< . . . 324
12.59 gdcm::network::CompositeMessageFactory Class Reference . . . 324
12.59.1 Detailed Description . . . 324
12.59.2 Member Function Documentation . . . 325
12.59.2.1 ConstructCEchoRQ() . . . 325
12.59.2.2 ConstructCFindRQ() . . . 325
12.59.2.3 ConstructCMoveRQ() . . . 325
12.59.2.4 ConstructCStoreRQ() . . . 325
12.59.2.5 ConstructCStoreRSP() . . . 325
12.60 gdcm::CompositeNetworkFunctions Class Reference . . . 325
12.60.1 Detailed Description . . . 326
12.60.2 Member Typedef Documentation . . . 326
12.60.2.1 KeyValuePairArrayType . . . 326
12.60.2.2 KeyValuePairType . . . 327

12.60.3 Member Function Documentation . . . 327
12.60.3.1 CEcho() . . . 327
12.60.3.2 CFind() . . . 327
12.60.3.3 CMove() . . . 328
12.60.3.4 ConstructQuery() [1/2] . . . 329
12.60.3.5 ConstructQuery() [2/2] . . . 329
12.60.3.6 CStore() . . . 329
12.61 gdcm::ConstCharWrapper Class Reference . . . 330
12.61.1 Detailed Description . . . 330
12.61.2 Constructor & Destructor Documentation . . . 331
12.61.2.1 ConstCharWrapper() . . . 331
12.61.3 Member Function Documentation . . . 331
12.61.3.1 operator const char ∗() . . . 331
12.62 gdcm::CP246ExplicitDataElement Class Reference . . . 331
12.62.1 Detailed Description . . . 334
12.62.2 Member Function Documentation . . . 334
12.62.2.1 GetLength() . . . 334
12.62.2.2 Read() . . . 334
12.62.2.3 ReadPreValue() . . . 334
12.62.2.4 ReadValue() . . . 334
12.62.2.5 ReadWithLength() . . . 335
12.63 gdcm::CryptoFactory Class Reference . . . 335
12.63.1 Detailed Description . . . 336
12.63.2 Member Enumeration Documentation . . . 336
12.63.2.1 CryptoLib . . . 336
12.63.3 Constructor & Destructor Documentation . . . 336
12.63.3.1 CryptoFactory() [1/2] . . . 336
12.63.3.2 CryptoFactory() [2/2] . . . 337
12.63.3.3 ∼CryptoFactory() . . . 337
12.63.4 Member Function Documentation . . . 337
12.63.4.1 CreateCMSProvider() . . . 337
12.63.4.2 GetFactoryInstance() . . . 337
12.64 gdcm::CryptographicMessageSyntax Class Reference . . . 337
12.64.1 Member Enumeration Documentation . . . 338
12.64.1.1 CipherTypes . . . 338
12.64.2 Constructor & Destructor Documentation . . . 338
12.64.2.1 CryptographicMessageSyntax() [1/2] . . . 338
12.64.2.2 ∼CryptographicMessageSyntax() . . . 339
12.64.2.3 CryptographicMessageSyntax() [2/2] . . . 339

12.64.3 Member Function Documentation . . . 339
12.64.3.1 Decrypt() . . . 339
12.64.3.2 Encrypt() . . . 339
12.64.3.3 GetCipherType() . . . 339
12.64.3.4 operator=() . . . 340
12.64.3.5 ParseCertificateFile() . . . 340
12.64.3.6 ParseKeyFile() . . . 340
12.64.3.7 SetCipherType() . . . 340
12.64.3.8 SetPassword() . . . 340
12.65 gdcm::CSAElement Class Reference . . . 341
12.65.1 Detailed Description . . . 342
12.65.2 Member Typedef Documentation . . . 343
12.65.2.1 DataPtr . . . 343
12.65.3 Constructor & Destructor Documentation . . . 343
12.65.3.1 CSAElement() [1/2] . . . 343
12.65.3.2 CSAElement() [2/2] . . . 343
12.65.4 Member Function Documentation . . . 343
12.65.4.1 GetByteValue() . . . 343
12.65.4.2 GetKey() . . . 344
12.65.4.3 GetName() . . . 344
12.65.4.4 GetNoOfItems() . . . 344
12.65.4.5 GetSyngoDT() . . . 344
12.65.4.6 GetValue() [1/2] . . . 344
12.65.4.7 GetValue() [2/2] . . . 345
12.65.4.8 GetVM() . . . 345
12.65.4.9 GetVR() . . . 345
12.65.4.10 IsEmpty() . . . 345
12.65.4.11 operator<() . . . 345
12.65.4.12 operator=() . . . 346
12.65.4.13 operator==() . . . 346
12.65.4.14 SetByteValue() . . . 346
12.65.4.15 SetKey() . . . 346
12.65.4.16 SetName() . . . 346
12.65.4.17 SetNoOfItems() . . . 346
12.65.4.18 SetSyngoDT() . . . 347
12.65.4.19 SetValue() . . . 347
12.65.4.20 SetVM() . . . 347
12.65.4.21 SetVR() . . . 347
12.65.5 Friends And Related Symbol Documentation . . . 347

12.65.5.1 operator<< . . . 347
12.65.6 Member Data Documentation . . . 348
12.65.6.1 DataField . . . 348
12.65.6.2 KeyField . . . 348
12.65.6.3 NameField . . . 348
12.65.6.4 NoOfItemsField . . . 348
12.65.6.5 SyngoDTField . . . 348
12.65.6.6 ValueMultiplicityField . . . 348
12.65.6.7 VRField . . . 349
12.66 gdcm::CSAHeader Class Reference . . . 349
12.66.1 Detailed Description . . . 350
12.66.2 Member Enumeration Documentation . . . 351
12.66.2.1 CSAHeaderType . . . 351
12.66.3 Constructor & Destructor Documentation . . . 351
12.66.3.1 CSAHeader() . . . 351
12.66.3.2 ~CSAHeader() . . . 351
12.66.4 Member Function Documentation . . . 351
12.66.4.1 FindCSAElementByName() . . . 351
12.66.4.2 GetCSADataInfo() . . . 352
12.66.4.3 GetCSAEEnd() . . . 352
12.66.4.4 GetCSAElementByName() . . . 352
12.66.4.5 GetCSAImageHeaderInfoTag() . . . 352
12.66.4.6 GetCSASeriesHeaderInfoTag() . . . 352
12.66.4.7 GetDataSet() . . . 353
12.66.4.8 GetFormat() . . . 353
12.66.4.9 GetInterfile() . . . 353
12.66.4.10 GetMrProtocol() . . . 353
12.66.4.11 LoadFromDataElement() . . . 353
12.66.4.12 Print() . . . 354
12.66.5 Friends And Related Symbol Documentation . . . 354
12.66.5.1 operator<< . . . 354
12.67 gdcm::CSAHeaderDict Class Reference . . . 354
12.67.1 Detailed Description . . . 355
12.67.2 Member Typedef Documentation . . . 355
12.67.2.1 ConstIterator . . . 355
12.67.2.2 Iterator . . . 355
12.67.2.3 MapCSAHeaderDictEntry . . . 355
12.67.3 Constructor & Destructor Documentation . . . 355
12.67.3.1 CSAHeaderDict() [1/2] . . . 355

12.67.3.2 CSAHeaderDict() [2/2] . . . 356
12.67.4 Member Function Documentation . . . 356
12.67.4.1 AddCSAHeaderDictEntry() . . . 356
12.67.4.2 Begin() . . . 356
12.67.4.3 End() . . . 356
12.67.4.4 GetCSAHeaderDictEntry() . . . 356
12.67.4.5 IsEmpty() . . . 356
12.67.4.6 LoadDefault() . . . 356
12.67.4.7 operator=() . . . 357
12.67.5 Friends And Related Symbol Documentation . . . 357
12.67.5.1 Dicts . . . 357
12.67.5.2 operator<< . . . 357
12.68 gdcm::CSAHeaderDictEntry Class Reference . . . 357
12.68.1 Detailed Description . . . 358
12.68.2 Constructor & Destructor Documentation . . . 359
12.68.2.1 CSAHeaderDictEntry() . . . 359
12.68.3 Member Function Documentation . . . 359
12.68.3.1 GetDescription() . . . 359
12.68.3.2 GetName() . . . 359
12.68.3.3 GetVM() . . . 359
12.68.3.4 GetVR() . . . 359
12.68.3.5 operator<() . . . 360
12.68.3.6 SetDescription() . . . 360
12.68.3.7 SetName() . . . 360
12.68.3.8 SetVM() . . . 360
12.68.3.9 SetVR() . . . 360
12.68.4 Friends And Related Symbol Documentation . . . 360
12.68.4.1 operator<< . . . 360
12.69 gdcm::CSAHeaderDictException Class Reference . . . 361
12.70 gdcm::network::CStoreRQ Class Reference . . . 361
12.70.1 Detailed Description . . . 362
12.70.2 Member Function Documentation . . . 363
12.70.2.1 ConstructPDV() . . . 363
12.71 gdcm::network::CStoreRSP Class Reference . . . 363
12.71.1 Detailed Description . . . 364
12.71.2 Member Function Documentation . . . 364
12.71.2.1 ConstructPDV() . . . 364
12.72 gdcm::Curve Class Reference . . . 364
12.72.1 Detailed Description . . . 366

12.72.2 Constructor & Destructor Documentation . . . 366
12.72.2.1 Curve() [1/2] . . . 366
12.72.2.2 ∼Curve() . . . 366
12.72.2.3 Curve() [2/2] . . . 366
12.72.3 Member Function Documentation . . . 367
12.72.3.1 Decode() . . . 367
12.72.3.2 GetAsPoints() . . . 367
12.72.3.3 GetCurveDataDescriptor() . . . 367
12.72.3.4 GetDataValueRepresentation() . . . 367
12.72.3.5 GetDimensions() . . . 367
12.72.3.6 GetGroup() . . . 367
12.72.3.7 GetNumberOfCurves() . . . 367
12.72.3.8 GetNumberOfPoints() . . . 367
12.72.3.9 GetTypeOfData() . . . 368
12.72.3.10 GetTypeOfDataDescription() . . . 368
12.72.3.11 IsEmpty() . . . 368
12.72.3.12 Print() . . . 368
12.72.3.13 SetCoordinateStartValue() . . . 368
12.72.3.14 SetCoordinateStepValue() . . . 368
12.72.3.15 SetCurve() . . . 368
12.72.3.16 SetCurveDataDescriptor() . . . 368
12.72.3.17 SetCurveDescription() . . . 369
12.72.3.18 SetDataValueRepresentation() . . . 369
12.72.3.19 SetDimensions() . . . 369
12.72.3.20 SetGroup() . . . 369
12.72.3.21 SetNumberOfPoints() . . . 369
12.72.3.22 SetTypeOfData() . . . 369
12.72.3.23 Update() . . . 369
12.73 gdcm::DataElement Class Reference . . . 370
12.73.1 Detailed Description . . . 372
12.73.2 Member Typedef Documentation . . . 373
12.73.2.1 ValuePtr . . . 373
12.73.3 Constructor & Destructor Documentation . . . 373
12.73.3.1 DataElement() [1/2] . . . 373
12.73.3.2 DataElement() [2/2] . . . 373
12.73.4 Member Function Documentation . . . 373
12.73.4.1 Clear() . . . 373
12.73.4.2 Empty() . . . 373
12.73.4.3 GetByteValue() . . . 374

12.73.4.4 GetLength() . . . 374
12.73.4.5 GetSequenceOfFragments() [1/2] . . . 374
12.73.4.6 GetSequenceOfFragments() [2/2] . . . 374
12.73.4.7 GetTag() [1/2] . . . 375
12.73.4.8 GetTag() [2/2] . . . 375
12.73.4.9 GetValue() [1/2] . . . 375
12.73.4.10 GetValue() [2/2] . . . 375
12.73.4.11 GetValueAsSQ() . . . 376
12.73.4.12 GetVL() [1/2] . . . 376
12.73.4.13 GetVL() [2/2] . . . 376
12.73.4.14 GetVR() . . . 377
12.73.4.15 IsEmpty() . . . 377
12.73.4.16 IsUndefinedLength() . . . 377
12.73.4.17 operator<() . . . 378
12.73.4.18 operator=() . . . 378
12.73.4.19 operator==() . . . 378
12.73.4.20 Read() . . . 378
12.73.4.21 ReadOrSkip() . . . 378
12.73.4.22 ReadPreValue() . . . 379
12.73.4.23 ReadValue() . . . 379
12.73.4.24 ReadValueWithLength() . . . 379
12.73.4.25 ReadWithLength() . . . 379
12.73.4.26 SetByteValue() . . . 380
12.73.4.27 SetTag() . . . 380
12.73.4.28 SetValue() . . . 381
12.73.4.29 SetValueFieldLength() . . . 381
12.73.4.30 SetVL() . . . 381
12.73.4.31 SetVLToUndefined() . . . 381
12.73.4.32 SetVR() . . . 382
12.73.4.33 Write() . . . 382
12.73.5 Friends And Related Symbol Documentation . . . 382
12.73.5.1 operator<< . . . 382
12.73.6 Member Data Documentation . . . 383
12.73.6.1 TagField . . . 383
12.73.6.2 ValueField . . . 383
12.73.6.3 ValueLengthField . . . 383
12.73.6.4 VRField . . . 383
12.74 gdcm::DataElementException Class Reference . . . 384
12.75 gdcm::DataEvent Class Reference . . . 384

12.75.1 Detailed Description . . . 386
12.75.2 Member Typedef Documentation . . . 386
12.75.2.1 Self . . . 386
12.75.2.2 Superclass . . . 386
12.75.3 Constructor & Destructor Documentation . . . 386
12.75.3.1 DataEvent() [1/2] . . . 386
12.75.3.2 ~DataEvent() . . . 387
12.75.3.3 DataEvent() [2/2] . . . 387
12.75.4 Member Function Documentation . . . 387
12.75.4.1 CheckEvent() . . . 387
12.75.4.2 GetData() . . . 387
12.75.4.3 GetDataLength() . . . 387
12.75.4.4 GetEventName() . . . 387
12.75.4.5 MakeObject() . . . 387
12.75.4.6 operator=() . . . 388
12.75.4.7 SetData() . . . 388
12.76 gdcm::DataSet Class Reference . . . 388
12.76.1 Detailed Description . . . 390
12.76.2 Member Typedef Documentation . . . 391
12.76.2.1 ConstIterator . . . 391
12.76.2.2 DataElementSet . . . 391
12.76.2.3 Iterator . . . 391
12.76.2.4 SizeType . . . 391
12.76.3 Member Function Documentation . . . 392
12.76.3.1 Begin() [1/2] . . . 392
12.76.3.2 Begin() [2/2] . . . 392
12.76.3.3 Clear() . . . 392
12.76.3.4 ComputeDataElement() . . . 392
12.76.3.5 ComputeGroupLength() . . . 392
12.76.3.6 End() [1/2] . . . 392
12.76.3.7 End() [2/2] . . . 393
12.76.3.8 FindDataElement() [1/2] . . . 393
12.76.3.9 FindDataElement() [2/2] . . . 393
12.76.3.10 FindNextDataElement() . . . 393
12.76.3.11 GetDataElement() [1/2] . . . 394
12.76.3.12 GetDataElement() [2/2] . . . 394
12.76.3.13 GetDEEnd() . . . 394
12.76.3.14 GetDES() [1/2] . . . 394
12.76.3.15 GetDES() [2/2] . . . 395

12.76.3.16 GetLength() . . . 395
12.76.3.17 GetMediaStorage() . . . 395
12.76.3.18 GetPrivateCreator() . . . 395
12.76.3.19 GetPrivateTag() . . . 395
12.76.3.20 Insert() . . . 396
12.76.3.21 InsertDataElement() . . . 396
12.76.3.22 IsEmpty() . . . 396
12.76.3.23 operator()() . . . 396
12.76.3.24 operator=() . . . 397
12.76.3.25 operator[]() . . . 397
12.76.3.26 Print() . . . 397
12.76.3.27 Read() . . . 397
12.76.3.28 ReadNested() . . . 397
12.76.3.29 ReadSelectedPrivateTags() . . . 397
12.76.3.30 ReadSelectedPrivateTagsWithLength() . . . 398
12.76.3.31 ReadSelectedTags() . . . 398
12.76.3.32 ReadSelectedTagsWithLength() . . . 398
12.76.3.33 ReadUpToTag() . . . 398
12.76.3.34 ReadUpToTagWithLength() . . . 398
12.76.3.35 ReadWithLength() . . . 399
12.76.3.36 Remove() . . . 399
12.76.3.37 Replace() . . . 399
12.76.3.38 ReplaceEmpty() . . . 399
12.76.3.39 Size() . . . 400
12.76.3.40 Write() . . . 400
12.76.4 Friends And Related Symbol Documentation . . . 400
12.76.4.1 CSAHeader . . . 400
12.76.4.2 operator<< . . . 400
12.77 gdcm::DataSetEvent Class Reference . . . 401
12.77.1 Detailed Description . . . 402
12.77.2 Member Typedef Documentation . . . 402
12.77.2.1 Self . . . 402
12.77.2.2 Superclass . . . 402
12.77.3 Constructor & Destructor Documentation . . . 403
12.77.3.1 DataSetEvent() [1/2] . . . 403
12.77.3.2 ~DataSetEvent() . . . 403
12.77.3.3 DataSetEvent() [2/2] . . . 403
12.77.4 Member Function Documentation . . . 403
12.77.4.1 CheckEvent() . . . 403

12.77.4.2 GetDataSet() . . . 403
12.77.4.3 GetEventName() . . . 403
12.77.4.4 MakeObject() . . . 404
12.77.4.5 operator=() . . . 404
12.77.5 Member Data Documentation . . . 404
12.77.5.1 m_DataSet . . . 404
12.78 gdcm::DataSetHelper Class Reference . . . 404
12.78.1 Detailed Description . . . 404
12.78.2 Member Function Documentation . . . 405
12.78.2.1 ComputeVR() . . . 405
12.79 gdcm::Decoder Class Reference . . . 405
12.79.1 Detailed Description . . . 406
12.79.2 Constructor & Destructor Documentation . . . 406
12.79.2.1 ∼Decoder() . . . 406
12.79.3 Member Function Documentation . . . 406
12.79.3.1 CanDecode() . . . 406
12.79.3.2 Decode() . . . 406
12.79.3.3 DecodeByStreams() . . . 407
12.80 gdcm::DefinedTerms Class Reference . . . 407
12.80.1 Detailed Description . . . 407
12.80.2 Constructor & Destructor Documentation . . . 407
12.80.2.1 DefinedTerms() . . . 407
12.81 gdcm::Defs Class Reference . . . 408
12.81.1 Detailed Description . . . 408
12.81.2 Constructor & Destructor Documentation . . . 409
12.81.2.1 Defs() [1/2] . . . 409
12.81.2.2 ∼Defs() . . . 409
12.81.2.3 Defs() [2/2] . . . 409
12.81.3 Member Function Documentation . . . 409
12.81.3.1 GetIODFromFile() . . . 409
12.81.3.2 GetIODNameFromMediaStorage() . . . 409
12.81.3.3 GetIODs() [1/2] . . . 409
12.81.3.4 GetIODs() [2/2] . . . 410
12.81.3.5 GetMacros() [1/2] . . . 410
12.81.3.6 GetMacros() [2/2] . . . 410
12.81.3.7 GetModules() [1/2] . . . 410
12.81.3.8 GetModules() [2/2] . . . 410
12.81.3.9 GetTypeFromTag() . . . 410
12.81.3.10 IsEmpty() . . . 411

12.81.3.11 LoadDefaults() . . . 411
12.81.3.12 LoadFromFile() . . . 411
12.81.3.13 operator=() . . . 411
12.81.3.14 Verify() [1/2] . . . 411
12.81.3.15 Verify() [2/2] . . . 411
12.81.4 Friends And Related Symbol Documentation . . . 411
12.81.4.1 Global . . . 411
12.82 gdcm::DeltaEncodingCodec Class Reference . . . 412
12.82.1 Detailed Description . . . 414
12.82.2 Constructor & Destructor Documentation . . . 415
12.82.2.1 DeltaEncodingCodec() . . . 415
12.82.2.2 ∼DeltaEncodingCodec() . . . 415
12.82.3 Member Function Documentation . . . 415
12.82.3.1 CanDecode() . . . 415
12.82.3.2 Decode() [1/2] . . . 415
12.82.3.3 Decode() [2/2] . . . 415
12.83 gdcm::DICOMDIR Class Reference . . . 415
12.83.1 Detailed Description . . . 416
12.83.2 Constructor & Destructor Documentation . . . 416
12.83.2.1 DICOMDIR() [1/2] . . . 416
12.83.2.2 DICOMDIR() [2/2] . . . 416
12.84 gdcm::DICOMDIRGenerator Class Reference . . . 416
12.84.1 Detailed Description . . . 417
12.84.2 Member Typedef Documentation . . . 418
12.84.2.1 FilenamesType . . . 418
12.84.2.2 FilenameType . . . 418
12.84.3 Constructor & Destructor Documentation . . . 418
12.84.3.1 DICOMDIRGenerator() . . . 418
12.84.3.2 ∼DICOMDIRGenerator() . . . 418
12.84.4 Member Function Documentation . . . 418
12.84.4.1 AddImageDirectoryRecord() . . . 418
12.84.4.2 AddPatientDirectoryRecord() . . . 418
12.84.4.3 AddSeriesDirectoryRecord() . . . 418
12.84.4.4 AddStudyDirectoryRecord() . . . 418
12.84.4.5 Generate() . . . 419
12.84.4.6 GetFile() . . . 419
12.84.4.7 GetScanner() . . . 419
12.84.4.8 SetDescriptor() . . . 419
12.84.4.9 SetFile() . . . 419

12.84.4.10 SetFilenames() . . . 420
12.84.4.11 SetRootDirectory() . . . 420
12.85 gdcm::Dict Class Reference . . . 420
12.85.1 Detailed Description . . . 421
12.85.2 Member Typedef Documentation . . . 421
12.85.2.1 ConstIterator . . . 421
12.85.2.2 Iterator . . . 421
12.85.2.3 MapDictEntry . . . 421
12.85.3 Constructor & Destructor Documentation . . . 421
12.85.3.1 Dict() [1/2] . . . 421
12.85.3.2 Dict() [2/2] . . . 422
12.85.4 Member Function Documentation . . . 422
12.85.4.1 AddDictEntry() . . . 422
12.85.4.2 Begin() . . . 422
12.85.4.3 End() . . . 422
12.85.4.4 GetDictEntry() . . . 422
12.85.4.5 GetDictEntryByKeyword() . . . 423
12.85.4.6 GetDictEntryByName() . . . 423
12.85.4.7 GetKeywordFromTag() . . . 423
12.85.4.8 IsEmpty() . . . 423
12.85.4.9 LoadDefault() . . . 423
12.85.4.10 operator=() . . . 423
12.85.5 Friends And Related Symbol Documentation . . . 424
12.85.5.1 Dicts . . . 424
12.85.5.2 operator<< . . . 424
12.86 gdcm::DictConverter Class Reference . . . 424
12.86.1 Detailed Description . . . 425
12.86.2 Member Enumeration Documentation . . . 425
12.86.2.1 OutputTypes . . . 425
12.86.3 Constructor & Destructor Documentation . . . 425
12.86.3.1 DictConverter() . . . 425
12.86.3.2 ~DictConverter() . . . 426
12.86.4 Member Function Documentation . . . 426
12.86.4.1 AddGroupLength() . . . 426
12.86.4.2 Convert() . . . 426
12.86.4.3 ConvertToCXX() . . . 426
12.86.4.4 ConvertToXML() . . . 426
12.86.4.5 GetDictName() . . . 426
12.86.4.6 GetInputFilename() . . . 426

12.86.4.7 GetOutputFilename() . . . 426
12.86.4.8 GetOutputType() . . . 427
12.86.4.9 Readuint16() . . . 427
12.86.4.10 ReadVM() . . . 427
12.86.4.11 ReadVR() . . . 427
12.86.4.12 SetDictName() . . . 427
12.86.4.13 SetInputFileName() . . . 427
12.86.4.14 SetOutputFileName() . . . 427
12.86.4.15 SetOutputType() . . . 427
12.86.4.16 WriteFooter() . . . 428
12.86.4.17 WriteHeader() . . . 428
12.87 gdcm::DictEntry Class Reference . . . 428
12.87.1 Detailed Description . . . 429
12.87.2 Constructor & Destructor Documentation . . . 429
12.87.2.1 DictEntry() . . . 429
12.87.3 Member Function Documentation . . . 429
12.87.3.1 GetKeyword() . . . 429
12.87.3.2 GetName() . . . 430
12.87.3.3 GetRetired() . . . 430
12.87.3.4 GetVM() . . . 430
12.87.3.5 GetVR() . . . 430
12.87.3.6 IsUnique() . . . 430
12.87.3.7 SetElementXX() . . . 431
12.87.3.8 SetGroupXX() . . . 431
12.87.3.9 SetKeyword() . . . 431
12.87.3.10 SetName() . . . 431
12.87.3.11 SetRetired() . . . 431
12.87.3.12 SetVM() . . . 431
12.87.3.13 SetVR() . . . 431
12.87.4 Friends And Related Symbol Documentation . . . 432
12.87.4.1 Dict . . . 432
12.87.4.2 operator<< . . . 432
12.88 gdcm::DictPrinter Class Reference . . . 432
12.88.1 Detailed Description . . . 434
12.88.2 Constructor & Destructor Documentation . . . 434
12.88.2.1 DictPrinter() . . . 434
12.88.2.2 ~DictPrinter() . . . 434
12.88.3 Member Function Documentation . . . 435
12.88.3.1 Print() . . . 435

12.88.3.2 PrintDataElement2() . . . 435
12.88.3.3 PrintDataSet2() . . . 435
12.89 gdcm::Dicts Class Reference . . . 435
12.89.1 Detailed Description . . . 436
12.89.2 Member Enumeration Documentation . . . 436
12.89.2.1 ConstructorType . . . 436
12.89.3 Constructor & Destructor Documentation . . . 437
12.89.3.1 Dicts() [1/2] . . . 437
12.89.3.2 ∼Dicts() . . . 437
12.89.3.3 Dicts() [2/2] . . . 437
12.89.4 Member Function Documentation . . . 437
12.89.4.1 GetConstructorString() . . . 437
12.89.4.2 GetCSAHeaderDict() . . . 437
12.89.4.3 GetDictEntry() [1/2] . . . 437
12.89.4.4 GetDictEntry() [2/2] . . . 438
12.89.4.5 GetPrivateDict() [1/2] . . . 438
12.89.4.6 GetPrivateDict() [2/2] . . . 438
12.89.4.7 GetPublicDict() . . . 438
12.89.4.8 IsEmpty() . . . 438
12.89.4.9 LoadDefaults() . . . 439
12.89.4.10 operator=() . . . 439
12.89.5 Friends And Related Symbol Documentation . . . 439
12.89.5.1 Global . . . 439
12.89.5.2 operator<< . . . 439
12.90 gdcm::network::DIMSE Class Reference . . . 439
12.90.1 Detailed Description . . . 440
12.90.2 Member Enumeration Documentation . . . 440
12.90.2.1 CommandTypes . . . 440
12.91 gdcm::DirectionCosines Class Reference . . . 441
12.91.1 Detailed Description . . . 442
12.91.2 Constructor & Destructor Documentation . . . 442
12.91.2.1 DirectionCosines() [1/2] . . . 442
12.91.2.2 DirectionCosines() [2/2] . . . 442
12.91.2.3 ∼DirectionCosines() . . . 443
12.91.3 Member Function Documentation . . . 443
12.91.3.1 ComputeDistAlongNormal() . . . 443
12.91.3.2 Cross() . . . 443
12.91.3.3 CrossDot() . . . 443
12.91.3.4 Dot() [1/2] . . . 443

12.91.3.5 Dot() [2/2] . . . 443
12.91.3.6 IsValid() . . . 444
12.91.3.7 Norm() . . . 444
12.91.3.8 Normalize() [1/2] . . . 444
12.91.3.9 Normalize() [2/2] . . . 444
12.91.3.10 operator const double ∗() . . . 444
12.91.3.11 Print() . . . 444
12.91.3.12 SetFromString() . . . 445
12.92 gdcm::Directory Class Reference . . . 445
12.92.1 Detailed Description . . . 446
12.92.2 Member Typedef Documentation . . . 446
12.92.2.1 FilenamesType . . . 446
12.92.2.2 FilenameType . . . 446
12.92.3 Constructor & Destructor Documentation . . . 446
12.92.3.1 Directory() . . . 446
12.92.3.2 ∼Directory() . . . 447
12.92.4 Member Function Documentation . . . 447
12.92.4.1 Explore() . . . 447
12.92.4.2 GetDirectories() . . . 447
12.92.4.3 GetFilenames() . . . 447
12.92.4.4 GetToplevel() . . . 447
12.92.4.5 Load() . . . 448
12.92.4.6 Print() . . . 448
12.92.5 Friends And Related Symbol Documentation . . . 448
12.92.5.1 operator<< . . . 448
12.93 gdcm::DirectoryHelper Class Reference . . . 449
12.93.1 Detailed Description . . . 449
12.93.2 Member Function Documentation . . . 449
12.93.2.1 GetCTImageSeriesUIDs() . . . 449
12.93.2.2 GetFilenamesFromSeriesUIDs() . . . 449
12.93.2.3 GetFrameOfReference() . . . 450
12.93.2.4 GetMRImageSeriesUIDs() . . . 450
12.93.2.5 GetRTStructSeriesUIDs() . . . 450
12.93.2.6 GetSeriesUIDsBySOPClassUID() . . . 450
12.93.2.7 GetSOPClassUID() . . . 450
12.93.2.8 GetStringValueFromTag() . . . 450
12.93.2.9 LoadImageFromFiles() . . . 450
12.93.2.10 RetrieveSOPInstanceUIDFromIndex() . . . 451
12.93.2.11 RetrieveSOPInstanceUIDFromZPosition() . . . 451

12.94 gdcm::DPath Class Reference . . . 451
12.94.1 Detailed Description . . . 452
12.94.2 Constructor & Destructor Documentation . . . 452
12.94.2.1 DPath() . . . 452
12.94.2.2 ∼DPath() . . . 452
12.94.3 Member Function Documentation . . . 452
12.94.3.1 ConstructFromString() . . . 452
12.94.3.2 IsValid() . . . 452
12.94.3.3 Match() . . . 452
12.94.3.4 operator<() . . . 453
12.94.3.5 Print() . . . 453
12.94.4 Friends And Related Symbol Documentation . . . 453
12.94.4.1 operator<< . . . 453
12.95 gdcm::DummyValueGenerator Class Reference . . . 453
12.95.1 Detailed Description . . . 453
12.95.2 Member Function Documentation . . . 454
12.95.2.1 Generate() . . . 454
12.96 gdcm::Dumper Class Reference . . . 454
12.96.1 Detailed Description . . . 456
12.96.2 Constructor & Destructor Documentation . . . 456
12.96.2.1 Dumper() . . . 456
12.96.2.2 ∼Dumper() . . . 456
12.97 gdcm::Element< TVR, TVM > Class Template Reference . . . 457
12.97.1 Detailed Description . . . 459
12.97.2 Member Typedef Documentation . . . 459
12.97.2.1 Type . . . 459
12.97.3 Member Function Documentation . . . 459
12.97.3.1 GetAsDataElement() . . . 459
12.97.3.2 GetLength() . . . 460
12.97.3.3 GetValue() [1/2] . . . 460
12.97.3.4 GetValue() [2/2] . . . 460
12.97.3.5 GetValues() . . . 460
12.97.3.6 GetVM() . . . 460
12.97.3.7 GetVR() . . . 461
12.97.3.8 operator[]() . . . 461
12.97.3.9 Print() . . . 461
12.97.3.10 Read() . . . 461
12.97.3.11 Set() . . . 461
12.97.3.12 SetFromDataElement() . . . 462

12.97.3.13 SetNoSwap() . . . 462
12.97.3.14 SetValue() . . . 462
12.97.3.15 Write() . . . 462
12.97.4 Member Data Documentation . . . 463
12.97.4.1 Internal . . . 463
12.98 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference . . . 463
12.98.1 Member Typedef Documentation . . . 466
12.98.1.1 Parent . . . 466
12.98.1.2 Type . . . 466
12.98.2 Member Function Documentation . . . 466
12.98.2.1 GetAsDataElement() . . . 466
12.98.2.2 GetLength() . . . 466
12.98.2.3 GetValue() . . . 466
12.98.2.4 GetValues() . . . 466
12.98.2.5 GetVM() . . . 467
12.98.2.6 GetVR() . . . 467
12.98.2.7 operator[]() . . . 467
12.98.2.8 Print() . . . 467
12.98.2.9 Read() . . . 467
12.98.2.10 Set() . . . 467
12.98.2.11 SetFromDataElement() . . . 467
12.98.2.12 SetLength() . . . 467
12.98.2.13 SetNoSwap() . . . 468
12.98.2.14 SetValue() . . . 468
12.98.2.15 Write() . . . 468
12.98.3 Member Data Documentation . . . 468
12.98.3.1 Internal . . . 468
12.99 gdcm::Element< TVR, VM::VM1_n > Class Template Reference . . . 468
12.99.1 Member Typedef Documentation . . . 470
12.99.1.1 Type . . . 470
12.99.2 Constructor & Destructor Documentation . . . 470
12.99.2.1 Element() [1/2] . . . 470
12.99.2.2 ~Element() . . . 470
12.99.2.3 Element() [2/2] . . . 470
12.99.3 Member Function Documentation . . . 471
12.99.3.1 GetAsDataElement() . . . 471
12.99.3.2 GetLength() . . . 471
12.99.3.3 GetValue() [1/2] . . . 471
12.99.3.4 GetValue() [2/2] . . . 471

12.99.3.5 GetValues() . . . 471
12.99.3.6 GetVM() . . . 471
12.99.3.7 GetVR() . . . 472
12.99.3.8 operator=() . . . 472
12.99.3.9 operator[]() . . . 472
12.99.3.10 Print() . . . 472
12.99.3.11 Read() . . . 472
12.99.3.12 Set() . . . 473
12.99.3.13 SetArray() . . . 473
12.99.3.14 SetFromDataElement() . . . 473
12.99.3.15 SetLength() . . . 473
12.99.3.16 SetNoSwap() . . . 473
12.99.3.17 SetValue() . . . 474
12.99.3.18 Write() . . . 474
12.99.3.19 WriteASCII() . . . 474
12.100 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference . . . 474
12.100.1 Member Typedef Documentation . . . 478
12.100.1.1 Parent . . . 478
12.100.1.2 Type . . . 479
12.100.2 Member Function Documentation . . . 479
12.100.2.1 GetAsDataElement() . . . 479
12.100.2.2 GetLength() . . . 479
12.100.2.3 GetValue() . . . 479
12.100.2.4 GetValues() . . . 479
12.100.2.5 GetVM() . . . 479
12.100.2.6 GetVR() . . . 479
12.100.2.7 operator[]() . . . 479
12.100.2.8 Print() . . . 479
12.100.2.9 Read() . . . 480
12.100.2.10 Set() . . . 480
12.100.2.11 SetFromDataElement() . . . 480
12.100.2.12 SetLength() . . . 480
12.100.2.13 SetNoSwap() . . . 480
12.100.2.14 SetValue() . . . 480
12.100.2.15 Write() . . . 480
12.100.3 Member Data Documentation . . . 481
12.100.3.1 Internal . . . 481
12.101 gdcm::Element< TVR, VM::VM2_n > Class Template Reference . . . 481
12.101.1 Member Typedef Documentation . . . 484

12.101.1.1 Parent . . . 484
12.101.1.2 Type . . . 484
12.101.2 Member Function Documentation . . . 484
12.101.2.1 GetAsDataElement() . . . 484
12.101.2.2 GetLength() . . . 484
12.101.2.3 GetValue() . . . 484
12.101.2.4 GetValues() . . . 484
12.101.2.5 GetVM() . . . 485
12.101.2.6 GetVR() . . . 485
12.101.2.7 operator[]() . . . 485
12.101.2.8 Print() . . . 485
12.101.2.9 Read() . . . 485
12.101.2.10 Set() . . . 485
12.101.2.11 SetFromDataElement() . . . 485
12.101.2.12 SetLength() . . . 485
12.101.2.13 SetNoSwap() . . . 486
12.101.2.14 SetValue() . . . 486
12.101.2.15 Write() . . . 486
12.101.3 Member Data Documentation . . . 486
12.101.3.1 Internal . . . 486
12.102 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference . . . 486
12.102.1 Member Typedef Documentation . . . 490
12.102.1.1 Parent . . . 490
12.102.1.2 Type . . . 491
12.102.2 Member Function Documentation . . . 491
12.102.2.1 GetAsDataElement() . . . 491
12.102.2.2 GetLength() . . . 491
12.102.2.3 GetValue() . . . 491
12.102.2.4 GetValues() . . . 491
12.102.2.5 GetVM() . . . 491
12.102.2.6 GetVR() . . . 491
12.102.2.7 operator[]() . . . 491
12.102.2.8 Print() . . . 491
12.102.2.9 Read() . . . 492
12.102.2.10 Set() . . . 492
12.102.2.11 SetFromDataElement() . . . 492
12.102.2.12 SetLength() . . . 492
12.102.2.13 SetNoSwap() . . . 492
12.102.2.14 SetValue() . . . 492

12.102.2.15 Write() . . . 492
12.102.3 Member Data Documentation . . . 493
12.102.3.1 Internal . . . 493
12.103 gdcm::Element< TVR, VM::VM3_4 > Class Template Reference . . . 493
12.103.1 Member Typedef Documentation . . . 496
12.103.1.1 Parent . . . 496
12.103.1.2 Type . . . 496
12.103.2 Member Function Documentation . . . 496
12.103.2.1 GetAsDataElement() . . . 496
12.103.2.2 GetLength() . . . 496
12.103.2.3 GetValue() . . . 496
12.103.2.4 GetValues() . . . 496
12.103.2.5 GetVM() . . . 497
12.103.2.6 GetVR() . . . 497
12.103.2.7 operator[]() . . . 497
12.103.2.8 Print() . . . 497
12.103.2.9 Read() . . . 497
12.103.2.10 Set() . . . 497
12.103.2.11 SetFromDataElement() . . . 497
12.103.2.12 SetLength() . . . 497
12.103.2.13 SetNoSwap() . . . 498
12.103.2.14 SetValue() . . . 498
12.103.2.15 Write() . . . 498
12.103.3 Member Data Documentation . . . 498
12.103.3.1 Internal . . . 498
12.104 gdcm::Element< TVR, VM::VM3_n > Class Template Reference . . . 498
12.104.1 Member Typedef Documentation . . . 502
12.104.1.1 Parent . . . 502
12.104.1.2 Type . . . 502
12.104.2 Member Function Documentation . . . 502
12.104.2.1 GetAsDataElement() . . . 502
12.104.2.2 GetLength() . . . 502
12.104.2.3 GetValue() . . . 502
12.104.2.4 GetValues() . . . 502
12.104.2.5 GetVM() . . . 503
12.104.2.6 GetVR() . . . 503
12.104.2.7 operator[]() . . . 503
12.104.2.8 Print() . . . 503
12.104.2.9 Read() . . . 503

12.104.2.10 Set() . . . 503
12.104.2.11 SetFromDataElement() . . . 503
12.104.2.12 SetLength() . . . 503
12.104.2.13 SetNoSwap() . . . 504
12.104.2.14 SetValue() . . . 504
12.104.2.15 Write() . . . 504
12.104.3 Member Data Documentation . . . 504
12.104.3.1 Internal . . . 504
12.105 gdcm::Element< VR::AS, VM::VM5 > Class Reference . . . 504
12.105.1 Member Typedef Documentation . . . 506
12.105.1.1 Type . . . 506
12.105.2 Member Function Documentation . . . 506
12.105.2.1 GetAsDataElement() . . . 506
12.105.2.2 GetLength() . . . 506
12.105.2.3 GetValue() . . . 506
12.105.2.4 GetValues() . . . 506
12.105.2.5 GetVM() . . . 507
12.105.2.6 GetVR() . . . 507
12.105.2.7 operator[]() . . . 507
12.105.2.8 Print() . . . 507
12.105.2.9 Read() . . . 507
12.105.2.10 Set() . . . 507
12.105.2.11 SetFromDataElement() . . . 507
12.105.2.12 SetNoSwap() . . . 507
12.105.2.13 SetValue() . . . 508
12.105.2.14 Write() . . . 508
12.105.3 Member Data Documentation . . . 508
12.105.3.1 Internal . . . 508
12.106 gdcm::Element< VR::OB, VM::VM1 > Class Reference . . . 508
12.106.1 Member Typedef Documentation . . . 511
12.106.1.1 Type . . . 511
12.106.2 Member Function Documentation . . . 511
12.106.2.1 GetAsDataElement() . . . 511
12.106.2.2 GetLength() . . . 511
12.106.2.3 GetValue() . . . 511
12.106.2.4 GetValues() . . . 511
12.106.2.5 GetVM() . . . 511
12.106.2.6 GetVR() . . . 511
12.106.2.7 operator[]() . . . 512

12.106.2.8 Print() . . . 512
12.106.2.9 Read() . . . 512
12.106.2.10 Set() . . . 512
12.106.2.11 SetFromDataElement() . . . 512
12.106.2.12 SetNoSwap() . . . 512
12.106.2.13 SetValue() . . . 512
12.106.2.14 Write() . . . 512
12.106.3 Member Data Documentation . . . 513
12.106.3.1 Internal . . . 513
12.107 gdcm::Element< VR::OW, VM::VM1 > Class Reference . . . 513
12.107.1 Member Typedef Documentation . . . 516
12.107.1.1 Type . . . 516
12.107.2 Member Function Documentation . . . 516
12.107.2.1 GetAsDataElement() . . . 516
12.107.2.2 GetLength() . . . 516
12.107.2.3 GetValue() . . . 516
12.107.2.4 GetValues() . . . 516
12.107.2.5 GetVM() . . . 516
12.107.2.6 GetVR() . . . 516
12.107.2.7 operator[]() . . . 517
12.107.2.8 Print() . . . 517
12.107.2.9 Read() . . . 517
12.107.2.10 Set() . . . 517
12.107.2.11 SetFromDataElement() . . . 517
12.107.2.12 SetNoSwap() . . . 517
12.107.2.13 SetValue() . . . 517
12.107.2.14 Write() . . . 517
12.107.3 Member Data Documentation . . . 518
12.107.3.1 Internal . . . 518
12.108 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference . . . 518
12.108.1 Detailed Description . . . 518
12.109 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Reference . . . 519
12.110 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Reference . . . 520
12.111 gdcm::EmptyMaskGenerator Class Reference . . . 521
12.111.1 Detailed Description . . . 521
12.111.2 Member Enumeration Documentation . . . 522
12.111.2.1 SOPClassUIDMode . . . 522
12.111.3 Constructor & Destructor Documentation . . . 522
12.111.3.1 EmptyMaskGenerator() . . . 522

12.111.3.2 ∼EmptyMaskGenerator() . . . 522
12.111.4 Member Function Documentation . . . 522
12.111.4.1 Execute() . . . 522
12.111.4.2 SetInputDirectory() . . . 522
12.111.4.3 SetOutputDirectory() . . . 523
12.111.4.4 SetSOPClassUIDMode() . . . 523
12.112 gdcm::EncapsulatedDocument Class Reference . . . 523
12.112.1 Detailed Description . . . 523
12.112.2 Constructor & Destructor Documentation . . . 524
12.112.2.1 EncapsulatedDocument() . . . 524
12.113 gdcm::EncodingImplementation< T > Class Template Reference . . . 524
12.113.1 Detailed Description . . . 524
12.114 gdcm::EncodingImplementation< VR::VRASCII > Class Reference . . . 525
12.114.1 Member Function Documentation . . . 526
12.114.1.1 Read() . . . 526
12.114.1.2 ReadComputeLength() . . . 526
12.114.1.3 ReadNoSwap() . . . 526
12.114.1.4 Write() [1/2] . . . 527
12.114.1.5 Write() [2/2] . . . 527
12.115 gdcm::EncodingImplementation< VR::VRBINARY > Class Reference . . . 527
12.115.1 Member Function Documentation . . . 528
12.115.1.1 Read() . . . 528
12.115.1.2 ReadComputeLength() . . . 528
12.115.1.3 ReadNoSwap() . . . 529
12.115.1.4 Write() . . . 529
12.116 gdcm::EndEvent Class Reference . . . 529
12.117 gdcm::EnumeratedValues Class Reference . . . 530
12.117.1 Detailed Description . . . 531
12.117.2 Constructor & Destructor Documentation . . . 531
12.117.2.1 EnumeratedValues() . . . 531
12.118 gdcm::EquipmentManufacturer Class Reference . . . 531
12.118.1 Detailed Description . . . 532
12.118.2 Member Enumeration Documentation . . . 532
12.118.2.1 Type . . . 532
12.118.3 Member Function Documentation . . . 532
12.118.3.1 Compute() . . . 532
12.118.3.2 TypeToString() . . . 532
12.119 gdcm::Event Class Reference . . . 533
12.119.1 Detailed Description . . . 534

12.119.2 Constructor & Destructor Documentation . . . 534
12.119.2.1 Event() [1/2] . . . 534
12.119.2.2 ~Event() . . . 534
12.119.2.3 Event() [2/2] . . . 534
12.119.3 Member Function Documentation . . . 534
12.119.3.1 CheckEvent() . . . 534
12.119.3.2 GetEventName() . . . 535
12.119.3.3 MakeObject() . . . 535
12.119.3.4 operator=() . . . 535
12.119.3.5 Print() . . . 535
12.120 gdcm::Exception Class Reference . . . 536
12.120.1 Detailed Description . . . 537
12.120.2 Constructor & Destructor Documentation . . . 537
12.120.2.1 Exception() . . . 537
12.120.2.2 ~Exception() . . . 537
12.120.3 Member Function Documentation . . . 537
12.120.3.1 GetDescription() . . . 537
12.120.3.2 what() . . . 538
12.121 gdcm::ExitEvent Class Reference . . . 538
12.122 gdcm::ExplicitDataElement Class Reference . . . 539
12.122.1 Detailed Description . . . 542
12.122.2 Member Function Documentation . . . 542
12.122.2.1 GetLength() . . . 542
12.122.2.2 Read() . . . 542
12.122.2.3 ReadPreValue() . . . 543
12.122.2.4 ReadValue() . . . 543
12.122.2.5 ReadWithLength() . . . 543
12.122.2.6 Write() . . . 543
12.123 gdcm::ExplicitImplicitDataElement Class Reference . . . 543
12.123.1 Detailed Description . . . 546
12.123.2 Member Function Documentation . . . 546
12.123.2.1 GetLength() . . . 546
12.123.2.2 Read() . . . 546
12.123.2.3 ReadPreValue() . . . 547
12.123.2.4 ReadValue() . . . 547
12.123.2.5 ReadWithLength() . . . 547
12.124 gdcm::Fiducials Class Reference . . . 547
12.124.1 Detailed Description . . . 547
12.124.2 Constructor & Destructor Documentation . . . 548

12.124.2.1 Fiducials() . . . 548
12.125 gdcm::File Class Reference . . . 548
12.125.1 Detailed Description . . . 550
12.125.2 Constructor & Destructor Documentation . . . 550
12.125.2.1 File() . . . 550
12.125.2.2 ∼File() . . . 550
12.125.3 Member Function Documentation . . . 550
12.125.3.1 GetDataSet() [1/2] . . . 550
12.125.3.2 GetDataSet() [2/2] . . . 551
12.125.3.3 GetHeader() [1/2] . . . 551
12.125.3.4 GetHeader() [2/2] . . . 551
12.125.3.5 Read() . . . 551
12.125.3.6 SetDataSet() . . . 552
12.125.3.7 SetHeader() . . . 552
12.125.3.8 Write() . . . 552
12.125.4 Friends And Related Symbol Documentation . . . 552
12.125.4.1 operator<< . . . 552
12.126 gdcm::FileAnonymizer Class Reference . . . 553
12.126.1 Detailed Description . . . 555
12.126.2 Constructor & Destructor Documentation . . . 555
12.126.2.1 FileAnonymizer() . . . 555
12.126.2.2 ∼FileAnonymizer() . . . 555
12.126.3 Member Function Documentation . . . 555
12.126.3.1 Empty() . . . 555
12.126.3.2 Remove() . . . 556
12.126.3.3 Replace() [1/2] . . . 556
12.126.3.4 Replace() [2/2] . . . 556
12.126.3.5 SetInputFileName() . . . 556
12.126.3.6 SetOutputFileName() . . . 557
12.126.3.7 Write() . . . 557
12.127 gdcm::FileChangeTransferSyntax Class Reference . . . 557
12.127.1 Detailed Description . . . 559
12.127.2 Constructor & Destructor Documentation . . . 559
12.127.2.1 FileChangeTransferSyntax() . . . 559
12.127.2.2 ∼FileChangeTransferSyntax() . . . 560
12.127.3 Member Function Documentation . . . 560
12.127.3.1 Change() . . . 560
12.127.3.2 GetCodec() . . . 560
12.127.3.3 New() . . . 560

12.127.3.4 SetInputFileName() . . . 560
12.127.3.5 SetOutputFileName() . . . 561
12.127.3.6 SetTransferSyntax() . . . 561
12.128 gdcm::FileDecompressLookupTable Class Reference . . . 561
12.128.1 Detailed Description . . . 563
12.128.2 Constructor & Destructor Documentation . . . 563
12.128.2.1 FileDecompressLookupTable() . . . 563
12.128.2.2 ∼FileDecompressLookupTable() . . . 563
12.128.3 Member Function Documentation . . . 563
12.128.3.1 Change() . . . 563
12.128.3.2 GetFile() . . . 564
12.128.3.3 GetPixmap() [1/2] . . . 564
12.128.3.4 GetPixmap() [2/2] . . . 564
12.128.3.5 SetFile() . . . 564
12.128.3.6 SetPixmap() . . . 564
12.129 gdcm::FileDerivation Class Reference . . . 564
12.129.1 Detailed Description . . . 565
12.129.2 Constructor & Destructor Documentation . . . 565
12.129.2.1 FileDerivation() . . . 565
12.129.2.2 ∼FileDerivation() . . . 566
12.129.3 Member Function Documentation . . . 566
12.129.3.1 AddDerivationDescription() . . . 566
12.129.3.2 AddPurposeOfReferenceCodeSequence() . . . 566
12.129.3.3 AddReference() . . . 566
12.129.3.4 AddSourceImageSequence() . . . 566
12.129.3.5 Derive() . . . 566
12.129.3.6 GetFile() [1/2] . . . 567
12.129.3.7 GetFile() [2/2] . . . 567
12.129.3.8 SetAppendDerivationHistory() . . . 567
12.129.3.9 SetDerivationCodeSequenceCodeValue() . . . 567
12.129.3.10 SetDerivationDescription() . . . 567
12.129.3.11 SetFile() . . . 568
12.129.3.12 SetPurposeOfReferenceCodeSequenceCodeValue() . . . 568
12.130 gdcm::FileExplicitFilter Class Reference . . . 568
12.130.1 Detailed Description . . . 569
12.130.2 Constructor & Destructor Documentation . . . 569
12.130.2.1 FileExplicitFilter() . . . 569
12.130.2.2 ∼FileExplicitFilter() . . . 569
12.130.3 Member Function Documentation . . . 569

12.130.3.1 Change() . . . 569
12.130.3.2 ChangeFMI() . . . 570
12.130.3.3 GetFile() . . . 570
12.130.3.4 ProcessDataSet() . . . 570
12.130.3.5 SetChangePrivateTags() . . . 570
12.130.3.6 SetFile() . . . 570
12.130.3.7 SetRecomputeItemLength() . . . 570
12.130.3.8 SetRecomputeSequenceLength() . . . 570
12.130.3.9 SetUseVRUN() . . . 571
12.131 gdcm::FileMetaInformation Class Reference . . . 571
12.131.1 Detailed Description . . . 575
12.131.2 Constructor & Destructor Documentation . . . 575
12.131.2.1 FileMetaInformation() [1/2] . . . 575
12.131.2.2 ∼FileMetaInformation() . . . 575
12.131.2.3 FileMetaInformation() [2/2] . . . 575
12.131.3 Member Function Documentation . . . 575
12.131.3.1 AppendImplementationClassUID() . . . 575
12.131.3.2 ComputeDataSetMediaStorageSOPClass() . . . 576
12.131.3.3 ComputeDataSetTransferSyntax() . . . 576
12.131.3.4 Default() . . . 576
12.131.3.5 FillFromDataSet() . . . 576
12.131.3.6 GetDataSetTransferSyntax() . . . 576
12.131.3.7 GetFileMetaInformationVersion() . . . 576
12.131.3.8 GetFullLength() . . . 576
12.131.3.9 GetGDCMImplementationClassUID() . . . 577
12.131.3.10 GetGDCMImplementationVersionName() . . . 577
12.131.3.11 GetGDCMSourceApplicationEntityTitle() . . . 577
12.131.3.12 GetImplementationClassUID() . . . 577
12.131.3.13 GetImplementationVersionName() . . . 577
12.131.3.14 GetMediaStorage() . . . 577
12.131.3.15 GetMediaStorageAsString() . . . 577
12.131.3.16 GetMetaInformationTS() . . . 577
12.131.3.17 GetPreamble() [1/2] . . . 577
12.131.3.18 GetPreamble() [2/2] . . . 578
12.131.3.19 GetSourceApplicationEntityTitle() . . . 578
12.131.3.20 Insert() . . . 578
12.131.3.21 IsValid() . . . 578
12.131.3.22 operator=() . . . 578
12.131.3.23 Read() . . . 578

12.131.3.24 ReadCompat() . . . 578
12.131.3.25 ReadCompatInternal() . . . 579
12.131.3.26 Replace() . . . 579
12.131.3.27 SetDataSetTransferSyntax() . . . 579
12.131.3.28 SetImplementationClassUID() . . . 579
12.131.3.29 SetImplementationVersionName() . . . 579
12.131.3.30 SetPreamble() . . . 579
12.131.3.31 SetSourceApplicationEntityTitle() . . . 580
12.131.3.32 Write() . . . 580
12.131.4 Friends And Related Symbol Documentation . . . 580
12.131.4.1 operator<< . . . 580
12.131.5 Member Data Documentation . . . 580
12.131.5.1 DataSetMS . . . 580
12.131.5.2 DataSetTS . . . 580
12.131.5.3 MetaInformationTS . . . 581
12.132 gdcm::Filename Class Reference . . . 581
12.132.1 Detailed Description . . . 582
12.132.2 Constructor & Destructor Documentation . . . 582
12.132.2.1 Filename() . . . 582
12.132.3 Member Function Documentation . . . 582
12.132.3.1 EndWith() . . . 582
12.132.3.2 GetExtension() . . . 582
12.132.3.3 GetFileName() . . . 582
12.132.3.4 GetName() . . . 582
12.132.3.5 GetPath() . . . 583
12.132.3.6 IsEmpty() . . . 583
12.132.3.7 IsIdentical() . . . 583
12.132.3.8 Join() . . . 583
12.132.3.9 operator const char *() . . . 583
12.132.3.10 ToUnixSlashes() . . . 583
12.132.3.11 ToWindowsSlashes() . . . 584
12.133 gdcm::FileNameEvent Class Reference . . . 584
12.133.1 Detailed Description . . . 586
12.133.2 Member Typedef Documentation . . . 586
12.133.2.1 Self . . . 586
12.133.2.2 Superclass . . . 586
12.133.3 Constructor & Destructor Documentation . . . 586
12.133.3.1 FileNameEvent() [1/2] . . . 586
12.133.3.2 ~FileNameEvent() . . . 586

12.133.3.3 FileNameEvent() [2/2] . . . 586
12.133.4 Member Function Documentation . . . 587
12.133.4.1 CheckEvent() . . . 587
12.133.4.2 GetEventName() . . . 587
12.133.4.3 GetFileName() . . . 587
12.133.4.4 MakeObject() . . . 587
12.133.4.5 operator=() . . . 587
12.133.4.6 SetFileName() . . . 587
12.134 gdcm::FilenameGenerator Class Reference . . . 588
12.134.1 Detailed Description . . . 588
12.134.2 Member Typedef Documentation . . . 589
12.134.2.1 FilenamesType . . . 589
12.134.2.2 FilenameType . . . 589
12.134.2.3 SizeType . . . 589
12.134.3 Constructor & Destructor Documentation . . . 589
12.134.3.1 FilenameGenerator() . . . 589
12.134.3.2 ∼FilenameGenerator() . . . 589
12.134.4 Member Function Documentation . . . 589
12.134.4.1 Generate() . . . 589
12.134.4.2 GetFilename() . . . 590
12.134.4.3 GetFilenames() . . . 590
12.134.4.4 GetNumberOfFilenames() . . . 590
12.134.4.5 GetPattern() . . . 590
12.134.4.6 GetPrefix() . . . 590
12.134.4.7 SetNumberOfFilenames() . . . 590
12.134.4.8 SetPattern() . . . 591
12.134.4.9 SetPrefix() . . . 591
12.135 gdcm::FileSet Class Reference . . . 591
12.135.1 Detailed Description . . . 592
12.135.2 Member Typedef Documentation . . . 592
12.135.2.1 FilesType . . . 592
12.135.2.2 FileType . . . 592
12.135.3 Constructor & Destructor Documentation . . . 592
12.135.3.1 FileSet() . . . 592
12.135.4 Member Function Documentation . . . 592
12.135.4.1 AddFile() [1/2] . . . 592
12.135.4.2 AddFile() [2/2] . . . 592
12.135.4.3 GetFiles() . . . 593
12.135.4.4 SetFiles() . . . 593

12.135.5 Friends And Related Symbol Documentation . . . 593
12.135.5.1 operator<< . . . 593
12.136 gdcm::FileStreamer Class Reference . . . 593
12.136.1 Detailed Description . . . 595
12.136.2 Constructor & Destructor Documentation . . . 596
12.136.2.1 FileStreamer() . . . 596
12.136.2.2 ~FileStreamer() . . . 596
12.136.3 Member Function Documentation . . . 596
12.136.3.1 AppendToDataElement() . . . 596
12.136.3.2 AppendToGroupDataElement() . . . 596
12.136.3.3 CheckDataElement() . . . 596
12.136.3.4 CheckTemplateFileName() . . . 597
12.136.3.5 New() . . . 597
12.136.3.6 ReserveDataElement() . . . 597
12.136.3.7 ReserveGroupDataElement() . . . 597
12.136.3.8 SetOutputFileName() . . . 597
12.136.3.9 SetTemplateFileName() . . . 598
12.136.3.10 StartDataElement() . . . 598
12.136.3.11 StartGroupDataElement() . . . 598
12.136.3.12 StopDataElement() . . . 598
12.136.3.13 StopGroupDataElement() . . . 599
12.137 gdcm::FileWithName Class Reference . . . 599
12.137.1 Detailed Description . . . 601
12.137.2 Constructor & Destructor Documentation . . . 601
12.137.2.1 FileWithName() . . . 601
12.137.3 Member Data Documentation . . . 601
12.137.3.1 filename . . . 601
12.138 gdcm::FindPatientRootQuery Class Reference . . . 602
12.138.1 Detailed Description . . . 604
12.138.2 Constructor & Destructor Documentation . . . 604
12.138.2.1 FindPatientRootQuery() . . . 604
12.138.3 Member Function Documentation . . . 604
12.138.3.1 GetAbstractSyntaxUID() . . . 604
12.138.3.2 GetTagListByLevel() . . . 605
12.138.3.3 InitializeDataSet() . . . 605
12.138.3.4 ValidateQuery() . . . 605
12.138.4 Friends And Related Symbol Documentation . . . 605
12.138.4.1 QueryFactory . . . 605
12.139 gdcm::FindStudyRootQuery Class Reference . . . 606

12.139.1 Detailed Description . . . 608
12.139.2 Constructor & Destructor Documentation . . . 608
12.139.2.1 FindStudyRootQuery() . . . 608
12.139.3 Member Function Documentation . . . 608
12.139.3.1 GetAbstractSyntaxUID() . . . 608
12.139.3.2 GetTagListByLevel() . . . 609
12.139.3.3 InitializeDataSet() . . . 609
12.139.3.4 ValidateQuery() . . . 609
12.139.4 Friends And Related Symbol Documentation . . . 609
12.139.4.1 QueryFactory . . . 609
12.140 gdcm::Fragment Class Reference . . . 610
12.140.1 Detailed Description . . . 612
12.140.2 Constructor & Destructor Documentation . . . 613
12.140.2.1 Fragment() . . . 613
12.140.3 Member Function Documentation . . . 613
12.140.3.1 ComputeLength() . . . 613
12.140.3.2 GetLength() . . . 613
12.140.3.3 Read() . . . 613
12.140.3.4 ReadBacktrack() . . . 613
12.140.3.5 ReadPreValue() . . . 614
12.140.3.6 ReadValue() . . . 614
12.140.3.7 Write() . . . 614
12.140.4 Friends And Related Symbol Documentation . . . 614
12.140.4.1 operator<< . . . 614
12.141 gdcm::Global Class Reference . . . 615
12.141.1 Detailed Description . . . 615
12.141.2 Constructor & Destructor Documentation . . . 616
12.141.2.1 Global() [1/2] . . . 616
12.141.2.2 ~Global() . . . 616
12.141.2.3 Global() [2/2] . . . 616
12.141.3 Member Function Documentation . . . 616
12.141.3.1 Append() . . . 616
12.141.3.2 GetDefs() . . . 616
12.141.3.3 GetDicts() [1/2] . . . 617
12.141.3.4 GetDicts() [2/2] . . . 617
12.141.3.5 GetInstance() . . . 617
12.141.3.6 LoadResourcesFiles() . . . 617
12.141.3.7 Locate() . . . 618
12.141.3.8 operator=() . . . 618

12.141.3.9 Prepend() . . . 618
12.141.4 Friends And Related Symbol Documentation . . . 618
12.141.4.1 operator<< . . . 618
12.142 gdcm::GroupDict Class Reference . . . 618
12.142.1 Detailed Description . . . 619
12.142.2 Member Typedef Documentation . . . 619
12.142.2.1 GroupStringVector . . . 619
12.142.3 Constructor & Destructor Documentation . . . 619
12.142.3.1 GroupDict() . . . 619
12.142.3.2 ~GroupDict() . . . 620
12.142.4 Member Function Documentation . . . 620
12.142.4.1 Add() . . . 620
12.142.4.2 GetAbbreviation() . . . 620
12.142.4.3 GetName() . . . 620
12.142.4.4 Insert() . . . 620
12.142.4.5 Size() . . . 620
12.142.5 Friends And Related Symbol Documentation . . . 621
12.142.5.1 operator<< . . . 621
12.143 gdcm::IconImageFilter Class Reference . . . 621
12.143.1 Detailed Description . . . 622
12.143.2 Constructor & Destructor Documentation . . . 622
12.143.2.1 IconImageFilter() . . . 622
12.143.2.2 ~IconImageFilter() . . . 622
12.143.3 Member Function Documentation . . . 623
12.143.3.1 Extract() . . . 623
12.143.3.2 ExtractIconImages() . . . 623
12.143.3.3 ExtractVeproIconImages() . . . 623
12.143.3.4 GetFile() [1/2] . . . 623
12.143.3.5 GetFile() [2/2] . . . 623
12.143.3.6 GetIconImage() . . . 623
12.143.3.7 GetNumberOfIconImages() . . . 624
12.143.3.8 SetFile() . . . 624
12.144 gdcm::IconImageGenerator Class Reference . . . 624
12.144.1 Detailed Description . . . 625
12.144.2 Constructor & Destructor Documentation . . . 625
12.144.2.1 IconImageGenerator() . . . 625
12.144.2.2 ~IconImageGenerator() . . . 625
12.144.3 Member Function Documentation . . . 625
12.144.3.1 AutoPixelMinMax() . . . 625

12.144.3.2 ConvertRGBToPaletteColor() . . . 626
12.144.3.3 Generate() . . . 626
12.144.3.4 GetIconImage() . . . 626
12.144.3.5 GetPixmap() [1/2] . . . 626
12.144.3.6 GetPixmap() [2/2] . . . 626
12.144.3.7 SetOutputDimensions() . . . 626
12.144.3.8 SetOutsideValuePixel() . . . 627
12.144.3.9 SetPixelMinMax() . . . 627
12.144.3.10 SetPixmap() . . . 627
12.145 gdcm::ignore_char Struct Reference . . . 627
12.145.1 Constructor & Destructor Documentation . . . 628
12.145.1.1 ignore_char() . . . 628
12.145.2 Member Data Documentation . . . 628
12.145.2.1 m_char . . . 628
12.146 gdcm::Image Class Reference . . . 628
12.146.1 Detailed Description . . . 633
12.146.2 Constructor & Destructor Documentation . . . 633
12.146.2.1 Image() . . . 633
12.146.2.2 ∼Image() . . . 633
12.146.3 Member Function Documentation . . . 634
12.146.3.1 GetDirectionCosines() [1/2] . . . 634
12.146.3.2 GetDirectionCosines() [2/2] . . . 634
12.146.3.3 GetIntercept() . . . 634
12.146.3.4 GetOrigin() [1/2] . . . 634
12.146.3.5 GetOrigin() [2/2] . . . 634
12.146.3.6 GetSlope() . . . 634
12.146.3.7 GetSpacing() [1/2] . . . 634
12.146.3.8 GetSpacing() [2/2] . . . 635
12.146.3.9 Print() . . . 635
12.146.3.10 SetDirectionCosines() [1/3] . . . 635
12.146.3.11 SetDirectionCosines() [2/3] . . . 635
12.146.3.12 SetDirectionCosines() [3/3] . . . 635
12.146.3.13 SetIntercept() . . . 635
12.146.3.14 SetOrigin() [1/3] . . . 636
12.146.3.15 SetOrigin() [2/3] . . . 636
12.146.3.16 SetOrigin() [3/3] . . . 636
12.146.3.17 SetSlope() . . . 636
12.146.3.18 SetSpacing() [1/2] . . . 636
12.146.3.19 SetSpacing() [2/2] . . . 636

12.147 gdcm::ImageApplyLookupTable Class Reference . . . 637
12.147.1 Detailed Description . . . 639
12.147.2 Constructor & Destructor Documentation . . . 639
12.147.2.1 ImageApplyLookupTable() . . . 639
12.147.2.2 ∼ImageApplyLookupTable() . . . 640
12.147.3 Member Function Documentation . . . 640
12.147.3.1 Apply() . . . 640
12.147.3.2 SetRGB8() . . . 640
12.148 gdcm::ImageChangePhotometricInterpretation Class Reference . . . 640
12.148.1 Detailed Description . . . 643
12.148.2 Constructor & Destructor Documentation . . . 643
12.148.2.1 ImageChangePhotometricInterpretation() . . . 643
12.148.2.2 ∼ImageChangePhotometricInterpretation() . . . 643
12.148.3 Member Function Documentation . . . 643
12.148.3.1 Change() . . . 643
12.148.3.2 ChangeMonochrome() . . . 643
12.148.3.3 ChangeRGB2YBR() . . . 643
12.148.3.4 ChangeYBR2RGB() . . . 643
12.148.3.5 GetPhotometricInterpretation() . . . 644
12.148.3.6 RGB2YBR() . . . 644
12.148.3.7 SetPhotometricInterpretation() . . . 644
12.148.3.8 YBR2RGB() . . . 644
12.149 gdcm::ImageChangePlanarConfiguration Class Reference . . . 645
12.149.1 Detailed Description . . . 647
12.149.2 Constructor & Destructor Documentation . . . 648
12.149.2.1 ImageChangePlanarConfiguration() . . . 648
12.149.2.2 ∼ImageChangePlanarConfiguration() . . . 648
12.149.3 Member Function Documentation . . . 648
12.149.3.1 Change() . . . 648
12.149.3.2 GetPlanarConfiguration() . . . 648
12.149.3.3 RGBPixelsToRGBPlanes() . . . 648
12.149.3.4 RGBPlanesToRGBPixels() . . . 649
12.149.3.5 SetPlanarConfiguration() . . . 649
12.150 gdcm::ImageChangeTransferSyntax Class Reference . . . 649
12.150.1 Detailed Description . . . 652
12.150.2 Constructor & Destructor Documentation . . . 652
12.150.2.1 ImageChangeTransferSyntax() . . . 652
12.150.2.2 ∼ImageChangeTransferSyntax() . . . 652
12.150.3 Member Function Documentation . . . 652

12.150.3.1 Change() . . . 652
12.150.3.2 GetTransferSyntax() . . . 653
12.150.3.3 SetCompressIconImage() . . . 653
12.150.3.4 SetForce() . . . 653
12.150.3.5 SetTransferSyntax() . . . 653
12.150.3.6 SetUserCodec() . . . 654
12.150.3.7 TryJPEG2000Codec() . . . 654
12.150.3.8 TryJPEGCodec() . . . 654
12.150.3.9 TryJPEGLSCodec() . . . 654
12.150.3.10 TryRAWCodec() . . . 654
12.150.3.11 TryRLECodec() . . . 655
12.151 gdcm::ImageCodec Class Reference . . . 655
12.151.1 Detailed Description . . . 658
12.151.2 Member Typedef Documentation . . . 658
12.151.2.1 LUTPtr . . . 658
12.151.3 Constructor & Destructor Documentation . . . 658
12.151.3.1 ImageCodec() . . . 658
12.151.3.2 ∼ImageCodec() . . . 658
12.151.4 Member Function Documentation . . . 658
12.151.4.1 AppendFrameEncode() . . . 658
12.151.4.2 AppendRowEncode() . . . 659
12.151.4.3 CanCode() . . . 659
12.151.4.4 CanDecode() . . . 659
12.151.4.5 CleanupUnusedBits() . . . 659
12.151.4.6 Clone() . . . 659
12.151.4.7 Decode() . . . 660
12.151.4.8 DecodeByStreams() . . . 660
12.151.4.9 DoByteSwap() . . . 660
12.151.4.10 DoInvertMonochrome() . . . 660
12.151.4.11 DoOverlayCleanup() . . . 660
12.151.4.12 DoPaddedCompositePixelCode() . . . 660
12.151.4.13 DoPlanarConfiguration() . . . 661
12.151.4.14 DoSimpleCopy() . . . 661
12.151.4.15 DoYBR() . . . 661
12.151.4.16 DoYBRFull422() . . . 661
12.151.4.17 GetDimensions() . . . 661
12.151.4.18 GetHeaderInfo() . . . 661
12.151.4.19 GetLossyFlag() . . . 661
12.151.4.20 GetLUT() . . . 662

12.151.4.21 GetNeedByteSwap() . . . 662
12.151.4.22 GetNumberOfDimensions() . . . 662
12.151.4.23 GetPhotometricInterpretation() . . . 662
12.151.4.24 GetPixelFormat() [1/2] . . . 662
12.151.4.25 GetPixelFormat() [2/2] . . . 662
12.151.4.26 GetPlanarConfiguration() . . . 662
12.151.4.27 IsFrameEncoder() . . . 663
12.151.4.28 IsLossy() . . . 663
12.151.4.29 IsRowEncoder() . . . 663
12.151.4.30 IsValid() . . . 663
12.151.4.31 SetDimensions() [1/2] . . . 663
12.151.4.32 SetDimensions() [2/2] . . . 663
12.151.4.33 SetLossyFlag() . . . 663
12.151.4.34 SetLUT() . . . 664
12.151.4.35 SetNeedByteSwap() . . . 664
12.151.4.36 SetNeedOverlayCleanup() . . . 664
12.151.4.37 SetNumberOfDimensions() . . . 664
12.151.4.38 SetPhotometricInterpretation() . . . 664
12.151.4.39 SetPixelFormat() . . . 665
12.151.4.40 SetPlanarConfiguration() . . . 665
12.151.4.41 StartEncode() . . . 665
12.151.4.42 StopEncode() . . . 665
12.151.5 Friends And Related Symbol Documentation . . . 665
12.151.5.1 FileChangeTransferSyntax . . . 665
12.151.5.2 ImageChangePhotometricInterpretation . . . 666
12.151.6 Member Data Documentation . . . 666
12.151.6.1 Dimensions . . . 666
12.151.6.2 LossyFlag . . . 666
12.151.6.3 LUT . . . 666
12.151.6.4 NeedByteSwap . . . 666
12.151.6.5 NeedOverlayCleanup . . . 666
12.151.6.6 NumberOfDimensions . . . 666
12.151.6.7 PF . . . 667
12.151.6.8 PI . . . 667
12.151.6.9 PlanarConfiguration . . . 667
12.151.6.10 RequestPaddedCompositePixelCode . . . 667
12.151.6.11 RequestPlanarConfiguration . . . 667
12.152 gdcm::ImageConverter Class Reference . . . 667
12.152.1 Detailed Description . . . 668

12.152.2 Constructor & Destructor Documentation . . . 668
12.152.2.1 ImageConverter() . . . 668
12.152.2.2 ∼ImageConverter() . . . 668
12.152.3 Member Function Documentation . . . 668
12.152.3.1 Convert() . . . 668
12.152.3.2 GetOuput() . . . 668
12.152.3.3 SetInput() . . . 668
12.153 gdcm::ImageFragmentSplitter Class Reference . . . 669
12.153.1 Detailed Description . . . 671
12.153.2 Constructor & Destructor Documentation . . . 671
12.153.2.1 ImageFragmentSplitter() . . . 671
12.153.2.2 ∼ImageFragmentSplitter() . . . 672
12.153.3 Member Function Documentation . . . 672
12.153.3.1 GetFragmentSizeMax() . . . 672
12.153.3.2 SetForce() . . . 672
12.153.3.3 SetFragmentSizeMax() . . . 672
12.153.3.4 Split() . . . 672
12.154 gdcm::ImageHelper Class Reference . . . 672
12.154.1 Detailed Description . . . 674
12.154.2 Member Function Documentation . . . 674
12.154.2.1 ComputeMediaStorageFromModality() . . . 674
12.154.2.2 ComputeSpacingFromImagePositionPatient() . . . 674
12.154.2.3 GetDimensionsValue() . . . 674
12.154.2.4 GetDirectionCosinesFromDataSet() . . . 675
12.154.2.5 GetDirectionCosinesValue() . . . 675
12.154.2.6 GetForcePixelSpacing() . . . 675
12.154.2.7 GetForceRescaleInterceptSlope() . . . 675
12.154.2.8 GetLUT() . . . 675
12.154.2.9 GetOriginValue() . . . 675
12.154.2.10 GetPhotometricInterpretationValue() . . . 675
12.154.2.11 GetPixelFormatValue() . . . 676
12.154.2.12 GetPlanarConfigurationValue() . . . 676
12.154.2.13 GetPMSRescaleInterceptSlope() . . . 676
12.154.2.14 GetPointerFromElement() . . . 676
12.154.2.15 GetRealWorldValueMappingContent() . . . 676
12.154.2.16 GetRescaleInterceptSlopeValue() . . . 676
12.154.2.17 GetSecondaryCaptureImagePlaneModule() . . . 677
12.154.2.18 GetSpacingTagFromMediaStorage() . . . 677
12.154.2.19 GetSpacingValue() . . . 677

12.154.2.20 GetZSpacingTagFromMediaStorage() . . . 677
12.154.2.21 SetDimensionsValue() . . . 677
12.154.2.22 SetDirectionCosinesValue() . . . 677
12.154.2.23 SetForcePixelSpacing() . . . 677
12.154.2.24 SetForceRescaleInterceptSlope() . . . 678
12.154.2.25 SetOriginValue() . . . 678
12.154.2.26 SetPMSRescaleInterceptSlope() . . . 678
12.154.2.27 SetRescaleInterceptSlopeValue() . . . 678
12.154.2.28 SetSecondaryCaptureImagePlaneModule() . . . 678
12.154.2.29 SetSpacingValue() . . . 679
12.155 gdcm::ImageReader Class Reference . . . 679
12.155.1 Detailed Description . . . 682
12.155.2 Constructor & Destructor Documentation . . . 682
12.155.2.1 ImageReader() . . . 682
12.155.2.2 ∼ImageReader() . . . 682
12.155.3 Member Function Documentation . . . 682
12.155.3.1 GetImage() [1/2] . . . 682
12.155.3.2 GetImage() [2/2] . . . 683
12.155.3.3 Read() . . . 683
12.155.3.4 ReadACRNEMAImage() . . . 683
12.155.3.5 ReadImage() . . . 683
12.156 gdcm::ImageRegionReader Class Reference . . . 684
12.156.1 Detailed Description . . . 687
12.156.2 Constructor & Destructor Documentation . . . 687
12.156.2.1 ImageRegionReader() . . . 687
12.156.2.2 ∼ImageRegionReader() . . . 688
12.156.3 Member Function Documentation . . . 688
12.156.3.1 ComputeBufferLength() . . . 688
12.156.3.2 GetRegion() . . . 688
12.156.3.3 Read() . . . 688
12.156.3.4 ReadInformation() . . . 688
12.156.3.5 ReadIntoBuffer() . . . 689
12.156.3.6 SetRegion() . . . 689
12.157 gdcm::ImageToImageFilter Class Reference . . . 689
12.157.1 Detailed Description . . . 691
12.157.2 Constructor & Destructor Documentation . . . 691
12.157.2.1 ImageToImageFilter() . . . 691
12.157.2.2 ∼ImageToImageFilter() . . . 691
12.157.3 Member Function Documentation . . . 691

12.157.3.1 GetInput() . . . 691
12.157.3.2 GetOutput() . . . 692
12.158 gdcm::ImageWriter Class Reference . . . 692
12.158.1 Detailed Description . . . 695
12.158.2 Constructor & Destructor Documentation . . . 695
12.158.2.1 ImageWriter() . . . 695
12.158.2.2 ~ImageWriter() . . . 695
12.158.3 Member Function Documentation . . . 695
12.158.3.1 ComputeTargetMediaStorage() . . . 695
12.158.3.2 GetImage() [1/2] . . . 696
12.158.3.3 GetImage() [2/2] . . . 696
12.158.3.4 Write() . . . 696
12.159 gdcm::network::ImplementationClassUIDSub Class Reference . . . 696
12.159.1 Detailed Description . . . 697
12.159.2 Constructor & Destructor Documentation . . . 697
12.159.2.1 ImplementationClassUIDSub() . . . 697
12.159.3 Member Function Documentation . . . 697
12.159.3.1 Print() . . . 697
12.159.3.2 Read() . . . 697
12.159.3.3 Size() . . . 697
12.159.3.4 Write() . . . 697
12.160 gdcm::network::ImplementationUIDSub Class Reference . . . 698
12.160.1 Detailed Description . . . 698
12.160.2 Constructor & Destructor Documentation . . . 698
12.160.2.1 ImplementationUIDSub() . . . 698
12.160.3 Member Function Documentation . . . 698
12.160.3.1 Write() . . . 698
12.161 gdcm::network::ImplementationVersionNameSub Class Reference . . . 698
12.161.1 Detailed Description . . . 699
12.161.2 Constructor & Destructor Documentation . . . 699
12.161.2.1 ImplementationVersionNameSub() . . . 699
12.161.3 Member Function Documentation . . . 699
12.161.3.1 Print() . . . 699
12.161.3.2 Read() . . . 699
12.161.3.3 Size() . . . 699
12.161.3.4 Write() . . . 699
12.162 gdcm::ImplicitDataElement Class Reference . . . 700
12.162.1 Detailed Description . . . 702
12.162.2 Member Function Documentation . . . 703

12.162.2.1 GetLength() . . . 703
12.162.2.2 Read() . . . 703
12.162.2.3 ReadPreValue() . . . 703
12.162.2.4 ReadValue() . . . 703
12.162.2.5 ReadValueWithLength() . . . 703
12.162.2.6 ReadWithLength() . . . 703
12.162.2.7 Write() . . . 704
12.163 gdcm::InitializeEvent Class Reference . . . 704
12.164 gdcm::IOD Class Reference . . . 705
12.164.1 Detailed Description . . . 706
12.164.2 Member Typedef Documentation . . . 706
12.164.2.1 MapIODEntry . . . 706
12.164.2.2 SizeType . . . 706
12.164.3 Constructor & Destructor Documentation . . . 707
12.164.3.1 IOD() . . . 707
12.164.4 Member Function Documentation . . . 707
12.164.4.1 AddIODEntry() . . . 707
12.164.4.2 Clear() . . . 707
12.164.4.3 GetIODEntry() . . . 707
12.164.4.4 GetNumberOfIODs() . . . 707
12.164.4.5 GetTypeFromTag() . . . 707
12.164.5 Friends And Related Symbol Documentation . . . 708
12.164.5.1 operator<< . . . 708
12.165 gdcm::IODEntry Class Reference . . . 708
12.165.1 Detailed Description . . . 709
12.165.2 Constructor & Destructor Documentation . . . 709
12.165.2.1 IODEntry() . . . 709
12.165.3 Member Function Documentation . . . 709
12.165.3.1 GetIE() . . . 709
12.165.3.2 GetName() . . . 710
12.165.3.3 GetRef() . . . 710
12.165.3.4 GetUsage() . . . 710
12.165.3.5 GetUsageType() . . . 710
12.165.3.6 SetIE() . . . 710
12.165.3.7 SetName() . . . 710
12.165.3.8 SetRef() . . . 710
12.165.3.9 SetUsage() . . . 710
12.165.4 Friends And Related Symbol Documentation . . . 711
12.165.4.1 operator<< . . . 711

12.166 gdcm::IODs Class Reference . . . 711
12.166.1 Detailed Description . . . 712
12.166.2 Member Typedef Documentation . . . 712
12.166.2.1 IODMapType . . . 712
12.166.2.2 IODMapTypeConstIterator . . . 712
12.166.2.3 IODName . . . 712
12.166.3 Constructor & Destructor Documentation . . . 712
12.166.3.1 IODs() . . . 712
12.166.4 Member Function Documentation . . . 713
12.166.4.1 AddIOD() . . . 713
12.166.4.2 Begin() . . . 713
12.166.4.3 Clear() . . . 713
12.166.4.4 End() . . . 713
12.166.4.5 GetIOD() . . . 713
12.166.5 Friends And Related Symbol Documentation . . . 713
12.166.5.1 operator<< . . . 713
12.167 gdcm::IPPSorter Class Reference . . . 714
12.167.1 Detailed Description . . . 716
12.167.2 Constructor & Destructor Documentation . . . 716
12.167.2.1 IPPSorter() . . . 716
12.167.3 Member Function Documentation . . . 716
12.167.3.1 GetDirectionCosinesTolerance() . . . 716
12.167.3.2 GetZSpacing() . . . 717
12.167.3.3 GetZSpacingTolerance() . . . 717
12.167.3.4 SetComputeZSpacing() . . . 717
12.167.3.5 SetDirectionCosinesTolerance() . . . 718
12.167.3.6 SetDropDuplicatePositions() . . . 718
12.167.3.7 SetZSpacingTolerance() . . . 718
12.167.3.8 Sort() . . . 718
12.167.4 Member Data Documentation . . . 719
12.167.4.1 ComputeZSpacing . . . 719
12.167.4.2 DirCosTolerance . . . 719
12.167.4.3 DropDuplicatePositions . . . 719
12.167.4.4 ZSpacing . . . 719
12.167.4.5 ZTolerance . . . 719
12.168 gdcm::Item Class Reference . . . 720
12.168.1 Detailed Description . . . 723
12.168.2 Constructor & Destructor Documentation . . . 723
12.168.2.1 Item() [1/2] . . . 723

12.168.2.2 Item() [2/2] . . . 723
12.168.3 Member Function Documentation . . . 723
12.168.3.1 Clear() . . . 723
12.168.3.2 FindDataElement() . . . 724
12.168.3.3 GetDataElement() . . . 724
12.168.3.4 GetLength() . . . 724
12.168.3.5 GetNestedDataSet() [1/2] . . . 724
12.168.3.6 GetNestedDataSet() [2/2] . . . 724
12.168.3.7 InsertDataElement() . . . 724
12.168.3.8 Read() . . . 725
12.168.3.9 SetNestedDataSet() . . . 725
12.168.3.10 Write() . . . 725
12.168.4 Friends And Related Symbol Documentation . . . 725
12.168.4.1 operator<< . . . 725
12.169 gdcm::IterationEvent Class Reference . . . 726
12.170 gdcm::JPEG12Codec Class Reference . . . 727
12.170.1 Detailed Description . . . 730
12.170.2 Constructor & Destructor Documentation . . . 731
12.170.2.1 JPEG12Codec() . . . 731
12.170.2.2 ~JPEG12Codec() . . . 731
12.170.3 Member Function Documentation . . . 731
12.170.3.1 DecodeByStreams() . . . 731
12.170.3.2 EncodeBuffer() . . . 731
12.170.3.3 GetHeaderInfo() . . . 731
12.170.3.4 InternalCode() . . . 731
12.170.3.5 IsStateSuspension() . . . 732
12.171 gdcm::JPEG16Codec Class Reference . . . 732
12.171.1 Detailed Description . . . 735
12.171.2 Constructor & Destructor Documentation . . . 736
12.171.2.1 JPEG16Codec() . . . 736
12.171.2.2 ~JPEG16Codec() . . . 736
12.171.3 Member Function Documentation . . . 736
12.171.3.1 DecodeByStreams() . . . 736
12.171.3.2 EncodeBuffer() . . . 736
12.171.3.3 GetHeaderInfo() . . . 736
12.171.3.4 InternalCode() . . . 736
12.171.3.5 IsStateSuspension() . . . 737
12.172 gdcm::JPEG2000Codec Class Reference . . . 737
12.172.1 Detailed Description . . . 740

12.172.2 Constructor & Destructor Documentation . . . 740
12.172.2.1 JPEG2000Codec() . . . 740
12.172.2.2 ∼JPEG2000Codec() . . . 740
12.172.3 Member Function Documentation . . . 740
12.172.3.1 AppendFrameEncode() . . . 740
12.172.3.2 AppendRowEncode() . . . 741
12.172.3.3 CanCode() . . . 741
12.172.3.4 CanDecode() . . . 741
12.172.3.5 Clone() . . . 741
12.172.3.6 Code() . . . 741
12.172.3.7 Decode() . . . 742
12.172.3.8 DecodeByStreams() . . . 742
12.172.3.9 DecodeExtent() . . . 742
12.172.3.10 GetHeaderInfo() . . . 742
12.172.3.11 GetQuality() . . . 742
12.172.3.12 GetRate() . . . 743
12.172.3.13 IsFrameEncoder() . . . 743
12.172.3.14 IsRowEncoder() . . . 743
12.172.3.15 SetMCT() . . . 743
12.172.3.16 SetNumberOfResolutions() . . . 743
12.172.3.17 SetNumberOfThreadsForDecompression() . . . 743
12.172.3.18 SetQuality() . . . 744
12.172.3.19 SetRate() . . . 744
12.172.3.20 SetReversible() . . . 744
12.172.3.21 SetTileSize() . . . 744
12.172.3.22 StartEncode() . . . 744
12.172.3.23 StopEncode() . . . 744
12.172.4 Friends And Related Symbol Documentation . . . 745
12.172.4.1 Bitmap . . . 745
12.172.4.2 ImageRegionReader . . . 745
12.173 gdcm::JPEG8Codec Class Reference . . . 745
12.173.1 Detailed Description . . . 748
12.173.2 Constructor & Destructor Documentation . . . 749
12.173.2.1 JPEG8Codec() . . . 749
12.173.2.2 ∼JPEG8Codec() . . . 749
12.173.3 Member Function Documentation . . . 749
12.173.3.1 DecodeByStreams() . . . 749
12.173.3.2 EncodeBuffer() . . . 749
12.173.3.3 GetHeaderInfo() . . . 749

12.173.3.4 InternalCode() . . . 749
12.173.3.5 IsStateSuspension() . . . 750
12.174 gdcm::JPEGCodec Class Reference . . . 750
12.174.1 Detailed Description . . . 753
12.174.2 Constructor & Destructor Documentation . . . 754
12.174.2.1 JPEGCodec() . . . 754
12.174.2.2 ∼JPEGCodec() . . . 754
12.174.3 Member Function Documentation . . . 754
12.174.3.1 AppendFrameEncode() . . . 754
12.174.3.2 AppendRowEncode() . . . 754
12.174.3.3 CanCode() . . . 754
12.174.3.4 CanDecode() . . . 755
12.174.3.5 Clone() . . . 755
12.174.3.6 Code() . . . 755
12.174.3.7 ComputeOffsetTable() . . . 755
12.174.3.8 Decode() . . . 755
12.174.3.9 DecodeByStreams() . . . 756
12.174.3.10 DecodeExtent() . . . 756
12.174.3.11 EncodeBuffer() . . . 756
12.174.3.12 GetHeaderInfo() . . . 756
12.174.3.13 GetLossless() . . . 756
12.174.3.14 GetQuality() . . . 757
12.174.3.15 IsFrameEncoder() . . . 757
12.174.3.16 IsRowEncoder() . . . 757
12.174.3.17 IsStateSuspension() . . . 757
12.174.3.18 IsValid() . . . 757
12.174.3.19 SetBitSample() . . . 757
12.174.3.20 SetLossless() . . . 757
12.174.3.21 SetPixelFormat() . . . 758
12.174.3.22 SetQuality() . . . 758
12.174.3.23 StartEncode() . . . 758
12.174.3.24 StopEncode() . . . 758
12.174.4 Friends And Related Symbol Documentation . . . 758
12.174.4.1 ImageRegionReader . . . 758
12.174.5 Member Data Documentation . . . 759
12.174.5.1 BitSample . . . 759
12.174.5.2 Quality . . . 759
12.175 gdcm::JPEGLSCodec Class Reference . . . 759
12.175.1 Detailed Description . . . 762

12.175.2 Constructor & Destructor Documentation . . . 762
12.175.2.1 JPEGLSCodec() . . . 762
12.175.2.2 ∼JPEGLSCodec() . . . 763
12.175.3 Member Function Documentation . . . 763
12.175.3.1 AppendFrameEncode() . . . 763
12.175.3.2 AppendRowEncode() . . . 763
12.175.3.3 CanCode() . . . 763
12.175.3.4 CanDecode() . . . 763
12.175.3.5 Clone() . . . 764
12.175.3.6 Code() . . . 764
12.175.3.7 Decode() [1/2] . . . 764
12.175.3.8 Decode() [2/2] . . . 764
12.175.3.9 DecodeExtent() . . . 764
12.175.3.10 GetBufferLength() . . . 765
12.175.3.11 GetHeaderInfo() . . . 765
12.175.3.12 GetLossless() . . . 765
12.175.3.13 IsFrameEncoder() . . . 765
12.175.3.14 IsRowEncoder() . . . 765
12.175.3.15 SetBufferLength() . . . 765
12.175.3.16 SetLossless() . . . 765
12.175.3.17 SetLossyError() . . . 765
12.175.3.18 StartEncode() . . . 766
12.175.3.19 StopEncode() . . . 766
12.175.4 Friends And Related Symbol Documentation . . . 766
12.175.4.1 ImageRegionReader . . . 766
12.176 gdcm::JSON Class Reference . . . 766
12.176.1 Detailed Description . . . 767
12.176.2 Constructor & Destructor Documentation . . . 767
12.176.2.1 JSON() . . . 767
12.176.2.2 ∼JSON() . . . 767
12.176.3 Member Function Documentation . . . 767
12.176.3.1 Code() . . . 767
12.176.3.2 Decode() . . . 767
12.176.3.3 GetPrettyPrint() . . . 767
12.176.3.4 PrettyPrintOff() . . . 768
12.176.3.5 PrettyPrintOn() . . . 768
12.176.3.6 SetPrettyPrint() . . . 768
12.177 gdcm::KAKADUCodec Class Reference . . . 768
12.177.1 Detailed Description . . . 771

12.177.2 Constructor & Destructor Documentation . . . 771
12.177.2.1 KAKADUCodec() . . . 771
12.177.2.2 ~KAKADUCodec() . . . 771
12.177.3 Member Function Documentation . . . 771
12.177.3.1 CanCode() . . . 771
12.177.3.2 CanDecode() . . . 771
12.177.3.3 Clone() . . . 771
12.177.3.4 Code() . . . 772
12.177.3.5 Decode() . . . 772
12.178 gdcm::LO Class Reference . . . 772
12.178.1 Detailed Description . . . 773
12.178.2 Member Typedef Documentation . . . 774
12.178.2.1 const_iterator . . . 774
12.178.2.2 const_reference . . . 774
12.178.2.3 const_reverse_iterator . . . 774
12.178.2.4 difference_type . . . 774
12.178.2.5 iterator . . . 774
12.178.2.6 pointer . . . 774
12.178.2.7 reference . . . 774
12.178.2.8 reverse_iterator . . . 774
12.178.2.9 size_type . . . 774
12.178.2.10 Superclass . . . 775
12.178.2.11 value_type . . . 775
12.178.3 Constructor & Destructor Documentation . . . 775
12.178.3.1 LO() [1/4] . . . 775
12.178.3.2 LO() [2/4] . . . 775
12.178.3.3 LO() [3/4] . . . 775
12.178.3.4 LO() [4/4] . . . 775
12.178.4 Member Function Documentation . . . 775
12.178.4.1 IsValid() . . . 775
12.179 gdcm::LookupTable Class Reference . . . 776
12.179.1 Detailed Description . . . 778
12.179.2 Member Enumeration Documentation . . . 778
12.179.2.1 LookupTableType . . . 778
12.179.3 Constructor & Destructor Documentation . . . 778
12.179.3.1 LookupTable() [1/2] . . . 778
12.179.3.2 ~LookupTable() . . . 779
12.179.3.3 LookupTable() [2/2] . . . 779
12.179.4 Member Function Documentation . . . 779

12.179.4.1 Allocate() . . . 779
12.179.4.2 Clear() . . . 779
12.179.4.3 Decode() [1/2] . . . 779
12.179.4.4 Decode() [2/2] . . . 779
12.179.4.5 Decode8() . . . 780
12.179.4.6 GetBitSample() . . . 780
12.179.4.7 GetBufferAsRGBA() . . . 780
12.179.4.8 GetLUT() . . . 780
12.179.4.9 GetLUTDescriptor() . . . 780
12.179.4.10 GetLUTLength() . . . 780
12.179.4.11 GetPointer() . . . 781
12.179.4.12 InitializeBlueLUT() . . . 781
12.179.4.13 Initialized() . . . 781
12.179.4.14 InitializeGreenLUT() . . . 781
12.179.4.15 InitializeLUT() . . . 781
12.179.4.16 InitializeRedLUT() . . . 781
12.179.4.17 IsRGB8() . . . 782
12.179.4.18 Print() . . . 782
12.179.4.19 SetBlueLUT() . . . 782
12.179.4.20 SetGreenLUT() . . . 782
12.179.4.21 SetLUT() . . . 782
12.179.4.22 SetRedLUT() . . . 782
12.179.4.23 WriteBufferAsRGBA() . . . 783
12.179.5 Member Data Documentation . . . 783
12.179.5.1 BitSample . . . 783
12.179.5.2 IncompleteLUT . . . 783
12.179.5.3 Internal . . . 783
12.180 gdcm::Scanner2::ltstr Struct Reference . . . 783
12.180.1 Member Function Documentation . . . 784
12.180.1.1 operator()() . . . 784
12.181 gdcm::Scanner::ltstr Struct Reference . . . 784
12.181.1 Member Function Documentation . . . 784
12.181.1.1 operator()() . . . 784
12.182 gdcm::StrictScanner2::ltstr Struct Reference . . . 784
12.182.1 Member Function Documentation . . . 785
12.182.1.1 operator()() . . . 785
12.183 gdcm::StrictScanner::ltstr Struct Reference . . . 785
12.183.1 Member Function Documentation . . . 785
12.183.1.1 operator()() . . . 785

12.184 gdcm::Macro Class Reference . . . 785
12.184.1 Detailed Description . . . 786
12.184.2 Member Typedef Documentation . . . 786
12.184.2.1 ArrayIncludeMacrosType . . . 786
12.184.2.2 MapModuleEntry . . . 786
12.184.3 Constructor & Destructor Documentation . . . 787
12.184.3.1 Macro() . . . 787
12.184.4 Member Function Documentation . . . 787
12.184.4.1 AddMacroEntry() . . . 787
12.184.4.2 Clear() . . . 787
12.184.4.3 FindMacroEntry() . . . 787
12.184.4.4 GetMacroEntry() . . . 787
12.184.4.5 GetName() . . . 787
12.184.4.6 SetName() . . . 788
12.184.4.7 Verify() . . . 788
12.184.5 Friends And Related Symbol Documentation . . . 788
12.184.5.1 operator<< . . . 788
12.185 gdcm::Macros Class Reference . . . 788
12.185.1 Detailed Description . . . 789
12.185.2 Member Typedef Documentation . . . 789
12.185.2.1 ModuleMapType . . . 789
12.185.3 Constructor & Destructor Documentation . . . 789
12.185.3.1 Macros() . . . 789
12.185.4 Member Function Documentation . . . 789
12.185.4.1 AddMacro() . . . 789
12.185.4.2 Clear() . . . 790
12.185.4.3 GetMacro() . . . 790
12.185.4.4 IsEmpty() . . . 790
12.185.5 Friends And Related Symbol Documentation . . . 790
12.185.5.1 operator<< . . . 790
12.186 gdcm::network::MaximumLengthSub Class Reference . . . 790
12.186.1 Detailed Description . . . 791
12.186.2 Constructor & Destructor Documentation . . . 791
12.186.2.1 MaximumLengthSub() . . . 791
12.186.3 Member Function Documentation . . . 791
12.186.3.1 GetMaximumLength() . . . 791
12.186.3.2 Print() . . . 791
12.186.3.3 Read() . . . 791
12.186.3.4 SetMaximumLength() . . . 791

12.186.3.5 Size() . . . 791
12.186.3.6 Write() . . . 792
12.187 gdcm::MD5 Class Reference . . . 792
12.187.1 Detailed Description . . . 792
12.187.2 Member Function Documentation . . . 792
12.187.2.1 Compute() . . . 792
12.187.2.2 ComputeFile() . . . 793
12.188 gdcm::MEC_MR3 Class Reference . . . 793
12.188.1 Detailed Description . . . 793
12.188.2 Member Function Documentation . . . 793
12.188.2.1 GetCanonMECMR3Tag() . . . 793
12.188.2.2 GetPMTFInformationDataTag() . . . 793
12.188.2.3 GetToshibaMECMR3Tag() . . . 794
12.188.2.4 Print() . . . 794
12.189 gdcm::MediaStorage Class Reference . . . 794
12.189.1 Detailed Description . . . 797
12.189.2 Member Enumeration Documentation . . . 798
12.189.2.1 MSType . . . 798
12.189.2.2 ObjectType . . . 800
12.189.3 Constructor & Destructor Documentation . . . 801
12.189.3.1 MediaStorage() . . . 801
12.189.4 Member Function Documentation . . . 801
12.189.4.1 GetModality() . . . 801
12.189.4.2 GetModalityDimension() . . . 801
12.189.4.3 GetMSString() . . . 801
12.189.4.4 GetMSType() . . . 802
12.189.4.5 GetNumberOfModality() . . . 802
12.189.4.6 GetNumberOfMSString() . . . 802
12.189.4.7 GetNumberOfMSType() . . . 802
12.189.4.8 GetString() . . . 802
12.189.4.9 GuessFromModality() . . . 802
12.189.4.10 IsImage() . . . 803
12.189.4.11 IsUndefined() . . . 803
12.189.4.12 operator MSType() . . . 803
12.189.4.13 SetFromDataSet() . . . 803
12.189.4.14 SetFromFile() . . . 803
12.189.4.15 SetFromHeader() . . . 804
12.189.4.16 SetFromModality() . . . 804
12.189.4.17 SetFromSourceImageSequence() . . . 804

12.189.5 Friends And Related Symbol Documentation . . . 804
12.189.5.1 operator<< . . . 804
12.190 gdcm::MemberCommand< T > Class Template Reference . . . 804
12.190.1 Detailed Description . . . 808
12.190.2 Member Typedef Documentation . . . 808
12.190.2.1 Self . . . 808
12.190.2.2 TConstMemberFunctionPointer . . . 808
12.190.2.3 TMemberFunctionPointer . . . 808
12.190.3 Constructor & Destructor Documentation . . . 808
12.190.3.1 MemberCommand() [1/2] . . . 808
12.190.3.2 MemberCommand() [2/2] . . . 808
12.190.3.3 ~MemberCommand() . . . 809
12.190.4 Member Function Documentation . . . 809
12.190.4.1 Execute() [1/2] . . . 809
12.190.4.2 Execute() [2/2] . . . 809
12.190.4.3 New() . . . 809
12.190.4.4 operator=() . . . 809
12.190.4.5 SetCallbackFunction() [1/2] . . . 810
12.190.4.6 SetCallbackFunction() [2/2] . . . 810
12.190.5 Member Data Documentation . . . 810
12.190.5.1 m_ConstMemberFunction . . . 810
12.190.5.2 m_MemberFunction . . . 810
12.190.5.3 m_This . . . 810
12.191 gdcm::MeshPrimitive Class Reference . . . 811
12.191.1 Detailed Description . . . 813
12.191.2 Member Typedef Documentation . . . 813
12.191.2.1 PrimitivesData . . . 813
12.191.3 Member Enumeration Documentation . . . 813
12.191.3.1 MPType . . . 813
12.191.4 Constructor & Destructor Documentation . . . 814
12.191.4.1 MeshPrimitive() . . . 814
12.191.4.2 ~MeshPrimitive() . . . 814
12.191.5 Member Function Documentation . . . 814
12.191.5.1 AddPrimitiveData() . . . 814
12.191.5.2 GetMPType() . . . 814
12.191.5.3 GetMPTypeString() . . . 814
12.191.5.4 GetNumberOfPrimitivesData() . . . 815
12.191.5.5 GetPrimitiveData() [1/4] . . . 815
12.191.5.6 GetPrimitiveData() [2/4] . . . 815

12.191.5.7 GetPrimitiveData() [3/4] . . . 815
12.191.5.8 GetPrimitiveData() [4/4] . . . 815
12.191.5.9 GetPrimitivesData() [1/2] . . . 815
12.191.5.10 GetPrimitivesData() [2/2] . . . 815
12.191.5.11 GetPrimitiveType() . . . 815
12.191.5.12 SetPrimitiveData() [1/2] . . . 815
12.191.5.13 SetPrimitiveData() [2/2] . . . 816
12.191.5.14 SetPrimitivesData() . . . 816
12.191.5.15 SetPrimitiveType() . . . 816
12.191.6 Member Data Documentation . . . 816
12.191.6.1 PrimitiveData . . . 816
12.191.6.2 PrimitiveType . . . 816
12.192 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference . . . 816
12.192.1 Detailed Description . . . 819
12.192.2 Constructor & Destructor Documentation . . . 819
12.192.2.1 ModalityPerformedProcedureStepCreateQuery() . . . 819
12.192.3 Member Function Documentation . . . 819
12.192.3.1 GetAbstractSyntaxUID() . . . 819
12.192.3.2 GetRequiredDataSet() . . . 819
12.192.3.3 ValidateQuery() . . . 819
12.192.4 Friends And Related Symbol Documentation . . . 820
12.192.4.1 QueryFactory . . . 820
12.193 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference . . . 820
12.193.1 Detailed Description . . . 822
12.193.2 Constructor & Destructor Documentation . . . 822
12.193.2.1 ModalityPerformedProcedureStepSetQuery() . . . 822
12.193.3 Member Function Documentation . . . 823
12.193.3.1 GetAbstractSyntaxUID() . . . 823
12.193.3.2 GetRequiredDataSet() . . . 823
12.193.3.3 ValidateQuery() . . . 823
12.193.4 Friends And Related Symbol Documentation . . . 823
12.193.4.1 QueryFactory . . . 823
12.194 gdcm::ModifiedEvent Class Reference . . . 824
12.195 gdcm::Module Class Reference . . . 825
12.195.1 Detailed Description . . . 826
12.195.2 Member Typedef Documentation . . . 826
12.195.2.1 ArrayIncludeMacrosType . . . 826
12.195.2.2 MapModuleEntry . . . 826
12.195.3 Constructor & Destructor Documentation . . . 826

12.195.3.1 Module() . . . 826
12.195.4 Member Function Documentation . . . 826
12.195.4.1 AddMacro() . . . 826
12.195.4.2 AddModuleEntry() . . . 827
12.195.4.3 Clear() . . . 827
12.195.4.4 FindModuleEntryInMacros() . . . 827
12.195.4.5 GetModuleEntryInMacros() . . . 827
12.195.4.6 GetName() . . . 827
12.195.4.7 SetName() . . . 827
12.195.4.8 Verify() . . . 827
12.195.5 Friends And Related Symbol Documentation . . . 828
12.195.5.1 operator<< . . . 828
12.196 gdcm::ModuleEntry Class Reference . . . 828
12.196.1 Detailed Description . . . 830
12.196.2 Member Typedef Documentation . . . 830
12.196.2.1 Description . . . 830
12.196.3 Constructor & Destructor Documentation . . . 830
12.196.3.1 ModuleEntry() . . . 830
12.196.3.2 ~ModuleEntry() . . . 830
12.196.4 Member Function Documentation . . . 831
12.196.4.1 GetDescription() . . . 831
12.196.4.2 GetName() . . . 831
12.196.4.3 GetType() . . . 831
12.196.4.4 SetDescription() . . . 831
12.196.4.5 SetName() . . . 831
12.196.4.6 SetType() . . . 831
12.196.5 Friends And Related Symbol Documentation . . . 832
12.196.5.1 operator<< . . . 832
12.196.6 Member Data Documentation . . . 832
12.196.6.1 DataElementType . . . 832
12.196.6.2 DescriptionField . . . 832
12.196.6.3 Name . . . 832
12.197 gdcm::Modules Class Reference . . . 832
12.197.1 Detailed Description . . . 833
12.197.2 Member Typedef Documentation . . . 833
12.197.2.1 ModuleMapType . . . 833
12.197.3 Constructor & Destructor Documentation . . . 833
12.197.3.1 Modules() . . . 833
12.197.4 Member Function Documentation . . . 834

12.197.4.1 AddModule() . . . 834
12.197.4.2 Clear() . . . 834
12.197.4.3 GetModule() . . . 834
12.197.4.4 IsEmpty() . . . 834
12.197.5 Friends And Related Symbol Documentation . . . 834
12.197.5.1 operator<< . . . 834
12.198 gdcm::MovePatientRootQuery Class Reference . . . 835
12.198.1 Detailed Description . . . 837
12.198.2 Constructor & Destructor Documentation . . . 837
12.198.2.1 MovePatientRootQuery() . . . 837
12.198.3 Member Function Documentation . . . 837
12.198.3.1 GetAbstractSyntaxUID() . . . 837
12.198.3.2 GetTagListByLevel() . . . 838
12.198.3.3 InitializeDataSet() . . . 838
12.198.3.4 ValidateQuery() . . . 838
12.198.4 Friends And Related Symbol Documentation . . . 838
12.198.4.1 QueryFactory . . . 838
12.199 gdcm::MoveStudyRootQuery Class Reference . . . 839
12.199.1 Detailed Description . . . 841
12.199.2 Constructor & Destructor Documentation . . . 841
12.199.2.1 MoveStudyRootQuery() . . . 841
12.199.3 Member Function Documentation . . . 841
12.199.3.1 GetAbstractSyntaxUID() . . . 841
12.199.3.2 GetTagListByLevel() . . . 842
12.199.3.3 InitializeDataSet() . . . 842
12.199.3.4 ValidateQuery() . . . 842
12.199.4 Friends And Related Symbol Documentation . . . 842
12.199.4.1 QueryFactory . . . 842
12.200 gdcm::MrProtocol Class Reference . . . 843
12.200.1 Detailed Description . . . 843
12.200.2 Constructor & Destructor Documentation . . . 843
12.200.2.1 MrProtocol() . . . 843
12.200.2.2 ~MrProtocol() . . . 844
12.200.3 Member Function Documentation . . . 844
12.200.3.1 FindMrProtocolByName() . . . 844
12.200.3.2 GetMrProtocolByName() . . . 844
12.200.3.3 GetSliceArray() . . . 844
12.200.3.4 GetVersion() . . . 844
12.200.3.5 Load() . . . 844

12.200.3.6 Print() . . . 844
12.200.4 Friends And Related Symbol Documentation . . . 845
12.200.4.1 operator<< . . . 845
12.201 gdcm::network::NActionRQ Class Reference . . . 845
12.201.1 Detailed Description . . . 846
12.201.2 Member Function Documentation . . . 846
12.201.2.1 ConstructPDV() . . . 846
12.202 gdcm::network::NActionRSP Class Reference . . . 846
12.202.1 Detailed Description . . . 847
12.202.2 Member Function Documentation . . . 847
12.202.2.1 ConstructPDVByDataSet() . . . 847
12.203 gdcm::network::NCreateRQ Class Reference . . . 848
12.203.1 Detailed Description . . . 849
12.203.2 Member Function Documentation . . . 849
12.203.2.1 ConstructPDV() . . . 849
12.204 gdcm::network::NCreateRSP Class Reference . . . 849
12.204.1 Detailed Description . . . 850
12.204.2 Member Function Documentation . . . 850
12.204.2.1 ConstructPDVByDataSet() . . . 850
12.205 gdcm::network::NDeleteRQ Class Reference . . . 851
12.205.1 Detailed Description . . . 852
12.205.2 Member Function Documentation . . . 852
12.205.2.1 ConstructPDV() . . . 852
12.206 gdcm::network::NDeleteRSP Class Reference . . . 852
12.206.1 Detailed Description . . . 853
12.206.2 Member Function Documentation . . . 853
12.206.2.1 ConstructPDVByDataSet() . . . 853
12.207 gdcm::NestedModuleEntries Class Reference . . . 854
12.207.1 Detailed Description . . . 856
12.207.2 Member Typedef Documentation . . . 856
12.207.2.1 SizeType . . . 856
12.207.3 Constructor & Destructor Documentation . . . 856
12.207.3.1 NestedModuleEntries() . . . 856
12.207.4 Member Function Documentation . . . 856
12.207.4.1 AddModuleEntry() . . . 856
12.207.4.2 GetModuleEntry() [1/2] . . . 856
12.207.4.3 GetModuleEntry() [2/2] . . . 857
12.207.4.4 GetNumberOfModuleEntries() . . . 857
12.207.5 Friends And Related Symbol Documentation . . . 857

12.207.5.1 operator<< . . . 857
12.208 gdcm::network::NEventReportRQ Class Reference . . . 857
12.208.1 Detailed Description . . . 858
12.208.2 Member Function Documentation . . . 858
12.208.2.1 ConstructPDV() . . . 858
12.209 gdcm::network::NEventReportRSP Class Reference . . . 859
12.209.1 Detailed Description . . . 860
12.209.2 Member Function Documentation . . . 860
12.209.2.1 ConstructPDVByDataSet() . . . 860
12.210 gdcm::network::NGetRQ Class Reference . . . 860
12.210.1 Detailed Description . . . 861
12.210.2 Member Function Documentation . . . 861
12.210.2.1 ConstructPDV() . . . 861
12.211 gdcm::network::NGetRSP Class Reference . . . 862
12.211.1 Detailed Description . . . 863
12.211.2 Member Function Documentation . . . 863
12.211.2.1 ConstructPDVByDataSet() . . . 863
12.212 gdcm::NoEvent Class Reference . . . 863
12.212.1 Detailed Description . . . 864
12.213 gdcm::network::NormalizedMessageFactory Class Reference . . . 864
12.213.1 Member Function Documentation . . . 864
12.213.1.1 ConstructNAction() . . . 864
12.213.1.2 ConstructNCreate() . . . 865
12.213.1.3 ConstructNDelete() . . . 865
12.213.1.4 ConstructNEventReport() . . . 865
12.213.1.5 ConstructNGet() . . . 865
12.213.1.6 ConstructNSet() . . . 865
12.214 gdcm::NormalizedNetworkFunctions Class Reference . . . 865
12.214.1 Detailed Description . . . 866
12.214.2 Member Function Documentation . . . 866
12.214.2.1 ConstructQuery() . . . 866
12.214.2.2 NAction() . . . 867
12.214.2.3 NCreate() . . . 867
12.214.2.4 NDelete() . . . 867
12.214.2.5 NEventReport() . . . 867
12.214.2.6 NGet() . . . 867
12.214.2.7 NSet() . . . 868
12.215 gdcm::network::NSetRQ Class Reference . . . 868
12.215.1 Detailed Description . . . 869

12.215.2 Member Function Documentation . . . 869
12.215.2.1 ConstructPDV() . . . 869
12.216 gdcm::network::NSetRSP Class Reference . . . 869
12.216.1 Detailed Description . . . 870
12.216.2 Member Function Documentation . . . 870
12.216.2.1 ConstructPDVByDataSet() . . . 870
12.217 gdcm::Object Class Reference . . . 871
12.217.1 Detailed Description . . . 872
12.217.2 Constructor & Destructor Documentation . . . 872
12.217.2.1 Object() [1/2] . . . 872
12.217.2.2 ∼Object() . . . 872
12.217.2.3 Object() [2/2] . . . 872
12.217.3 Member Function Documentation . . . 873
12.217.3.1 operator=() . . . 873
12.217.3.2 Print() . . . 873
12.217.3.3 Register() . . . 873
12.217.3.4 UnRegister() . . . 873
12.217.4 Friends And Related Symbol Documentation . . . 873
12.217.4.1 operator<< . . . 873
12.217.4.2 SmartPointer . . . 874
12.218 gdcm::OpenSSLCryptoFactory Class Reference . . . 874
12.218.1 Constructor & Destructor Documentation . . . 875
12.218.1.1 OpenSSLCryptoFactory() . . . 875
12.218.2 Member Function Documentation . . . 876
12.218.2.1 CreateCMSProvider() . . . 876
12.218.2.2 InitOpenSSL() . . . 876
12.219 gdcm::OpenSSLCryptographicMessageSyntax Class Reference . . . 876
12.219.1 Constructor & Destructor Documentation . . . 878
12.219.1.1 OpenSSLCryptographicMessageSyntax() . . . 878
12.219.1.2 ∼OpenSSLCryptographicMessageSyntax() . . . 878
12.219.2 Member Function Documentation . . . 878
12.219.2.1 Decrypt() . . . 878
12.219.2.2 Encrypt() . . . 878
12.219.2.3 GetCipherType() . . . 878
12.219.2.4 ParseCertificateFile() . . . 879
12.219.2.5 ParseKeyFile() . . . 879
12.219.2.6 SetCipherType() . . . 879
12.219.2.7 SetPassword() . . . 879
12.220 gdcm::OpenSSLP7CryptoFactory Class Reference . . . 880

12.220.1 Constructor & Destructor Documentation . . . 881
12.220.1.1 OpenSSLP7CryptoFactory() . . . 881
12.220.2 Member Function Documentation . . . 881
12.220.2.1 CreateCMSProvider() . . . 881
12.221 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference . . . 882
12.221.1 Detailed Description . . . 883
12.221.2 Constructor & Destructor Documentation . . . 883
12.221.2.1 OpenSSLP7CryptographicMessageSyntax() . . . 883
12.221.2.2 ∼OpenSSLP7CryptographicMessageSyntax() . . . 883
12.221.3 Member Function Documentation . . . 883
12.221.3.1 Decrypt() . . . 883
12.221.3.2 Encrypt() . . . 884
12.221.3.3 GetCipherType() . . . 884
12.221.3.4 ParseCertificateFile() . . . 884
12.221.3.5 ParseKeyFile() . . . 884
12.221.3.6 SetCipherType() . . . 884
12.221.3.7 SetPassword() . . . 885
12.222 gdcm::Orientation Class Reference . . . 885
12.222.1 Detailed Description . . . 886
12.222.2 Member Enumeration Documentation . . . 886
12.222.2.1 OrientationType . . . 886
12.222.3 Constructor & Destructor Documentation . . . 886
12.222.3.1 Orientation() . . . 886
12.222.3.2 ∼Orientation() . . . 886
12.222.4 Member Function Documentation . . . 887
12.222.4.1 GetLabel() . . . 887
12.222.4.2 GetMajorAxisFromPatientRelativeDirectionCosine() . . . 887
12.222.4.3 GetObliquityThresholdCosineValue() . . . 887
12.222.4.4 GetType() . . . 887
12.222.4.5 Print() . . . 887
12.222.4.6 SetObliquityThresholdCosineValue() . . . 888
12.222.5 Friends And Related Symbol Documentation . . . 888
12.222.5.1 operator<< . . . 888
12.223 gdcm::Overlay Class Reference . . . 888
12.223.1 Detailed Description . . . 891
12.223.2 Member Enumeration Documentation . . . 891
12.223.2.1 OverlayType . . . 891
12.223.3 Constructor & Destructor Documentation . . . 891
12.223.3.1 Overlay() [1/2] . . . 891

12.223.3.2 ~Overlay() . . . 892
12.223.3.3 Overlay() [2/2] . . . 892
12.223.4 Member Function Documentation . . . 892
12.223.4.1 Decompress() . . . 892
12.223.4.2 GetBitPosition() . . . 892
12.223.4.3 GetBitsAllocated() . . . 892
12.223.4.4 GetColumns() . . . 892
12.223.4.5 GetDescription() . . . 892
12.223.4.6 GetGroup() . . . 893
12.223.4.7 GetOrigin() . . . 893
12.223.4.8 GetOverlayData() . . . 893
12.223.4.9 GetOverlayTypeAsString() . . . 893
12.223.4.10 GetOverlayTypeFromString() . . . 893
12.223.4.11 GetRows() . . . 893
12.223.4.12 GetType() . . . 893
12.223.4.13 GetTypeAsEnum() . . . 894
12.223.4.14 GetUnpackBuffer() . . . 894
12.223.4.15 GetUnpackBufferLength() . . . 894
12.223.4.16 GrabOverlayFromPixelData() . . . 894
12.223.4.17 IsEmpty() . . . 894
12.223.4.18 IsInPixelData() [1/2] . . . 894
12.223.4.19 IsInPixelData() [2/2] . . . 895
12.223.4.20 IsZero() . . . 895
12.223.4.21 operator=() . . . 895
12.223.4.22 Print() . . . 895
12.223.4.23 SetBitPosition() . . . 895
12.223.4.24 SetBitsAllocated() . . . 895
12.223.4.25 SetColumns() . . . 896
12.223.4.26 SetDescription() . . . 896
12.223.4.27 SetFrameOrigin() . . . 896
12.223.4.28 SetGroup() . . . 896
12.223.4.29 SetNumberOfFrames() . . . 896
12.223.4.30 SetOrigin() . . . 896
12.223.4.31 SetOverlay() . . . 897
12.223.4.32 SetRows() . . . 897
12.223.4.33 SetType() . . . 897
12.223.4.34 Update() . . . 897
12.224 gdcm::ParseException Class Reference . . . 898
12.224.1 Detailed Description . . . 899

12.224.2 Constructor & Destructor Documentation . . . 899
12.224.2.1 ParseException() [1/2] . . . 899
12.224.2.2 ~ParseException() . . . 899
12.224.2.3 ParseException() [2/2] . . . 899
12.224.3 Member Function Documentation . . . 900
12.224.3.1 GetLastElement() . . . 900
12.224.3.2 operator=() . . . 900
12.224.3.3 SetLastElement() . . . 900
12.225 gdcm::Parser Class Reference . . . 900
12.225.1 Detailed Description . . . 901
12.225.2 Member Typedef Documentation . . . 901
12.225.2.1 EndElementHandler . . . 901
12.225.2.2 StartElementHandler . . . 901
12.225.3 Member Enumeration Documentation . . . 902
12.225.3.1 ErrorType . . . 902
12.225.4 Constructor & Destructor Documentation . . . 902
12.225.4.1 Parser() . . . 902
12.225.4.2 ~Parser() . . . 902
12.225.5 Member Function Documentation . . . 902
12.225.5.1 GetBuffer() . . . 902
12.225.5.2 GetCurrentByteIndex() . . . 902
12.225.5.3 GetErrorCode() . . . 903
12.225.5.4 GetErrorString() . . . 903
12.225.5.5 GetUserData() . . . 903
12.225.5.6 Parse() . . . 903
12.225.5.7 ParseBuffer() . . . 903
12.225.5.8 Process() . . . 903
12.225.5.9 SetElementHandler() . . . 903
12.225.5.10 SetUserData() . . . 903
12.226 gdcm::Patient Class Reference . . . 904
12.226.1 Detailed Description . . . 904
12.226.2 Constructor & Destructor Documentation . . . 904
12.226.2.1 Patient() . . . 904
12.227 gdcm::network::PDataTFPDU Class Reference . . . 904
12.227.1 Detailed Description . . . 905
12.227.2 Member Typedef Documentation . . . 906
12.227.2.1 SizeType . . . 906
12.227.3 Constructor & Destructor Documentation . . . 906
12.227.3.1 PDataTFPDU() . . . 906

12.227.4 Member Function Documentation . . . 906
12.227.4.1 AddPresentationDataValue() . . . 906
12.227.4.2 GetNumberOfPresentationDataValues() . . . 906
12.227.4.3 GetPresentationDataValue() . . . 906
12.227.4.4 IsLastFragment() . . . 906
12.227.4.5 Print() . . . 906
12.227.4.6 Read() . . . 907
12.227.4.7 ReadInto() . . . 907
12.227.4.8 Size() . . . 907
12.227.4.9 Write() . . . 907
12.228 gdcm::PDBElement Class Reference . . . 908
12.228.1 Detailed Description . . . 909
12.228.2 Constructor & Destructor Documentation . . . 909
12.228.2.1 PDBElement() . . . 909
12.228.3 Member Function Documentation . . . 909
12.228.3.1 GetName() . . . 909
12.228.3.2 GetValue() . . . 909
12.228.3.3 operator==() . . . 909
12.228.3.4 SetName() . . . 910
12.228.3.5 SetValue() . . . 910
12.228.4 Friends And Related Symbol Documentation . . . 910
12.228.4.1 operator<< . . . 910
12.228.5 Member Data Documentation . . . 910
12.228.5.1 NameField . . . 910
12.228.5.2 ValueField . . . 910
12.229 gdcm::PDBHeader Class Reference . . . 911
12.229.1 Detailed Description . . . 911
12.229.2 Constructor & Destructor Documentation . . . 912
12.229.2.1 PDBHeader() . . . 912
12.229.2.2 ∼PDBHeader() . . . 912
12.229.3 Member Function Documentation . . . 912
12.229.3.1 FindPDBElementByName() . . . 912
12.229.3.2 GetPDBEEnd() . . . 912
12.229.3.3 GetPDBElementByName() . . . 912
12.229.3.4 GetPDBInfoTag() . . . 912
12.229.3.5 LoadFromDataElement() . . . 913
12.229.3.6 Print() . . . 913
12.229.4 Friends And Related Symbol Documentation . . . 913
12.229.4.1 operator<< . . . 913

12.230 gdcm::PDFCodec Class Reference . . . 913
12.230.1 Detailed Description . . . 915
12.230.2 Constructor & Destructor Documentation . . . 915
12.230.2.1 PDFCodec() . . . 915
12.230.2.2 ∼PDFCodec() . . . 915
12.230.3 Member Function Documentation . . . 915
12.230.3.1 CanCode() . . . 915
12.230.3.2 CanDecode() . . . 916
12.230.3.3 Decode() . . . 916
12.231 gdcm::network::PDUFactory Class Reference . . . 916
12.231.1 Detailed Description . . . 917
12.231.2 Member Function Documentation . . . 917
12.231.2.1 ConstructAbortPDU() . . . 917
12.231.2.2 ConstructPDU() . . . 917
12.231.2.3 ConstructReleasePDU() . . . 917
12.231.2.4 CreateCEchoPDU() . . . 917
12.231.2.5 CreateCFindPDU() . . . 917
12.231.2.6 CreateCMovePDU() . . . 917
12.231.2.7 CreateCStoreRQPDU() . . . 918
12.231.2.8 CreateCStoreRSPPDU() . . . 918
12.231.2.9 CreateNActionPDU() . . . 918
12.231.2.10 CreateNCreatePDU() . . . 918
12.231.2.11 CreateNDeletePDU() . . . 918
12.231.2.12 CreateNEventReportPDU() . . . 918
12.231.2.13 CreateNGetPDU() . . . 918
12.231.2.14 CreateNSetPDU() . . . 919
12.231.2.15 DetermineEventByPDU() . . . 919
12.231.2.16 GetPDVs() . . . 919
12.232 gdcm::PersonName Class Reference . . . 919
12.232.1 Detailed Description . . . 920
12.232.2 Member Function Documentation . . . 920
12.232.2.1 GetMaxLength() . . . 920
12.232.2.2 GetNumberOfComponents() . . . 920
12.232.2.3 Print() . . . 920
12.232.2.4 SetBlob() . . . 920
12.232.2.5 SetComponents() [1/2] . . . 921
12.232.2.6 SetComponents() [2/2] . . . 921
12.232.3 Member Data Documentation . . . 921
12.232.3.1 Component . . . 921

12.232.3.2 MaxLength . . . 921
12.232.3.3 MaxNumberOfComponents . . . 921
12.232.3.4 Padding . . . 921
12.232.3.5 Separator . . . 922
12.233 gdcm::PGXCodec Class Reference . . . 922
12.233.1 Detailed Description . . . 925
12.233.2 Constructor & Destructor Documentation . . . 925
12.233.2.1 PGXCodec() . . . 925
12.233.2.2 ∼PGXCodec() . . . 925
12.233.3 Member Function Documentation . . . 925
12.233.3.1 CanCode() . . . 925
12.233.3.2 CanDecode() . . . 925
12.233.3.3 Clone() . . . 925
12.233.3.4 GetHeaderInfo() . . . 926
12.233.3.5 Read() . . . 926
12.233.3.6 Write() . . . 926
12.234 gdcm::PhotometricInterpretation Class Reference . . . 926
12.234.1 Detailed Description . . . 927
12.234.2 Member Enumeration Documentation . . . 927
12.234.2.1 PIType . . . 927
12.234.3 Constructor & Destructor Documentation . . . 928
12.234.3.1 PhotometricInterpretation() . . . 928
12.234.4 Member Function Documentation . . . 928
12.234.4.1 GetPIString() . . . 928
12.234.4.2 GetPIType() . . . 928
12.234.4.3 GetSamplesPerPixel() . . . 929
12.234.4.4 GetString() . . . 929
12.234.4.5 GetType() . . . 929
12.234.4.6 IsLossless() . . . 929
12.234.4.7 IsLossy() . . . 929
12.234.4.8 IsRetired() . . . 929
12.234.4.9 IsSameColorSpace() . . . 929
12.234.4.10 operator PIType() . . . 929
12.234.5 Friends And Related Symbol Documentation . . . 930
12.234.5.1 operator<< . . . 930
12.235 gdcm::PixelFormat Class Reference . . . 930
12.235.1 Detailed Description . . . 932
12.235.2 Member Enumeration Documentation . . . 932
12.235.2.1 ScalarType . . . 932

12.235.3 Constructor & Destructor Documentation . . . 933
12.235.3.1 PixelFormat() [1/3] . . . 933
12.235.3.2 PixelFormat() [2/3] . . . 933
12.235.3.3 PixelFormat() [3/3] . . . 933
12.235.4 Member Function Documentation . . . 933
12.235.4.1 GetBitsAllocated() . . . 933
12.235.4.2 GetBitsStored() . . . 934
12.235.4.3 GetHighBit() . . . 934
12.235.4.4 GetMax() . . . 934
12.235.4.5 GetMin() . . . 934
12.235.4.6 GetPixelRepresentation() . . . 934
12.235.4.7 GetPixelSize() . . . 934
12.235.4.8 GetSamplesPerPixel() . . . 935
12.235.4.9 GetScalarType() . . . 935
12.235.4.10 GetScalarTypeAsString() . . . 935
12.235.4.11 IsCompatible() . . . 935
12.235.4.12 IsValid() . . . 935
12.235.4.13 operator ScalarType() . . . 936
12.235.4.14 operator"!=() [1/2] . . . 936
12.235.4.15 operator"!=() [2/2] . . . 936
12.235.4.16 operator==() [1/2] . . . 936
12.235.4.17 operator==() [2/2] . . . 936
12.235.4.18 Print() . . . 936
12.235.4.19 SetBitsAllocated() . . . 937
12.235.4.20 SetBitsStored() . . . 937
12.235.4.21 SetHighBit() . . . 937
12.235.4.22 SetPixelRepresentation() . . . 937
12.235.4.23 SetSamplesPerPixel() . . . 937
12.235.4.24 SetScalarType() . . . 938
12.235.4.25 Validate() . . . 938
12.235.5 Friends And Related Symbol Documentation . . . 938
12.235.5.1 Bitmap . . . 938
12.235.5.2 operator<< . . . 938
12.236 gdcm::Pixmap Class Reference . . . 939
12.236.1 Detailed Description . . . 942
12.236.2 Constructor & Destructor Documentation . . . 943
12.236.2.1 Pixmap() . . . 943
12.236.2.2 ~Pixmap() . . . 943
12.236.3 Member Function Documentation . . . 943

12.236.3.1 AreOverlaysInPixelData() . . . 943
12.236.3.2 GetCurve() [1/2] . . . 943
12.236.3.3 GetCurve() [2/2] . . . 943
12.236.3.4 GetIconImage() [1/2] . . . 943
12.236.3.5 GetIconImage() [2/2] . . . 944
12.236.3.6 GetNumberOfCurves() . . . 944
12.236.3.7 GetNumberOfOverlays() . . . 944
12.236.3.8 GetOverlay() [1/2] . . . 944
12.236.3.9 GetOverlay() [2/2] . . . 944
12.236.3.10 Print() . . . 944
12.236.3.11 RemoveOverlay() . . . 945
12.236.3.12 SetIconImage() . . . 945
12.236.3.13 SetNumberOfCurves() . . . 945
12.236.3.14 SetNumberOfOverlays() . . . 945
12.236.3.15 UnusedBitsPresentInPixelData() . . . 945
12.236.4 Member Data Documentation . . . 945
12.236.4.1 Curves . . . 945
12.236.4.2 Icon . . . 946
12.236.4.3 Overlays . . . 946
12.237 gdcm::PixmapReader Class Reference . . . 946
12.237.1 Detailed Description . . . 948
12.237.2 Constructor & Destructor Documentation . . . 949
12.237.2.1 PixmapReader() . . . 949
12.237.2.2 ~PixmapReader() . . . 949
12.237.3 Member Function Documentation . . . 949
12.237.3.1 GetPixmap() [1/2] . . . 949
12.237.3.2 GetPixmap() [2/2] . . . 949
12.237.3.3 Read() . . . 949
12.237.3.4 ReadACRNEMAImage() . . . 950
12.237.3.5 ReadImage() . . . 950
12.237.3.6 ReadImageInternal() . . . 950
12.237.4 Member Data Documentation . . . 950
12.237.4.1 PixelData . . . 950
12.238 gdcm::PixmapToPixmapFilter Class Reference . . . 950
12.238.1 Detailed Description . . . 952
12.238.2 Constructor & Destructor Documentation . . . 952
12.238.2.1 PixmapToPixmapFilter() . . . 952
12.238.2.2 ~PixmapToPixmapFilter() . . . 952
12.238.3 Member Function Documentation . . . 952

12.238.3.1 GetInput() . . . 952
12.238.3.2 GetOutput() . . . 952
12.238.3.3 GetOutputAsPixmap() . . . 953
12.239 gdcm::PixmapWriter Class Reference . . . 953
12.239.1 Detailed Description . . . 955
12.239.2 Constructor & Destructor Documentation . . . 956
12.239.2.1 PixmapWriter() . . . 956
12.239.2.2 ~PixmapWriter() . . . 956
12.239.3 Member Function Documentation . . . 956
12.239.3.1 DoIconImage() . . . 956
12.239.3.2 GetImage() [1/2] . . . 956
12.239.3.3 GetImage() [2/2] . . . 956
12.239.3.4 GetPixmap() [1/2] . . . 956
12.239.3.5 GetPixmap() [2/2] . . . 957
12.239.3.6 PrepareWrite() . . . 957
12.239.3.7 SetImage() . . . 957
12.239.3.8 SetPixmap() . . . 957
12.239.3.9 Write() . . . 957
12.239.4 Member Data Documentation . . . 958
12.239.4.1 PixelData . . . 958
12.240 gdcm::PNMCodec Class Reference . . . 958
12.240.1 Detailed Description . . . 961
12.240.2 Constructor & Destructor Documentation . . . 961
12.240.2.1 PNMCodec() . . . 961
12.240.2.2 ~PNMCodec() . . . 961
12.240.3 Member Function Documentation . . . 961
12.240.3.1 CanCode() . . . 961
12.240.3.2 CanDecode() . . . 962
12.240.3.3 Clone() . . . 962
12.240.3.4 GetBufferLength() . . . 962
12.240.3.5 GetHeaderInfo() . . . 962
12.240.3.6 Read() . . . 962
12.240.3.7 SetBufferLength() . . . 962
12.240.3.8 Write() . . . 963
12.241 gdcm::Preamble Class Reference . . . 963
12.241.1 Detailed Description . . . 964
12.241.2 Constructor & Destructor Documentation . . . 964
12.241.2.1 Preamble() [1/2] . . . 964
12.241.2.2 ~Preamble() . . . 964

12.241.2.3 Preamble() [2/2] . . . 964
12.241.3 Member Function Documentation . . . 964
12.241.3.1 Clear() . . . 964
12.241.3.2 Create() . . . 965
12.241.3.3 GetInternal() . . . 965
12.241.3.4 GetLength() . . . 965
12.241.3.5 IsEmpty() . . . 965
12.241.3.6 IsValid() . . . 965
12.241.3.7 operator=() . . . 965
12.241.3.8 Print() . . . 965
12.241.3.9 Read() . . . 966
12.241.3.10 Remove() . . . 966
12.241.3.11 Valid() . . . 966
12.241.3.12 Write() . . . 966
12.241.4 Friends And Related Symbol Documentation . . . 966
12.241.4.1 operator<< . . . 966
12.242 gdcm::PresentationContext Class Reference . . . 967
12.242.1 Detailed Description . . . 968
12.242.2 Member Typedef Documentation . . . 968
12.242.2.1 SizeType . . . 968
12.242.2.2 TransferSyntaxArrayType . . . 968
12.242.3 Constructor & Destructor Documentation . . . 968
12.242.3.1 PresentationContext() [1/2] . . . 968
12.242.3.2 PresentationContext() [2/2] . . . 968
12.242.4 Member Function Documentation . . . 969
12.242.4.1 AddTransferSyntax() . . . 969
12.242.4.2 GetAbstractSyntax() . . . 969
12.242.4.3 GetNumberOfTransferSyntaxes() . . . 969
12.242.4.4 GetPresentationContextID() . . . 969
12.242.4.5 GetTransferSyntax() . . . 969
12.242.4.6 operator==() . . . 969
12.242.4.7 Print() . . . 969
12.242.4.8 SetAbstractSyntax() . . . 970
12.242.4.9 SetPresentationContextID() . . . 970
12.242.5 Member Data Documentation . . . 970
12.242.5.1 AbstractSyntax . . . 970
12.242.5.2 ID . . . 970
12.242.5.3 TransferSyntaxes . . . 970
12.243 gdcm::network::PresentationContextAC Class Reference . . . 970

12.243.1 Detailed Description . . . 971
12.243.2 Constructor & Destructor Documentation . . . 971
12.243.2.1 PresentationContextAC() . . . 971
12.243.3 Member Function Documentation . . . 971
12.243.3.1 GetPresentationContextID() . . . 971
12.243.3.2 GetReason() . . . 971
12.243.3.3 GetTransferSyntax() . . . 971
12.243.3.4 Print() . . . 972
12.243.3.5 Read() . . . 972
12.243.3.6 SetPresentationContextID() . . . 972
12.243.3.7 SetReason() . . . 972
12.243.3.8 SetTransferSyntax() . . . 972
12.243.3.9 Size() . . . 972
12.243.3.10 Write() . . . 972
12.244 gdcm::PresentationContextGenerator Class Reference . . . 973
12.244.1 Detailed Description . . . 973
12.244.2 Member Typedef Documentation . . . 974
12.244.2.1 PresentationContextArrayType . . . 974
12.244.2.2 SizeType . . . 974
12.244.3 Constructor & Destructor Documentation . . . 974
12.244.3.1 PresentationContextGenerator() . . . 974
12.244.4 Member Function Documentation . . . 974
12.244.4.1 AddFromFile() . . . 974
12.244.4.2 AddPresentationContext() . . . 974
12.244.4.3 GenerateFromFilenames() . . . 974
12.244.4.4 GenerateFromUID() . . . 975
12.244.4.5 GetDefaultTransferSyntax() . . . 975
12.244.4.6 GetPresentationContexts() . . . 975
12.244.4.7 SetDefaultTransferSyntax() . . . 975
12.244.4.8 SetMergeModeToAbstractSyntax() . . . 975
12.244.4.9 SetMergeModeToTransferSyntax() . . . 975
12.245 gdcm::network::PresentationContextRQ Class Reference . . . 976
12.245.1 Detailed Description . . . 976
12.245.2 Member Typedef Documentation . . . 976
12.245.2.1 SizeType . . . 976
12.245.3 Constructor & Destructor Documentation . . . 977
12.245.3.1 PresentationContextRQ() [1/3] . . . 977
12.245.3.2 PresentationContextRQ() [2/3] . . . 977
12.245.3.3 PresentationContextRQ() [3/3] . . . 977

12.245.4 Member Function Documentation . . . 977
12.245.4.1 AddTransferSyntax() . . . 977
12.245.4.2 GetAbstractSyntax() [1/2] . . . 977
12.245.4.3 GetAbstractSyntax() [2/2] . . . 977
12.245.4.4 GetNumberOfTransferSyntaxes() . . . 977
12.245.4.5 GetPresentationContextID() . . . 978
12.245.4.6 GetTransferSyntax() [1/2] . . . 978
12.245.4.7 GetTransferSyntax() [2/2] . . . 978
12.245.4.8 GetTransferSyntaxes() . . . 978
12.245.4.9 operator==() . . . 978
12.245.4.10 Print() . . . 978
12.245.4.11 Read() . . . 978
12.245.4.12 SetAbstractSyntax() . . . 978
12.245.4.13 SetPresentationContextID() . . . 979
12.245.4.14 Size() . . . 979
12.245.4.15 Write() . . . 979
12.246 gdcm::network::PresentationDataValue Class Reference . . . 979
12.246.1 Detailed Description . . . 980
12.246.2 Constructor & Destructor Documentation . . . 980
12.246.2.1 PresentationDataValue() . . . 980
12.246.3 Member Function Documentation . . . 980
12.246.3.1 ConcatenatePDVBlobs() . . . 980
12.246.3.2 ConcatenatePDVBlobsAsExplicit() . . . 980
12.246.3.3 GetBlob() . . . 980
12.246.3.4 GetIsCommand() . . . 980
12.246.3.5 GetIsLastFragment() . . . 981
12.246.3.6 GetMessageHeader() . . . 981
12.246.3.7 GetPresentationContextID() . . . 981
12.246.3.8 Print() . . . 981
12.246.3.9 Read() . . . 981
12.246.3.10 ReadInto() . . . 981
12.246.3.11 SetBlob() . . . 981
12.246.3.12 SetCommand() . . . 981
12.246.3.13 SetDataSet() . . . 982
12.246.3.14 SetLastFragment() . . . 982
12.246.3.15 SetMessageHeader() . . . 982
12.246.3.16 SetPresentationContextID() . . . 982
12.246.3.17 Size() . . . 982
12.246.3.18 Write() . . . 982

12.247 gdcm::Printer Class Reference . . . 983
12.247.1 Detailed Description . . . 984
12.247.2 Member Enumeration Documentation . . . 985
12.247.2.1 PrintStyles . . . 985
12.247.3 Constructor & Destructor Documentation . . . 985
12.247.3.1 Printer() . . . 985
12.247.3.2 ∼Printer() . . . 985
12.247.4 Member Function Documentation . . . 985
12.247.4.1 GetPrintStyle() . . . 985
12.247.4.2 Print() . . . 985
12.247.4.3 PrintDataElement() . . . 986
12.247.4.4 PrintDataSet() . . . 986
12.247.4.5 PrintSQ() . . . 986
12.247.4.6 SetColor() . . . 986
12.247.4.7 SetFile() . . . 986
12.247.4.8 SetStyle() . . . 987
12.247.5 Member Data Documentation . . . 987
12.247.5.1 F . . . 987
12.247.5.2 MaxPrintLength . . . 987
12.247.5.3 PrintStyle . . . 987
12.248 gdcm::PrivateDict Class Reference . . . 987
12.248.1 Detailed Description . . . 988
12.248.2 Constructor & Destructor Documentation . . . 988
12.248.2.1 PrivateDict() . . . 988
12.248.2.2 ∼PrivateDict() . . . 988
12.248.3 Member Function Documentation . . . 988
12.248.3.1 AddDictEntry() . . . 988
12.248.3.2 FindDictEntry() . . . 988
12.248.3.3 GetDictEntry() . . . 989
12.248.3.4 IsEmpty() . . . 989
12.248.3.5 LoadDefault() . . . 989
12.248.3.6 PrintXML() . . . 989
12.248.3.7 RemoveDictEntry() . . . 989
12.248.4 Friends And Related Symbol Documentation . . . 989
12.248.4.1 Dicts . . . 989
12.248.4.2 operator<< . . . 990
12.249 gdcm::PrivateTag Class Reference . . . 990
12.249.1 Detailed Description . . . 992
12.249.2 Constructor & Destructor Documentation . . . 993

12.249.2.1 PrivateTag() [1/2] . . . 993
12.249.2.2 PrivateTag() [2/2] . . . 993
12.249.3 Member Function Documentation . . . 993
12.249.3.1 GetAsDataElement() . . . 993
12.249.3.2 GetOwner() . . . 993
12.249.3.3 operator"!=() [1/2] . . . 993
12.249.3.4 operator"!=() [2/2] . . . 994
12.249.3.5 operator<() . . . 994
12.249.3.6 operator=() . . . 994
12.249.3.7 operator==() [1/2] . . . 994
12.249.3.8 operator==() [2/2] . . . 994
12.249.3.9 ReadFromCommaSeparatedString() . . . 994
12.249.3.10 SetOwner() . . . 995
12.249.4 Friends And Related Symbol Documentation . . . 995
12.249.4.1 operator<< . . . 995
12.250 gdcm::ProgressEvent Class Reference . . . 995
12.250.1 Detailed Description . . . 997
12.250.2 Member Typedef Documentation . . . 997
12.250.2.1 Self . . . 997
12.250.2.2 Superclass . . . 997
12.250.3 Constructor & Destructor Documentation . . . 997
12.250.3.1 ProgressEvent() [1/2] . . . 997
12.250.3.2 ~ProgressEvent() . . . 997
12.250.3.3 ProgressEvent() [2/2] . . . 997
12.250.4 Member Function Documentation . . . 998
12.250.4.1 CheckEvent() . . . 998
12.250.4.2 GetEventName() . . . 998
12.250.4.3 GetProgress() . . . 998
12.250.4.4 MakeObject() . . . 998
12.250.4.5 operator=() . . . 998
12.250.4.6 SetProgress() . . . 998
12.251 gdcm::PVRGCodec Class Reference . . . 999
12.251.1 Detailed Description . . . 1002
12.251.2 Constructor & Destructor Documentation . . . 1002
12.251.2.1 PVRGCodec() . . . 1002
12.251.2.2 ~PVRGCodec() . . . 1002
12.251.3 Member Function Documentation . . . 1002
12.251.3.1 CanCode() . . . 1002
12.251.3.2 CanDecode() . . . 1002

12.251.3.3 Clone() . . . 1003
12.251.3.4 Code() . . . 1003
12.251.3.5 Decode() . . . 1003
12.251.3.6 SetLossyFlag() . . . 1003
12.252 gdcm::PythonFilter Class Reference . . . 1003
12.252.1 Detailed Description . . . 1004
12.252.2 Constructor & Destructor Documentation . . . 1004
12.252.2.1 PythonFilter() . . . 1004
12.252.2.2 ∼PythonFilter() . . . 1004
12.252.3 Member Function Documentation . . . 1004
12.252.3.1 GetFile() [1/2] . . . 1004
12.252.3.2 GetFile() [2/2] . . . 1004
12.252.3.3 SetDicts() . . . 1004
12.252.3.4 SetFile() . . . 1005
12.252.3.5 ToPyObject() . . . 1005
12.252.3.6 UseDictAlways() . . . 1005
12.253 gdcm::QueryBase Class Reference . . . 1005
12.253.1 Detailed Description . . . 1006
12.253.2 Constructor & Destructor Documentation . . . 1006
12.253.2.1 ∼QueryBase() . . . 1006
12.253.3 Member Function Documentation . . . 1006
12.253.3.1 GetAllRequiredTags() . . . 1006
12.253.3.2 GetAllTags() . . . 1006
12.253.3.3 GetHierachicalSearchTags() . . . 1007
12.253.3.4 GetName() . . . 1007
12.253.3.5 GetOptionalTags() . . . 1007
12.253.3.6 GetQueryLevel() . . . 1007
12.253.3.7 GetRequiredTags() . . . 1007
12.253.3.8 GetUniqueTags() . . . 1007
12.254 gdcm::QueryFactory Class Reference . . . 1008
12.254.1 Detailed Description . . . 1008
12.254.2 Member Function Documentation . . . 1008
12.254.2.1 GetCharacterFromCurrentLocale() . . . 1008
12.254.2.2 ListCharSets() . . . 1008
12.254.2.3 ProduceCharacterSetDataElement() . . . 1009
12.254.2.4 ProduceQuery() [1/2] . . . 1009
12.254.2.5 ProduceQuery() [2/2] . . . 1009
12.255 gdcm::QueryImage Class Reference . . . 1009
12.255.1 Detailed Description . . . 1010

12.255.2 Member Function Documentation . . . 1010
12.255.2.1 GetHierachicalSearchTags() . . . 1010
12.255.2.2 GetName() . . . 1011
12.255.2.3 GetOptionalTags() . . . 1011
12.255.2.4 GetQueryLevel() . . . 1011
12.255.2.5 GetRequiredTags() . . . 1011
12.255.2.6 GetUniqueTags() . . . 1011
12.256 gdcm::QueryPatient Class Reference . . . 1012
12.256.1 Detailed Description . . . 1013
12.256.2 Member Function Documentation . . . 1013
12.256.2.1 GetHierachicalSearchTags() . . . 1013
12.256.2.2 GetName() . . . 1013
12.256.2.3 GetOptionalTags() . . . 1013
12.256.2.4 GetQueryLevel() . . . 1013
12.256.2.5 GetRequiredTags() . . . 1014
12.256.2.6 GetUniqueTags() . . . 1014
12.257 gdcm::QuerySeries Class Reference . . . 1014
12.257.1 Detailed Description . . . 1015
12.257.2 Member Function Documentation . . . 1015
12.257.2.1 GetHierachicalSearchTags() . . . 1015
12.257.2.2 GetName() . . . 1016
12.257.2.3 GetOptionalTags() . . . 1016
12.257.2.4 GetQueryLevel() . . . 1016
12.257.2.5 GetRequiredTags() . . . 1016
12.257.2.6 GetUniqueTags() . . . 1016
12.258 gdcm::QueryStudy Class Reference . . . 1017
12.258.1 Detailed Description . . . 1018
12.258.2 Member Function Documentation . . . 1018
12.258.2.1 GetHierachicalSearchTags() . . . 1018
12.258.2.2 GetName() . . . 1018
12.258.2.3 GetOptionalTags() . . . 1018
12.258.2.4 GetQueryLevel() . . . 1018
12.258.2.5 GetRequiredTags() . . . 1019
12.258.2.6 GetUniqueTags() . . . 1019
12.259 gdcm::RAWCodec Class Reference . . . 1019
12.259.1 Detailed Description . . . 1022
12.259.2 Constructor & Destructor Documentation . . . 1022
12.259.2.1 RAWCodec() . . . 1022
12.259.2.2 ~RAWCodec() . . . 1022

12.259.3 Member Function Documentation . . . 1022
12.259.3.1 CanCode() . . . 1022
12.259.3.2 CanDecode() . . . 1022
12.259.3.3 Clone() . . . 1023
12.259.3.4 Code() . . . 1023
12.259.3.5 Decode() . . . 1023
12.259.3.6 DecodeByStreams() . . . 1023
12.259.3.7 DecodeBytes() . . . 1023
12.259.3.8 GetHeaderInfo() . . . 1024
12.260 gdcm::Reader Class Reference . . . 1024
12.260.1 Detailed Description . . . 1026
12.260.2 Constructor & Destructor Documentation . . . 1027
12.260.2.1 Reader() . . . 1027
12.260.2.2 ∼Reader() . . . 1027
12.260.3 Member Function Documentation . . . 1027
12.260.3.1 CanRead() . . . 1027
12.260.3.2 GetFile() [1/2] . . . 1027
12.260.3.3 GetFile() [2/2] . . . 1027
12.260.3.4 GetStreamCurrentPosition() . . . 1028
12.260.3.5 GetStreamPtr() . . . 1028
12.260.3.6 Read() . . . 1028
12.260.3.7 ReadDataSet() . . . 1028
12.260.3.8 ReadMetaInformation() . . . 1028
12.260.3.9 ReadPreamble() . . . 1029
12.260.3.10 ReadSelectedPrivateTags() . . . 1029
12.260.3.11 ReadSelectedTags() . . . 1029
12.260.3.12 ReadUpToTag() . . . 1029
12.260.3.13 SetFile() . . . 1029
12.260.3.14 SetFileName() . . . 1030
12.260.3.15 SetStream() . . . 1030
12.260.4 Friends And Related Symbol Documentation . . . 1030
12.260.4.1 StreamImageReader . . . 1030
12.260.5 Member Data Documentation . . . 1031
12.260.5.1 F . . . 1031
12.261 gdcm::RealWorldValueMappingContent Struct Reference . . . 1031
12.261.1 Member Data Documentation . . . 1032
12.261.1.1 CodeMeaning . . . 1032
12.261.1.2 CodeValue . . . 1032
12.261.1.3 RealWorldValueIntercept . . . 1032

12.261.1.4 RealWorldValueSlope . . . 1032
12.262 gdcm::Region Class Reference . . . 1032
12.262.1 Detailed Description . . . 1033
12.262.2 Constructor & Destructor Documentation . . . 1033
12.262.2.1 Region() . . . 1033
12.262.2.2 ∼Region() . . . 1033
12.262.3 Member Function Documentation . . . 1033
12.262.3.1 Area() . . . 1033
12.262.3.2 Clone() . . . 1034
12.262.3.3 ComputeBoundingBox() . . . 1034
12.262.3.4 Empty() . . . 1034
12.262.3.5 IsValid() . . . 1034
12.262.3.6 Print() . . . 1034
12.263 gdcm::Rescaler Class Reference . . . 1035
12.263.1 Detailed Description . . . 1035
12.263.2 Constructor & Destructor Documentation . . . 1036
12.263.2.1 Rescaler() . . . 1036
12.263.2.2 ∼Rescaler() . . . 1036
12.263.3 Member Function Documentation . . . 1036
12.263.3.1 ComputeInterceptSlopePixelType() . . . 1036
12.263.3.2 ComputePixelTypeFromMinMax() . . . 1037
12.263.3.3 GetIntercept() . . . 1037
12.263.3.4 GetSlope() . . . 1037
12.263.3.5 InverseRescale() . . . 1037
12.263.3.6 InverseRescaleFunctionIntoBestFit() . . . 1037
12.263.3.7 Rescale() . . . 1037
12.263.3.8 RescaleFunctionIntoBestFit() . . . 1038
12.263.3.9 SetIntercept() . . . 1038
12.263.3.10 SetMinMaxForPixelType() . . . 1038
12.263.3.11 SetPixelFormat() . . . 1038
12.263.3.12 SetSlope() . . . 1038
12.263.3.13 SetTargetPixelType() . . . 1039
12.263.3.14 SetUseTargetPixelType() . . . 1039
12.264 gdcm::RLECodec Class Reference . . . 1039
12.264.1 Detailed Description . . . 1042
12.264.2 Constructor & Destructor Documentation . . . 1042
12.264.2.1 RLECodec() . . . 1042
12.264.2.2 ∼RLECodec() . . . 1042
12.264.3 Member Function Documentation . . . 1043

12.264.3.1 AppendFrameEncode() . . . 1043
12.264.3.2 AppendRowEncode() . . . 1043
12.264.3.3 CanCode() . . . 1043
12.264.3.4 CanDecode() . . . 1043
12.264.3.5 Clone() . . . 1043
12.264.3.6 Code() . . . 1044
12.264.3.7 Decode() . . . 1044
12.264.3.8 DecodeByStreams() . . . 1044
12.264.3.9 DecodeExtent() . . . 1044
12.264.3.10 GetBufferLength() . . . 1044
12.264.3.11 GetHeaderInfo() . . . 1045
12.264.3.12 IsFrameEncoder() . . . 1045
12.264.3.13 IsRowEncoder() . . . 1045
12.264.3.14 SetBufferLength() . . . 1045
12.264.3.15 SetLength() . . . 1045
12.264.3.16 StartEncode() . . . 1045
12.264.3.17 StopEncode() . . . 1045
12.264.4 Friends And Related Symbol Documentation . . . 1046
12.264.4.1 ImageRegionReader . . . 1046
12.265 gdcm::network::RoleSelectionSub Class Reference . . . 1046
12.265.1 Detailed Description . . . 1046
12.265.2 Constructor & Destructor Documentation . . . 1046
12.265.2.1 RoleSelectionSub() . . . 1046
12.265.3 Member Function Documentation . . . 1047
12.265.3.1 Print() . . . 1047
12.265.3.2 Read() . . . 1047
12.265.3.3 SetTuple() . . . 1047
12.265.3.4 Size() . . . 1047
12.265.3.5 Write() . . . 1047
12.266 gdcm::Scanner Class Reference . . . 1048
12.266.1 Detailed Description . . . 1051
12.266.2 Member Typedef Documentation . . . 1051
12.266.2.1 ConstIterator . . . 1051
12.266.2.2 MappingType . . . 1051
12.266.2.3 TagToValue . . . 1051
12.266.2.4 TagToValueValueType . . . 1052
12.266.2.5 ValuesType . . . 1052
12.266.3 Constructor & Destructor Documentation . . . 1052
12.266.3.1 Scanner() . . . 1052

12.266.3.2 ~Scanner() . . . 1052
12.266.4 Member Function Documentation . . . 1052
12.266.4.1 AddPrivateTag() . . . 1052
12.266.4.2 AddSkipTag() . . . 1052
12.266.4.3 AddTag() . . . 1053
12.266.4.4 Begin() . . . 1053
12.266.4.5 ClearSkipTags() . . . 1053
12.266.4.6 ClearTags() . . . 1053
12.266.4.7 End() . . . 1053
12.266.4.8 GetAllFilenamesFromTagToValue() . . . 1053
12.266.4.9 GetFilenameFromTagToValue() . . . 1053
12.266.4.10 GetFilenames() . . . 1054
12.266.4.11 GetKeys() . . . 1054
12.266.4.12 GetMapping() . . . 1054
12.266.4.13 GetMappingFromTagToValue() . . . 1054
12.266.4.14 GetMappings() . . . 1054
12.266.4.15 GetOrderedValues() . . . 1054
12.266.4.16 GetValue() . . . 1055
12.266.4.17 GetValues() [1/2] . . . 1055
12.266.4.18 GetValues() [2/2] . . . 1055
12.266.4.19 IsKey() . . . 1055
12.266.4.20 New() . . . 1056
12.266.4.21 Print() . . . 1056
12.266.4.22 PrintTable() . . . 1056
12.266.4.23 ProcessPublicTag() . . . 1056
12.266.4.24 Scan() . . . 1056
12.266.5 Friends And Related Symbol Documentation . . . 1057
12.266.5.1 operator<< . . . 1057
12.267 gdcm::Scanner2 Class Reference . . . 1057
12.267.1 Detailed Description . . . 1060
12.267.2 Member Typedef Documentation . . . 1061
12.267.2.1 PrivateConstIterator . . . 1061
12.267.2.2 PrivateMappingType . . . 1061
12.267.2.3 PrivateTagToValue . . . 1061
12.267.2.4 PrivateTagToValueValueType . . . 1061
12.267.2.5 PublicConstIterator . . . 1061
12.267.2.6 PublicMappingType . . . 1061
12.267.2.7 PublicTagToValue . . . 1061
12.267.2.8 PublicTagToValueValueType . . . 1061

12.267.2.9 ValuesType . . . 1062
12.267.3 Constructor & Destructor Documentation . . . 1062
12.267.3.1 Scanner2() . . . 1062
12.267.3.2 ∼Scanner2() . . . 1062
12.267.4 Member Function Documentation . . . 1062
12.267.4.1 AddPrivateTag() . . . 1062
12.267.4.2 AddPublicTag() . . . 1062
12.267.4.3 AddSkipTag() . . . 1062
12.267.4.4 Begin() . . . 1062
12.267.4.5 ClearPrivateTags() . . . 1063
12.267.4.6 ClearPublicTags() . . . 1063
12.267.4.7 ClearSkipTags() . . . 1063
12.267.4.8 End() . . . 1063
12.267.4.9 GetAllFilenamesFromPrivateTagToValue() . . . 1063
12.267.4.10 GetAllFilenamesFromPublicTagToValue() . . . 1063
12.267.4.11 GetFilenameFromPrivateTagToValue() . . . 1063
12.267.4.12 GetFilenameFromPublicTagToValue() . . . 1063
12.267.4.13 GetFilenames() . . . 1064
12.267.4.14 GetKeys() . . . 1064
12.267.4.15 GetMappingFromPrivateTagToValue() . . . 1064
12.267.4.16 GetMappingFromPublicTagToValue() . . . 1064
12.267.4.17 GetPrivateMapping() . . . 1064
12.267.4.18 GetPrivateMappings() . . . 1064
12.267.4.19 GetPrivateOrderedValues() . . . 1064
12.267.4.20 GetPrivateValue() . . . 1065
12.267.4.21 GetPrivateValues() . . . 1065
12.267.4.22 GetPublicMapping() . . . 1065
12.267.4.23 GetPublicMappings() . . . 1065
12.267.4.24 GetPublicOrderedValues() . . . 1065
12.267.4.25 GetPublicValue() . . . 1065
12.267.4.26 GetPublicValues() . . . 1066
12.267.4.27 GetValues() . . . 1066
12.267.4.28 IsKey() . . . 1066
12.267.4.29 New() . . . 1066
12.267.4.30 Print() . . . 1066
12.267.4.31 PrintTable() . . . 1066
12.267.4.32 PrivateBegin() . . . 1067
12.267.4.33 PrivateEnd() . . . 1067
12.267.4.34 ProcessPrivateTag() . . . 1067

12.267.4.35 ProcessPublicTag() . . . 1067
12.267.4.36 Scan() . . . 1067
12.267.5 Friends And Related Symbol Documentation . . . 1067
12.267.5.1 operator<< . . . 1067
12.268 gdcm::Segment Class Reference . . . 1068
12.268.1 Detailed Description . . . 1070
12.268.2 Member Typedef Documentation . . . 1070
12.268.2.1 BasicCodedEntryVector . . . 1070
12.268.2.2 SurfaceVector . . . 1070
12.268.3 Member Enumeration Documentation . . . 1071
12.268.3.1 ALGOType . . . 1071
12.268.4 Constructor & Destructor Documentation . . . 1071
12.268.4.1 Segment() . . . 1071
12.268.4.2 ~Segment() . . . 1071
12.268.5 Member Function Documentation . . . 1071
12.268.5.1 AddSurface() . . . 1071
12.268.5.2 GetALGOType() . . . 1071
12.268.5.3 GetALGOTypeString() . . . 1071
12.268.5.4 GetAnatomicRegion() [1/2] . . . 1072
12.268.5.5 GetAnatomicRegion() [2/2] . . . 1072
12.268.5.6 GetAnatomicRegionModifiers() [1/2] . . . 1072
12.268.5.7 GetAnatomicRegionModifiers() [2/2] . . . 1072
12.268.5.8 GetPropertyCategory() [1/2] . . . 1072
12.268.5.9 GetPropertyCategory() [2/2] . . . 1072
12.268.5.10 GetPropertyType() [1/2] . . . 1072
12.268.5.11 GetPropertyType() [2/2] . . . 1072
12.268.5.12 GetPropertyTypeModifiers() [1/2] . . . 1072
12.268.5.13 GetPropertyTypeModifiers() [2/2] . . . 1072
12.268.5.14 GetSegmentAlgorithmName() . . . 1073
12.268.5.15 GetSegmentAlgorithmType() . . . 1073
12.268.5.16 GetSegmentDescription() . . . 1073
12.268.5.17 GetSegmentLabel() . . . 1073
12.268.5.18 GetSegmentNumber() . . . 1073
12.268.5.19 GetSurface() . . . 1073
12.268.5.20 GetSurfaceCount() . . . 1073
12.268.5.21 GetSurfaces() [1/2] . . . 1073
12.268.5.22 GetSurfaces() [2/2] . . . 1073
12.268.5.23 SetAnatomicRegion() . . . 1074
12.268.5.24 SetAnatomicRegionModifiers() . . . 1074

12.268.5.25 SetPropertyCategory() . . . 1074
12.268.5.26 SetPropertyType() . . . 1074
12.268.5.27 SetPropertyTypeModifiers() . . . 1074
12.268.5.28 SetSegmentAlgorithmName() . . . 1074
12.268.5.29 SetSegmentAlgorithmType() [1/2] . . . 1074
12.268.5.30 SetSegmentAlgorithmType() [2/2] . . . 1074
12.268.5.31 SetSegmentDescription() . . . 1075
12.268.5.32 SetSegmentLabel() . . . 1075
12.268.5.33 SetSegmentNumber() . . . 1075
12.268.5.34 SetSurfaceCount() . . . 1075
12.268.6 Member Data Documentation . . . 1075
12.268.6.1 AnatomicRegion . . . 1075
12.268.6.2 AnatomicRegionModifiers . . . 1075
12.268.6.3 PropertyCategory . . . 1075
12.268.6.4 PropertyType . . . 1075
12.268.6.5 PropertyTypeModifiers . . . 1076
12.268.6.6 SegmentAlgorithmName . . . 1076
12.268.6.7 SegmentAlgorithmType . . . 1076
12.268.6.8 SegmentDescription . . . 1076
12.268.6.9 SegmentLabel . . . 1076
12.268.6.10 SegmentNumber . . . 1076
12.268.6.11 SurfaceCount . . . 1076
12.268.6.12 Surfaces . . . 1076
12.269 gdcm::SegmentedPaletteColorLookupTable Class Reference . . . 1077
12.269.1 Detailed Description . . . 1079
12.269.2 Constructor & Destructor Documentation . . . 1079
12.269.2.1 SegmentedPaletteColorLookupTable() . . . 1079
12.269.2.2 ~SegmentedPaletteColorLookupTable() . . . 1079
12.269.3 Member Function Documentation . . . 1080
12.269.3.1 Print() . . . 1080
12.269.3.2 SetLUT() . . . 1080
12.270 gdcm::SegmentReader Class Reference . . . 1080
12.270.1 Detailed Description . . . 1083
12.270.2 Member Typedef Documentation . . . 1083
12.270.2.1 SegmentMap . . . 1083
12.270.2.2 SegmentVector . . . 1083
12.270.3 Constructor & Destructor Documentation . . . 1083
12.270.3.1 SegmentReader() . . . 1083
12.270.3.2 ~SegmentReader() . . . 1083

12.270.4 Member Function Documentation . . . 1083
12.270.4.1 GetSegments() [1/2] . . . 1083
12.270.4.2 GetSegments() [2/2] . . . 1083
12.270.4.3 Read() . . . 1084
12.270.4.4 ReadSegment() . . . 1084
12.270.4.5 ReadSegments() . . . 1084
12.270.5 Member Data Documentation . . . 1084
12.270.5.1 Segments . . . 1084
12.271 gdcm::SegmentWriter Class Reference . . . 1084
12.271.1 Detailed Description . . . 1087
12.271.2 Member Typedef Documentation . . . 1088
12.271.2.1 SegmentVector . . . 1088
12.271.3 Constructor & Destructor Documentation . . . 1088
12.271.3.1 SegmentWriter() . . . 1088
12.271.3.2 ∼SegmentWriter() . . . 1088
12.271.4 Member Function Documentation . . . 1088
12.271.4.1 AddSegment() . . . 1088
12.271.4.2 GetNumberOfSegments() . . . 1088
12.271.4.3 GetSegment() . . . 1088
12.271.4.4 GetSegments() [1/2] . . . 1088
12.271.4.5 GetSegments() [2/2] . . . 1088
12.271.4.6 PrepareWrite() . . . 1089
12.271.4.7 SetNumberOfSegments() . . . 1089
12.271.4.8 SetSegments() . . . 1089
12.271.4.9 Write() . . . 1089
12.271.5 Member Data Documentation . . . 1089
12.271.5.1 Segments . . . 1089
12.272 gdcm::SequenceOfFragments Class Reference . . . 1090
12.272.1 Detailed Description . . . 1092
12.272.2 Member Typedef Documentation . . . 1092
12.272.2.1 ConstIterator . . . 1092
12.272.2.2 FragmentVector . . . 1092
12.272.2.3 Iterator . . . 1092
12.272.2.4 SizeType . . . 1093
12.272.3 Constructor & Destructor Documentation . . . 1093
12.272.3.1 SequenceOfFragments() . . . 1093
12.272.4 Member Function Documentation . . . 1093
12.272.4.1 AddFragment() . . . 1093
12.272.4.2 Begin() [1/2] . . . 1093

12.272.4.3 Begin() [2/2] . . . 1093
12.272.4.4 Clear() . . . 1093
12.272.4.5 ComputeByteLength() . . . 1094
12.272.4.6 ComputeLength() . . . 1094
12.272.4.7 End() [1/2] . . . 1094
12.272.4.8 End() [2/2] . . . 1094
12.272.4.9 GetBuffer() . . . 1094
12.272.4.10 GetFragBuffer() . . . 1094
12.272.4.11 GetFragment() . . . 1094
12.272.4.12 GetLength() . . . 1095
12.272.4.13 GetNumberOfFragments() . . . 1095
12.272.4.14 GetTable() [1/2] . . . 1095
12.272.4.15 GetTable() [2/2] . . . 1095
12.272.4.16 New() . . . 1095
12.272.4.17 operator==() . . . 1095
12.272.4.18 Print() . . . 1096
12.272.4.19 Read() . . . 1096
12.272.4.20 ReadPreValue() . . . 1096
12.272.4.21 ReadValue() . . . 1096
12.272.4.22 SetLength() . . . 1096
12.272.4.23 Write() . . . 1097
12.272.4.24 WriteBuffer() . . . 1097
12.273 gdcm::SequenceOfItems Class Reference . . . 1097
12.273.1 Detailed Description . . . 1100
12.273.2 Member Typedef Documentation . . . 1100
12.273.2.1 ConstIterator . . . 1100
12.273.2.2 ItemVector . . . 1100
12.273.2.3 Iterator . . . 1101
12.273.2.4 SizeType . . . 1101
12.273.3 Constructor & Destructor Documentation . . . 1101
12.273.3.1 SequenceOfItems() . . . 1101
12.273.4 Member Function Documentation . . . 1101
12.273.4.1 AddItem() . . . 1101
12.273.4.2 AddNewUndefinedLengthItem() . . . 1101
12.273.4.3 Begin() [1/2] . . . 1102
12.273.4.4 Begin() [2/2] . . . 1102
12.273.4.5 Clear() . . . 1102
12.273.4.6 ComputeLength() . . . 1102
12.273.4.7 End() [1/2] . . . 1102

12.273.4.8 End() [2/2] . . . 1102
12.273.4.9 FindDataElement() . . . 1102
12.273.4.10 GetItem() [1/2] . . . 1103
12.273.4.11 GetItem() [2/2] . . . 1103
12.273.4.12 GetLength() . . . 1103
12.273.4.13 GetNumberOfItems() . . . 1103
12.273.4.14 IsEmpty() . . . 1103
12.273.4.15 IsUndefinedLength() . . . 1104
12.273.4.16 New() . . . 1104
12.273.4.17 operator=() . . . 1104
12.273.4.18 operator==() . . . 1104
12.273.4.19 Print() . . . 1104
12.273.4.20 Read() . . . 1105
12.273.4.21 RemoveItemByIndex() . . . 1105
12.273.4.22 SetLength() . . . 1105
12.273.4.23 SetLengthToUndefined() . . . 1105
12.273.4.24 SetNumberOfItems() . . . 1105
12.273.4.25 Write() . . . 1106
12.273.5 Member Data Documentation . . . 1106
12.273.5.1 Items . . . 1106
12.273.5.2 SequenceLengthField . . . 1106
12.274 gdcm::SerieHelper Class Reference . . . 1106
12.274.1 Detailed Description . . . 1108
12.274.2 Member Typedef Documentation . . . 1108
12.274.2.1 Rule . . . 1108
12.274.2.2 SerieRestrictions . . . 1108
12.274.2.3 SingleSerieUIDFileSetmap . . . 1108
12.274.3 Constructor & Destructor Documentation . . . 1109
12.274.3.1 SerieHelper() . . . 1109
12.274.3.2 ~SerieHelper() . . . 1109
12.274.4 Member Function Documentation . . . 1109
12.274.4.1 AddFile() . . . 1109
12.274.4.2 AddFileName() . . . 1109
12.274.4.3 AddRestriction() [1/3] . . . 1109
12.274.4.4 AddRestriction() [2/3] . . . 1109
12.274.4.5 AddRestriction() [3/3] . . . 1109
12.274.4.6 Clear() . . . 1110
12.274.4.7 CreateDefaultUniqueSeriesIdentifier() . . . 1110
12.274.4.8 CreateUniqueSeriesIdentifier() . . . 1110

12.274.4.9 FileNameOrdering() . . . 1110
12.274.4.10 GetFirstSingleSerieUIDFileSet() . . . 1110
12.274.4.11 GetNextSingleSerieUIDFileSet() . . . 1110
12.274.4.12 ImageNumberOrdering() . . . 1110
12.274.4.13 ImagePositionPatientOrdering() . . . 1110
12.274.4.14 OrderFileList() . . . 1110
12.274.4.15 SetDirectory() . . . 1111
12.274.4.16 SetLoadMode() . . . 1111
12.274.4.17 SetUseSeriesDetails() . . . 1111
12.274.4.18 UserOrdering() . . . 1111
12.274.5 Member Data Documentation . . . 1111
12.274.5.1 ItFileSetHt . . . 1111
12.274.5.2 SingleSerieUIDFileSetHT . . . 1111
12.275 gdcm::Series Class Reference . . . 1111
12.275.1 Detailed Description . . . 1112
12.275.2 Constructor & Destructor Documentation . . . 1112
12.275.2.1 Series() . . . 1112
12.276 gdcm::network::ServiceClassApplicationInformation Class Reference . . . 1112
12.276.1 Detailed Description . . . 1112
12.276.2 Constructor & Destructor Documentation . . . 1112
12.276.2.1 ServiceClassApplicationInformation() . . . 1112
12.276.3 Member Function Documentation . . . 1113
12.276.3.1 Print() . . . 1113
12.276.3.2 Read() . . . 1113
12.276.3.3 SetTuple() . . . 1113
12.276.3.4 Size() . . . 1113
12.276.3.5 Write() . . . 1113
12.277 gdcm::ServiceClassUser Class Reference . . . 1114
12.277.1 Detailed Description . . . 1116
12.277.2 Constructor & Destructor Documentation . . . 1117
12.277.2.1 ServiceClassUser() [1/2] . . . 1117
12.277.2.2 ~ServiceClassUser() . . . 1117
12.277.2.3 ServiceClassUser() [2/2] . . . 1117
12.277.3 Member Function Documentation . . . 1117
12.277.3.1 GetAETitle() . . . 1117
12.277.3.2 GetCalledAETitle() . . . 1117
12.277.3.3 GetTimeout() . . . 1117
12.277.3.4 InitializeConnection() . . . 1118
12.277.3.5 IsPresentationContextAccepted() . . . 1118

12.277.3.6 New() . . . 1118
12.277.3.7 operator=() . . . 1118
12.277.3.8 SendEcho() . . . 1118
12.277.3.9 SendFind() . . . 1118
12.277.3.10 SendMove() [1/3] . . . 1119
12.277.3.11 SendMove() [2/3] . . . 1119
12.277.3.12 SendMove() [3/3] . . . 1119
12.277.3.13 SendStore() [1/3] . . . 1119
12.277.3.14 SendStore() [2/3] . . . 1119
12.277.3.15 SendStore() [3/3] . . . 1120
12.277.3.16 SetAETitle() . . . 1120
12.277.3.17 SetCalledAETitle() . . . 1120
12.277.3.18 SetHostname() . . . 1120
12.277.3.19 SetPort() . . . 1120
12.277.3.20 SetPortSCP() . . . 1121
12.277.3.21 SetPresentationContexts() . . . 1121
12.277.3.22 SetTimeout() . . . 1121
12.277.3.23 StartAssociation() . . . 1121
12.277.3.24 StopAssociation() . . . 1122
12.278 gdcm::SHA1 Class Reference . . . 1122
12.278.1 Detailed Description . . . 1122
12.278.2 Constructor & Destructor Documentation . . . 1123
12.278.2.1 SHA1() [1/2] . . . 1123
12.278.2.2 ∼SHA1() . . . 1123
12.278.2.3 SHA1() [2/2] . . . 1123
12.278.3 Member Function Documentation . . . 1123
12.278.3.1 Compute() . . . 1123
12.278.3.2 ComputeFile() . . . 1123
12.278.3.3 operator=() . . . 1123
12.279 gdcm::SimpleMemberCommand< T > Class Template Reference . . . 1124
12.279.1 Detailed Description . . . 1127
12.279.2 Member Typedef Documentation . . . 1127
12.279.2.1 Self . . . 1127
12.279.2.2 TMemberFunctionPointer . . . 1127
12.279.3 Constructor & Destructor Documentation . . . 1127
12.279.3.1 SimpleMemberCommand() [1/2] . . . 1127
12.279.3.2 SimpleMemberCommand() [2/2] . . . 1127
12.279.3.3 ∼SimpleMemberCommand() . . . 1127
12.279.4 Member Function Documentation . . . 1128

12.279.4.1 Execute() [1/2] . . . 1128
12.279.4.2 Execute() [2/2] . . . 1128
12.279.4.3 New() . . . 1128
12.279.4.4 operator=() . . . 1128
12.279.4.5 SetCallbackFunction() . . . 1128
12.279.5 Member Data Documentation . . . 1129
12.279.5.1 m_MemberFunction . . . 1129
12.279.5.2 m_This . . . 1129
12.280 gdcm::SimpleSubjectWatcher Class Reference . . . 1129
12.280.1 Detailed Description . . . 1130
12.280.2 Constructor & Destructor Documentation . . . 1130
12.280.2.1 SimpleSubjectWatcher() [1/2] . . . 1130
12.280.2.2 ∼SimpleSubjectWatcher() . . . 1130
12.280.2.3 SimpleSubjectWatcher() [2/2] . . . 1130
12.280.3 Member Function Documentation . . . 1130
12.280.3.1 EndFilter() . . . 1130
12.280.3.2 operator=() . . . 1130
12.280.3.3 ShowAbort() . . . 1131
12.280.3.4 ShowAnonymization() . . . 1131
12.280.3.5 ShowData() . . . 1131
12.280.3.6 ShowDataSet() . . . 1131
12.280.3.7 ShowFileName() . . . 1131
12.280.3.8 ShowIteration() . . . 1131
12.280.3.9 ShowProgress() . . . 1131
12.280.3.10 StartFilter() . . . 1132
12.280.3.11 TestAbortOff() . . . 1132
12.280.3.12 TestAbortOn() . . . 1132
12.281 gdcm::MrProtocol::Slice Struct Reference . . . 1132
12.281.1 Member Data Documentation . . . 1133
12.281.1.1 Normal . . . 1133
12.281.1.2 Position . . . 1133
12.282 gdcm::MrProtocol::SliceArray Struct Reference . . . 1133
12.282.1 Member Data Documentation . . . 1134
12.282.1.1 Slices . . . 1134
12.283 gdcm::SmartPointer< ObjectType > Class Template Reference . . . 1134
12.283.1 Detailed Description . . . 1135
12.283.2 Constructor & Destructor Documentation . . . 1136
12.283.2.1 SmartPointer() [1/4] . . . 1136
12.283.2.2 SmartPointer() [2/4] . . . 1136

12.283.2.3 SmartPointer() [3/4] . . . 1136
12.283.2.4 SmartPointer() [4/4] . . . 1136
12.283.2.5 ∼SmartPointer() . . . 1136
12.283.3 Member Function Documentation . . . 1136
12.283.3.1 GetPointer() . . . 1136
12.283.3.2 operator ObjectType ∗() . . . 1137
12.283.3.3 operator∗() . . . 1137
12.283.3.4 operator->() . . . 1137
12.283.3.5 operator=() [1/3] . . . 1137
12.283.3.6 operator=() [2/3] . . . 1137
12.283.3.7 operator=() [3/3] . . . 1137
12.284 gdcm::network::SOPClassExtendedNegociationSub Class Reference . . . 1138
12.284.1 Detailed Description . . . 1138
12.284.2 Constructor & Destructor Documentation . . . 1138
12.284.2.1 SOPClassExtendedNegociationSub() . . . 1138
12.284.3 Member Function Documentation . . . 1138
12.284.3.1 Print() . . . 1138
12.284.3.2 Read() . . . 1138
12.284.3.3 SetTuple() . . . 1139
12.284.3.4 Size() . . . 1139
12.284.3.5 Write() . . . 1139
12.285 gdcm::SOPClassUIDToIOD Class Reference . . . 1139
12.285.1 Detailed Description . . . 1140
12.285.2 Member Typedef Documentation . . . 1140
12.285.2.1 const . . . 1140
12.285.3 Member Function Documentation . . . 1140
12.285.3.1 GetIOD() . . . 1140
12.285.3.2 GetIODFromSOPClassUID() . . . 1140
12.285.3.3 GetNumberOfSOPClassToIOD() . . . 1140
12.285.3.4 GetSOPClassUIDFromIOD() . . . 1140
12.285.3.5 GetSOPClassUIDToIOD() . . . 1141
12.285.3.6 GetSOPClassUIDToIODs() . . . 1141
12.286 gdcm::Sorter Class Reference . . . 1141
12.286.1 Detailed Description . . . 1142
12.286.2 Member Typedef Documentation . . . 1143
12.286.2.1 SelectionMap . . . 1143
12.286.2.2 SortFunction . . . 1143
12.286.3 Constructor & Destructor Documentation . . . 1143
12.286.3.1 Sorter() . . . 1143

12.286.3.2 ∼Sorter() . . . 1143
12.286.4 Member Function Documentation . . . 1143
12.286.4.1 AddSelect() . . . 1143
12.286.4.2 GetFilenames() . . . 1143
12.286.4.3 Print() . . . 1144
12.286.4.4 SetSortFunction() . . . 1144
12.286.4.5 SetTagsToRead() . . . 1144
12.286.4.6 Sort() . . . 1144
12.286.4.7 StableSort() . . . 1145
12.286.5 Friends And Related Symbol Documentation . . . 1145
12.286.5.1 operator<< . . . 1145
12.286.6 Member Data Documentation . . . 1145
12.286.6.1 Filenames . . . 1145
12.286.6.2 Selection . . . 1145
12.286.6.3 SortFunc . . . 1145
12.286.6.4 TagsToRead . . . 1145
12.287 gdcm::Spacing Class Reference . . . 1146
12.287.1 Detailed Description . . . 1146
12.287.2 Member Enumeration Documentation . . . 1147
12.287.2.1 SpacingType . . . 1147
12.287.3 Constructor & Destructor Documentation . . . 1147
12.287.3.1 Spacing() . . . 1147
12.287.3.2 ∼Spacing() . . . 1148
12.287.4 Member Function Documentation . . . 1148
12.287.4.1 ComputePixelAspectRatioFromPixelSpacing() . . . 1148
12.288 gdcm::Spectroscopy Class Reference . . . 1148
12.288.1 Detailed Description . . . 1148
12.288.2 Constructor & Destructor Documentation . . . 1148
12.288.2.1 Spectroscopy() . . . 1148
12.289 gdcm::SplitMosaicFilter Class Reference . . . 1149
12.289.1 Detailed Description . . . 1149
12.289.2 Constructor & Destructor Documentation . . . 1150
12.289.2.1 SplitMosaicFilter() . . . 1150
12.289.2.2 ∼SplitMosaicFilter() . . . 1150
12.289.3 Member Function Documentation . . . 1150
12.289.3.1 ComputeMOSAICDimensions() . . . 1150
12.289.3.2 ComputeMOSAICImagePositionPatient() . . . 1150
12.289.3.3 ComputeMOSAICSliceNormal() . . . 1150
12.289.3.4 ComputeMOSAICSlicePosition() . . . 1150

12.289.3.5 GetAcquisitionSize() . . . 1151
12.289.3.6 GetFile() [1/2] . . . 1151
12.289.3.7 GetFile() [2/2] . . . 1151
12.289.3.8 GetImage() [1/2] . . . 1151
12.289.3.9 GetImage() [2/2] . . . 1151
12.289.3.10 GetNumberOfImagesInMosaic() . . . 1151
12.289.3.11 SetFile() . . . 1151
12.289.3.12 SetImage() . . . 1151
12.289.3.13 Split() . . . 1152
12.290 gdcm::StartEvent Class Reference . . . 1152
12.291 gdcm::static_assert_test< x > Struct Template Reference . . . 1153
12.292 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference . . . 1154
12.293 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Reference . . . 1154
12.293.1 Member Enumeration Documentation . . . 1155
12.293.1.1 anonymous enum . . . 1155
12.294 gdcm::StreamImageReader Class Reference . . . 1155
12.294.1 Detailed Description . . . 1156
12.294.2 Constructor & Destructor Documentation . . . 1156
12.294.2.1 StreamImageReader() . . . 1156
12.294.2.2 ~StreamImageReader() . . . 1156
12.294.3 Member Function Documentation . . . 1157
12.294.3.1 CanReadImage() . . . 1157
12.294.3.2 DefinePixelExtent() . . . 1157
12.294.3.3 DefineProperBufferLength() . . . 1157
12.294.3.4 GetDimensionsValueForResolution() . . . 1158
12.294.3.5 GetFile() . . . 1158
12.294.3.6 Read() . . . 1158
12.294.3.7 ReadImageInformation() . . . 1158
12.294.3.8 SetFileName() . . . 1159
12.294.3.9 SetStream() . . . 1159
12.295 gdcm::StreamImageWriter Class Reference . . . 1159
12.295.1 Detailed Description . . . 1161
12.295.2 Constructor & Destructor Documentation . . . 1161
12.295.2.1 StreamImageWriter() . . . 1161
12.295.2.2 ~StreamImageWriter() . . . 1161
12.295.3 Member Function Documentation . . . 1162
12.295.3.1 CanWriteFile() . . . 1162
12.295.3.2 DefinePixelExtent() . . . 1162
12.295.3.3 DefineProperBufferLength() . . . 1162

12.295.3.4 SetFile() . . . 1163
12.295.3.5 SetFileName() . . . 1163
12.295.3.6 SetStream() . . . 1163
12.295.3.7 Write() . . . 1163
12.295.3.8 WriteImageInformation() . . . 1164
12.295.3.9 WriteImageSubregionRAW() . . . 1164
12.295.3.10 WriteRawHeader() . . . 1164
12.295.4 Member Data Documentation . . . 1164
12.295.4.1 mElementOffsets . . . 1164
12.295.4.2 mElementOffsets1 . . . 1164
12.295.4.3 mspFile . . . 1165
12.295.4.4 mWriter . . . 1165
12.295.4.5 mXMax . . . 1165
12.295.4.6 mXMin . . . 1165
12.295.4.7 mYMax . . . 1165
12.295.4.8 mYMin . . . 1165
12.295.4.9 mZMax . . . 1165
12.295.4.10 mZMin . . . 1165
12.296 gdcm::StrictScanner Class Reference . . . 1166
12.296.1 Detailed Description . . . 1169
12.296.2 Member Typedef Documentation . . . 1169
12.296.2.1 ConstIterator . . . 1169
12.296.2.2 MappingType . . . 1169
12.296.2.3 TagToValue . . . 1170
12.296.2.4 TagToValueValueType . . . 1170
12.296.2.5 ValuesType . . . 1170
12.296.3 Constructor & Destructor Documentation . . . 1170
12.296.3.1 StrictScanner() . . . 1170
12.296.3.2 ~StrictScanner() . . . 1170
12.296.4 Member Function Documentation . . . 1170
12.296.4.1 AddPrivateTag() . . . 1170
12.296.4.2 AddSkipTag() . . . 1171
12.296.4.3 AddTag() . . . 1171
12.296.4.4 Begin() . . . 1171
12.296.4.5 ClearSkipTags() . . . 1171
12.296.4.6 ClearTags() . . . 1171
12.296.4.7 End() . . . 1171
12.296.4.8 GetAllFilenamesFromTagToValue() . . . 1171
12.296.4.9 GetFilenameFromTagToValue() . . . 1172

12.296.4.10 GetFilenames() . . . 1172
12.296.4.11 GetKeys() . . . 1172
12.296.4.12 GetMapping() . . . 1172
12.296.4.13 GetMappingFromTagToValue() . . . 1172
12.296.4.14 GetMappings() . . . 1172
12.296.4.15 GetOrderedValues() . . . 1173
12.296.4.16 GetValue() . . . 1173
12.296.4.17 GetValues() [1/2] . . . 1173
12.296.4.18 GetValues() [2/2] . . . 1173
12.296.4.19 IsKey() . . . 1173
12.296.4.20 New() . . . 1174
12.296.4.21 Print() . . . 1174
12.296.4.22 PrintTable() . . . 1174
12.296.4.23 ProcessPublicTag() . . . 1174
12.296.4.24 Scan() . . . 1174
12.296.5 Friends And Related Symbol Documentation . . . 1175
12.296.5.1 operator<< . . . 1175
12.297 gdcm::StrictScanner2 Class Reference . . . 1175
12.297.1 Detailed Description . . . 1178
12.297.2 Member Typedef Documentation . . . 1179
12.297.2.1 PrivateConstIterator . . . 1179
12.297.2.2 PrivateMappingType . . . 1179
12.297.2.3 PrivateTagToValue . . . 1179
12.297.2.4 PrivateTagToValueValueType . . . 1179
12.297.2.5 PublicConstIterator . . . 1179
12.297.2.6 PublicMappingType . . . 1179
12.297.2.7 PublicTagToValue . . . 1179
12.297.2.8 PublicTagToValueValueType . . . 1179
12.297.2.9 ValuesType . . . 1180
12.297.3 Constructor & Destructor Documentation . . . 1180
12.297.3.1 StrictScanner2() . . . 1180
12.297.3.2 ~StrictScanner2() . . . 1180
12.297.4 Member Function Documentation . . . 1180
12.297.4.1 AddPrivateTag() . . . 1180
12.297.4.2 AddPublicTag() . . . 1180
12.297.4.3 AddSkipTag() . . . 1180
12.297.4.4 Begin() . . . 1180
12.297.4.5 ClearPrivateTags() . . . 1181
12.297.4.6 ClearPublicTags() . . . 1181

12.297.4.7 ClearSkipTags() . . . 1181
12.297.4.8 End() . . . 1181
12.297.4.9 GetAllFilenamesFromPrivateTagToValue() . . . 1181
12.297.4.10 GetAllFilenamesFromPublicTagToValue() . . . 1181
12.297.4.11 GetFilenameFromPrivateTagToValue() . . . 1181
12.297.4.12 GetFilenameFromPublicTagToValue() . . . 1181
12.297.4.13 GetFilenames() . . . 1182
12.297.4.14 GetKeys() . . . 1182
12.297.4.15 GetMappingFromPrivateTagToValue() . . . 1182
12.297.4.16 GetMappingFromPublicTagToValue() . . . 1182
12.297.4.17 GetPrivateMapping() . . . 1182
12.297.4.18 GetPrivateMappings() . . . 1182
12.297.4.19 GetPrivateOrderedValues() . . . 1182
12.297.4.20 GetPrivateValue() . . . 1183
12.297.4.21 GetPrivateValues() . . . 1183
12.297.4.22 GetPublicMapping() . . . 1183
12.297.4.23 GetPublicMappings() . . . 1183
12.297.4.24 GetPublicOrderedValues() . . . 1183
12.297.4.25 GetPublicValue() . . . 1183
12.297.4.26 GetPublicValues() . . . 1184
12.297.4.27 GetValues() . . . 1184
12.297.4.28 IsKey() . . . 1184
12.297.4.29 New() . . . 1184
12.297.4.30 Print() . . . 1184
12.297.4.31 PrintTable() . . . 1184
12.297.4.32 PrivateBegin() . . . 1185
12.297.4.33 PrivateEnd() . . . 1185
12.297.4.34 ProcessPrivateTag() . . . 1185
12.297.4.35 ProcessPublicTag() . . . 1185
12.297.4.36 Scan() . . . 1185
12.297.5 Friends And Related Symbol Documentation . . . 1185
12.297.5.1 operator<< . . . 1185
12.298 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference . . . 1186
12.298.1 Detailed Description . . . 1187
12.298.2 Member Typedef Documentation . . . 1187
12.298.2.1 const_iterator . . . 1187
12.298.2.2 const_reference . . . 1188
12.298.2.3 const_reverse_iterator . . . 1188
12.298.2.4 difference_type . . . 1188

12.298.2.5 iterator . . . 1188
12.298.2.6 pointer . . . 1188
12.298.2.7 reference . . . 1188
12.298.2.8 reverse_iterator . . . 1188
12.298.2.9 size_type . . . 1188
12.298.2.10 value_type . . . 1189
12.298.3 Constructor & Destructor Documentation . . . 1189
12.298.3.1 String() [1/4] . . . 1189
12.298.3.2 String() [2/4] . . . 1189
12.298.3.3 String() [3/4] . . . 1189
12.298.3.4 String() [4/4] . . . 1189
12.298.4 Member Function Documentation . . . 1189
12.298.4.1 IsValid() . . . 1189
12.298.4.2 operator const char *() . . . 1190
12.298.4.3 Trim() [1/2] . . . 1190
12.298.4.4 Trim() [2/2] . . . 1190
12.298.4.5 Truncate() . . . 1190
12.299 gdcm::StringFilter Class Reference . . . 1190
12.299.1 Detailed Description . . . 1191
12.299.2 Constructor & Destructor Documentation . . . 1191
12.299.2.1 StringFilter() . . . 1191
12.299.2.2 ~StringFilter() . . . 1192
12.299.3 Member Function Documentation . . . 1192
12.299.3.1 ExecuteQuery() [1/2] . . . 1192
12.299.3.2 ExecuteQuery() [2/2] . . . 1192
12.299.3.3 FromString() . . . 1192
12.299.3.4 GetFile() [1/2] . . . 1192
12.299.3.5 GetFile() [2/2] . . . 1192
12.299.3.6 SetDicts() . . . 1192
12.299.3.7 SetFile() . . . 1193
12.299.3.8 ToString() [1/3] . . . 1193
12.299.3.9 ToString() [2/3] . . . 1193
12.299.3.10 ToString() [3/3] . . . 1193
12.299.3.11 ToStringPair() [1/3] . . . 1193
12.299.3.12 ToStringPair() [2/3] . . . 1194
12.299.3.13 ToStringPair() [3/3] . . . 1194
12.299.3.14 UseDictAlways() . . . 1194
12.300 gdcm::Study Class Reference . . . 1194
12.300.1 Detailed Description . . . 1194

12.300.2 Constructor & Destructor Documentation . . . 1194
12.300.2.1 Study() . . . 1194
12.301 gdcm::Subject Class Reference . . . 1195
12.301.1 Detailed Description . . . 1196
12.301.2 Constructor & Destructor Documentation . . . 1196
12.301.2.1 Subject() . . . 1196
12.301.2.2 ~Subject() . . . 1197
12.301.3 Member Function Documentation . . . 1197
12.301.3.1 AddObserver() [1/2] . . . 1197
12.301.3.2 AddObserver() [2/2] . . . 1197
12.301.3.3 GetCommand() . . . 1197
12.301.3.4 HasObserver() . . . 1197
12.301.3.5 InvokeEvent() [1/2] . . . 1197
12.301.3.6 InvokeEvent() [2/2] . . . 1198
12.301.3.7 RemoveAllObservers() . . . 1198
12.301.3.8 RemoveObserver() . . . 1198
12.302 gdcm::Surface Class Reference . . . 1198
12.302.1 Detailed Description . . . 1201
12.302.2 Member Enumeration Documentation . . . 1201
12.302.2.1 STATES . . . 1201
12.302.2.2 VIEWType . . . 1202
12.302.3 Constructor & Destructor Documentation . . . 1202
12.302.3.1 Surface() . . . 1202
12.302.3.2 ~Surface() . . . 1202
12.302.4 Member Function Documentation . . . 1202
12.302.4.1 GetAlgorithmFamily() [1/2] . . . 1202
12.302.4.2 GetAlgorithmFamily() [2/2] . . . 1203
12.302.4.3 GetAlgorithmName() . . . 1203
12.302.4.4 GetAlgorithmVersion() . . . 1203
12.302.4.5 GetAxisOfRotation() . . . 1203
12.302.4.6 GetCenterOfRotation() . . . 1203
12.302.4.7 GetFiniteVolume() . . . 1203
12.302.4.8 GetManifold() . . . 1203
12.302.4.9 GetMaximumPointDistance() . . . 1203
12.302.4.10 GetMeanPointDistance() . . . 1204
12.302.4.11 GetMeshPrimitive() [1/2] . . . 1204
12.302.4.12 GetMeshPrimitive() [2/2] . . . 1204
12.302.4.13 GetNumberOfSurfacePoints() . . . 1204
12.302.4.14 GetNumberOfVectors() . . . 1204

12.302.4.15 GetPointCoordinatesData() [1/2] 1204
12.302.4.16 GetPointCoordinatesData() [2/2] 1204
12.302.4.17 GetPointPositionAccuracy() 1204
12.302.4.18 GetPointsBoundingBoxCoordinates() 1205
12.302.4.19 GetProcessingAlgorithm() [1/2] 1205
12.302.4.20 GetProcessingAlgorithm() [2/2] 1205
12.302.4.21 GetRecommendedDisplayCIELabValue() [1/2] 1205
12.302.4.22 GetRecommendedDisplayCIELabValue() [2/2] 1205
12.302.4.23 GetRecommendedDisplayGrayscaleValue() 1205
12.302.4.24 GetRecommendedPresentationOpacity() 1205
12.302.4.25 GetRecommendedPresentationType() 1205
12.302.4.26 GetSTATES() 1206
12.302.4.27 GetSTATESString() 1206
12.302.4.28 GetSurfaceComments() 1206
12.302.4.29 GetSurfaceNumber() 1206
12.302.4.30 GetSurfaceProcessing() 1206
12.302.4.31 GetSurfaceProcessingDescription() 1206
12.302.4.32 GetSurfaceProcessingRatio() 1206
12.302.4.33 GetVectorAccuracy() 1206
12.302.4.34 GetVectorCoordinateData() [1/2] 1206
12.302.4.35 GetVectorCoordinateData() [2/2] 1207
12.302.4.36 GetVectorDimensionality() 1207
12.302.4.37 GetVIEWType() 1207
12.302.4.38 GetVIEWTypeString() 1207
12.302.4.39 SetAlgorithmFamily() 1207
12.302.4.40 SetAlgorithmName() 1207
12.302.4.41 SetAlgorithmVersion() 1207
12.302.4.42 SetAxisOfRotation() 1207
12.302.4.43 SetCenterOfRotation() 1208
12.302.4.44 SetFiniteVolume() 1208
12.302.4.45 SetManifold() 1208
12.302.4.46 SetMaximumPointDistance() 1208
12.302.4.47 SetMeanPointDistance() 1208
12.302.4.48 SetMeshPrimitive() 1208
12.302.4.49 SetNumberOfSurfacePoints() 1208
12.302.4.50 SetNumberOfVectors() 1208
12.302.4.51 SetPointCoordinatesData() 1209
12.302.4.52 SetPointPositionAccuracy() 1209
12.302.4.53 SetPointsBoundingBoxCoordinates() 1209

12.302.4.54 SetProcessingAlgorithm() . . . 1209
12.302.4.55 SetRecommendedDisplayCIELabValue() [1/3] . . . 1209
12.302.4.56 SetRecommendedDisplayCIELabValue() [2/3] . . . 1209
12.302.4.57 SetRecommendedDisplayCIELabValue() [3/3] . . . 1209
12.302.4.58 SetRecommendedDisplayGrayscaleValue() . . . 1209
12.302.4.59 SetRecommendedPresentationOpacity() . . . 1210
12.302.4.60 SetRecommendedPresentationType() . . . 1210
12.302.4.61 SetSurfaceComments() . . . 1210
12.302.4.62 SetSurfaceNumber() . . . 1210
12.302.4.63 SetSurfaceProcessing() . . . 1210
12.302.4.64 SetSurfaceProcessingDescription() . . . 1210
12.302.4.65 SetSurfaceProcessingRatio() . . . 1210
12.302.4.66 SetVectorAccuracy() . . . 1210
12.302.4.67 SetVectorCoordinateData() . . . 1211
12.302.4.68 SetVectorDimensionality() . . . 1211
12.303 gdcm::SurfaceHelper Class Reference . . . 1211
12.303.1 Detailed Description . . . 1212
12.303.2 Member Typedef Documentation . . . 1212
12.303.2.1 ColorArray . . . 1212
12.303.3 Member Function Documentation . . . 1212
12.303.3.1 RecommendedDisplayCIELabToRGB() [1/2] . . . 1212
12.303.3.2 RecommendedDisplayCIELabToRGB() [2/2] . . . 1213
12.303.3.3 RGBToRecommendedDisplayCIELab() . . . 1213
12.303.3.4 RGBToRecommendedDisplayGrayscale() . . . 1214
12.304 gdcm::SurfaceReader Class Reference . . . 1214
12.304.1 Detailed Description . . . 1218
12.304.2 Constructor & Destructor Documentation . . . 1218
12.304.2.1 SurfaceReader() . . . 1218
12.304.2.2 ∼SurfaceReader() . . . 1218
12.304.3 Member Function Documentation . . . 1218
12.304.3.1 GetNumberOfSurfaces() . . . 1218
12.304.3.2 Read() . . . 1219
12.304.3.3 ReadPointMacro() . . . 1219
12.304.3.4 ReadSurface() . . . 1219
12.304.3.5 ReadSurfaces() . . . 1219
12.305 gdcm::SurfaceWriter Class Reference . . . 1219
12.305.1 Detailed Description . . . 1223
12.305.2 Constructor & Destructor Documentation . . . 1223
12.305.2.1 SurfaceWriter() . . . 1223

12.305.2.2 ~SurfaceWriter() . . . 1223
12.305.3 Member Function Documentation . . . 1223
12.305.3.1 ComputeNumberOfSurfaces() . . . 1223
12.305.3.2 GetNumberOfSurfaces() . . . 1224
12.305.3.3 PrepareWrite() . . . 1224
12.305.3.4 PrepareWritePointMacro() . . . 1224
12.305.3.5 SetNumberOfSurfaces() . . . 1224
12.305.3.6 Write() . . . 1224
12.305.4 Member Data Documentation . . . 1224
12.305.4.1 NumberOfSurfaces . . . 1224
12.306 gdcm::SwapCode Class Reference . . . 1224
12.306.1 Detailed Description . . . 1225
12.306.2 Member Enumeration Documentation . . . 1225
12.306.2.1 SwapCodeType . . . 1225
12.306.3 Constructor & Destructor Documentation . . . 1226
12.306.3.1 SwapCode() . . . 1226
12.306.4 Member Function Documentation . . . 1226
12.306.4.1 GetIndex() . . . 1226
12.306.4.2 GetSwapCodeString() . . . 1226
12.306.4.3 operator SwapCode::SwapCodeType() . . . 1226
12.306.5 Friends And Related Symbol Documentation . . . 1227
12.306.5.1 operator<< . . . 1227
12.307 gdcm::SwapperDoOp Class Reference . . . 1227
12.307.1 Member Function Documentation . . . 1227
12.307.1.1 Swap() . . . 1227
12.307.1.2 SwapArray() . . . 1227
12.308 gdcm::SwapperNoOp Class Reference . . . 1228
12.308.1 Detailed Description . . . 1228
12.308.2 Member Function Documentation . . . 1228
12.308.2.1 Swap() . . . 1228
12.308.2.2 SwapArray() . . . 1228
12.309 gdcm::System Class Reference . . . 1228
12.309.1 Detailed Description . . . 1230
12.309.2 Member Function Documentation . . . 1230
12.309.2.1 ConvertToUNC() . . . 1230
12.309.2.2 DeleteDirectory() . . . 1230
12.309.2.3 EncodeBytes() . . . 1230
12.309.2.4 FileExists() . . . 1230
12.309.2.5 FileIsDirectory() . . . 1231

12.309.2.6 FileIsSymlink() . . . 1231
12.309.2.7 FileSize() . . . 1231
12.309.2.8 FileTime() . . . 1231
12.309.2.9 FormatDateTime() . . . 1232
12.309.2.10 GetCurrentDateTime() . . . 1232
12.309.2.11 GetCurrentModuleFileName() . . . 1232
12.309.2.12 GetCurrentProcessFileName() . . . 1232
12.309.2.13 GetCurrentResourcesDirectory() . . . 1232
12.309.2.14 GetCWD() . . . 1232
12.309.2.15 GetHostName() . . . 1233
12.309.2.16 GetLastSystemError() . . . 1233
12.309.2.17 GetLocaleCharset() . . . 1233
12.309.2.18 GetPermissions() . . . 1233
12.309.2.19 GetTimezoneOffsetFromUTC() . . . 1233
12.309.2.20 MakeDirectory() . . . 1233
12.309.2.21 ParseDateTime() [1/2] . . . 1234
12.309.2.22 ParseDateTime() [2/2] . . . 1234
12.309.2.23 RemoveFile() . . . 1234
12.309.2.24 SetPermissions() . . . 1234
12.309.2.25 StrCaseCmp() . . . 1234
12.309.2.26 StrNCaseCmp() . . . 1235
12.309.2.27 StrSep() . . . 1235
12.309.2.28 StrTokR() . . . 1235
12.310 gdcm::Table Class Reference . . . 1235
12.310.1 Detailed Description . . . 1237
12.310.2 Member Typedef Documentation . . . 1237
12.310.2.1 MapTableEntry . . . 1237
12.310.3 Constructor & Destructor Documentation . . . 1237
12.310.3.1 Table() [1/2] . . . 1237
12.310.3.2 ~Table() . . . 1237
12.310.3.3 Table() [2/2] . . . 1237
12.310.4 Member Function Documentation . . . 1237
12.310.4.1 GetTableEntry() . . . 1237
12.310.4.2 InsertEntry() . . . 1238
12.310.4.3 operator=() . . . 1238
12.310.5 Friends And Related Symbol Documentation . . . 1238
12.310.5.1 operator<< . . . 1238
12.310.6 Member Data Documentation . . . 1238
12.310.6.1 TableInternal . . . 1238

12.311 gdcm::TableEntry Class Reference . . . 1238
12.311.1 Detailed Description . . . 1239
12.311.2 Constructor & Destructor Documentation . . . 1239
12.311.2.1 TableEntry() . . . 1239
12.311.2.2 ∼TableEntry() . . . 1239
12.312 gdcm::TableReader Class Reference . . . 1239
12.312.1 Detailed Description . . . 1240
12.312.2 Constructor & Destructor Documentation . . . 1240
12.312.2.1 TableReader() . . . 1240
12.312.2.2 ∼TableReader() . . . 1240
12.312.3 Member Function Documentation . . . 1241
12.312.3.1 CharacterDataHandler() . . . 1241
12.312.3.2 EndElement() . . . 1241
12.312.3.3 GetDefs() . . . 1241
12.312.3.4 GetFilename() . . . 1241
12.312.3.5 HandleIOD() . . . 1241
12.312.3.6 HandleIODEntry() . . . 1241
12.312.3.7 HandleMacro() . . . 1241
12.312.3.8 HandleMacroEntry() . . . 1242
12.312.3.9 HandleMacroEntryDescription() . . . 1242
12.312.3.10 HandleModule() . . . 1242
12.312.3.11 HandleModuleEntry() . . . 1242
12.312.3.12 HandleModuleEntryDescription() . . . 1242
12.312.3.13 HandleModuleInclude() . . . 1242
12.312.3.14 Read() . . . 1242
12.312.3.15 SetFilename() . . . 1242
12.312.3.16 StartElement() . . . 1243
12.313 gdcm::network::TableRow Class Reference . . . 1243
12.313.1 Constructor & Destructor Documentation . . . 1244
12.313.1.1 TableRow() . . . 1244
12.313.1.2 ∼TableRow() . . . 1244
12.313.2 Member Data Documentation . . . 1244
12.313.2.1 transitions . . . 1244
12.314 gdcm::Tag Class Reference . . . 1244
12.314.1 Detailed Description . . . 1246
12.314.2 Constructor & Destructor Documentation . . . 1247
12.314.2.1 Tag() [1/3] . . . 1247
12.314.2.2 Tag() [2/3] . . . 1247
12.314.2.3 Tag() [3/3] . . . 1247

12.314.3 Member Function Documentation . . . 1247
12.314.3.1 GetElement() . . . 1247
12.314.3.2 GetElementTag() . . . 1248
12.314.3.3 GetGroup() . . . 1248
12.314.3.4 GetLength() . . . 1248
12.314.3.5 GetPrivateCreator() . . . 1248
12.314.3.6 IsGroupLength() . . . 1248
12.314.3.7 IsGroupXX() . . . 1249
12.314.3.8 IsIllegal() . . . 1249
12.314.3.9 IsPrivate() . . . 1249
12.314.3.10 IsPrivateCreator() . . . 1249
12.314.3.11 IsPublic() . . . 1250
12.314.3.12 operator"!=() . . . 1250
12.314.3.13 operator<() . . . 1250
12.314.3.14 operator<=() . . . 1250
12.314.3.15 operator=() . . . 1250
12.314.3.16 operator==() . . . 1250
12.314.3.17 operator[]() [1/2] . . . 1251
12.314.3.18 operator[]() [2/2] . . . 1251
12.314.3.19 PrintAsContinuousString() . . . 1251
12.314.3.20 PrintAsContinuousUpperCaseString() . . . 1251
12.314.3.21 PrintAsPipeSeparatedString() . . . 1251
12.314.3.22 Read() . . . 1251
12.314.3.23 ReadFromCommaSeparatedString() . . . 1252
12.314.3.24 ReadFromContinuousString() . . . 1252
12.314.3.25 ReadFromPipeSeparatedString() . . . 1252
12.314.3.26 SetElement() . . . 1252
12.314.3.27 SetElementTag() [1/2] . . . 1253
12.314.3.28 SetElementTag() [2/2] . . . 1253
12.314.3.29 SetGroup() . . . 1253
12.314.3.30 SetPrivateCreator() . . . 1253
12.314.3.31 Write() . . . 1253
12.314.4 Friends And Related Symbol Documentation . . . 1254
12.314.4.1 operator<< . . . 1254
12.314.4.2 operator>> . . . 1254
12.314.5 Member Data Documentation . . . 1254
12.314.5.1 bytes . . . 1254
12.314.5.2 tag . . . 1254
12.314.5.3 tags . . . 1254

12.315 gdcm::TagPath Class Reference . . . 1255
12.315.1 Detailed Description . . . 1255
12.315.2 Constructor & Destructor Documentation . . . 1255
12.315.2.1 TagPath() . . . 1255
12.315.2.2 ∼TagPath() . . . 1255
12.315.3 Member Function Documentation . . . 1256
12.315.3.1 ConstructFromString() . . . 1256
12.315.3.2 ConstructFromTagList() . . . 1256
12.315.3.3 IsValid() . . . 1256
12.315.3.4 Print() . . . 1256
12.315.3.5 Push() [1/2] . . . 1256
12.315.3.6 Push() [2/2] . . . 1256
12.316 gdcm::Testing Class Reference . . . 1257
12.316.1 Detailed Description . . . 1258
12.316.2 Member Typedef Documentation . . . 1258
12.316.2.1 MD5DataImagesType . . . 1258
12.316.2.2 MediaStorageDataFilesType . . . 1258
12.316.3 Constructor & Destructor Documentation . . . 1258
12.316.3.1 Testing() . . . 1258
12.316.3.2 ∼Testing() . . . 1258
12.316.4 Member Function Documentation . . . 1259
12.316.4.1 ComputeFileMD5() . . . 1259
12.316.4.2 ComputeMD5() . . . 1259
12.316.4.3 GetDataExtraRoot() . . . 1259
12.316.4.4 GetDataRoot() . . . 1259
12.316.4.5 GetFileName() . . . 1260
12.316.4.6 GetFileNames() . . . 1260
12.316.4.7 GetLossyFlagFromFile() . . . 1260
12.316.4.8 GetMD5DataImage() . . . 1260
12.316.4.9 GetMD5DataImages() . . . 1260
12.316.4.10 GetMD5FromBrokenFile() . . . 1260
12.316.4.11 GetMD5FromFile() . . . 1261
12.316.4.12 GetMediaStorageDataFile() . . . 1261
12.316.4.13 GetMediaStorageDataFiles() . . . 1261
12.316.4.14 GetMediaStorageFromFile() . . . 1261
12.316.4.15 GetNumberOfFileNames() . . . 1261
12.316.4.16 GetNumberOfMD5DataImages() . . . 1261
12.316.4.17 GetNumberOfMediaStorageDataFiles() . . . 1261
12.316.4.18 GetPixelSpacingDataRoot() . . . 1262

12.316.4.19 GetSelectedPrivateGroupOffsetFromFile() . . . 1262
12.316.4.20 GetSelectedTagsOffsetFromFile() . . . 1262
12.316.4.21 GetSourceDirectory() . . . 1262
12.316.4.22 GetStreamOffsetFromFile() . . . 1262
12.316.4.23 GetTempDirectory() . . . 1262
12.316.4.24 GetTempDirectoryW() . . . 1263
12.316.4.25 GetTempFilename() . . . 1263
12.316.4.26 GetTempFilenameW() . . . 1263
12.316.4.27 Print() . . . 1263
12.317 gdcm::Trace Class Reference . . . 1263
12.317.1 Detailed Description . . . 1264
12.317.2 Constructor & Destructor Documentation . . . 1265
12.317.2.1 Trace() . . . 1265
12.317.2.2 ∼Trace() . . . 1265
12.317.3 Member Function Documentation . . . 1265
12.317.3.1 DebugOff() . . . 1265
12.317.3.2 DebugOn() . . . 1265
12.317.3.3 ErrorOff() . . . 1265
12.317.3.4 ErrorOn() . . . 1265
12.317.3.5 GetDebugFlag() . . . 1266
12.317.3.6 GetDebugStream() . . . 1266
12.317.3.7 GetErrorFlag() . . . 1266
12.317.3.8 GetErrorStream() . . . 1266
12.317.3.9 GetStream() . . . 1266
12.317.3.10 GetWarningFlag() . . . 1266
12.317.3.11 GetWarningStream() . . . 1266
12.317.3.12 SetDebug() . . . 1266
12.317.3.13 SetDebugStream() . . . 1267
12.317.3.14 SetError() . . . 1267
12.317.3.15 SetErrorStream() . . . 1267
12.317.3.16 SetStream() . . . 1267
12.317.3.17 SetStreamToFile() . . . 1267
12.317.3.18 SetWarning() . . . 1268
12.317.3.19 SetWarningStream() . . . 1268
12.317.3.20 WarningOff() . . . 1268
12.317.3.21 WarningOn() . . . 1268
12.318 gdcm::TransferSyntax Class Reference . . . 1268
12.318.1 Detailed Description . . . 1270
12.318.2 Member Enumeration Documentation . . . 1270

12.318.2.1 NegociatedType . . . 1270
12.318.2.2 TSType . . . 1271
12.318.3 Constructor & Destructor Documentation . . . 1272
12.318.3.1 TransferSyntax() . . . 1272
12.318.4 Member Function Documentation . . . 1272
12.318.4.1 CanStoreLossy() . . . 1272
12.318.4.2 GetNegociatedType() . . . 1272
12.318.4.3 GetString() . . . 1272
12.318.4.4 GetSwapCode() . . . 1272
12.318.4.5 GetTSString() . . . 1273
12.318.4.6 GetTSType() . . . 1273
12.318.4.7 IsEncapsulated() . . . 1273
12.318.4.8 IsEncoded() . . . 1273
12.318.4.9 IsExplicit() . . . 1273
12.318.4.10 IsImplicit() . . . 1273
12.318.4.11 IsLossless() . . . 1273
12.318.4.12 IsLossy() . . . 1274
12.318.4.13 IsValid() . . . 1274
12.318.4.14 operator TSType() . . . 1274
12.318.5 Friends And Related Symbol Documentation . . . 1274
12.318.5.1 operator<< . . . 1274
12.319 gdcm::network::TransferSyntaxSub Class Reference . . . 1274
12.319.1 Detailed Description . . . 1275
12.319.2 Constructor & Destructor Documentation . . . 1275
12.319.2.1 TransferSyntaxSub() . . . 1275
12.319.3 Member Function Documentation . . . 1275
12.319.3.1 GetName() . . . 1275
12.319.3.2 operator==() . . . 1275
12.319.3.3 Print() . . . 1275
12.319.3.4 Read() . . . 1275
12.319.3.5 SetName() . . . 1276
12.319.3.6 SetNameFromUID() . . . 1276
12.319.3.7 Size() . . . 1276
12.319.3.8 Write() . . . 1276
12.320 gdcm::network::Transition Struct Reference . . . 1276
12.320.1 Constructor & Destructor Documentation . . . 1277
12.320.1.1 Transition() [1/2] . . . 1277
12.320.1.2 ~Transition() . . . 1277
12.320.1.3 Transition() [2/2] . . . 1277

12.320.2 Member Function Documentation . . . 1278
12.320.2.1 MakeNew() . . . 1278
12.320.3 Member Data Documentation . . . 1278
12.320.3.1 mAction . . . 1278
12.320.3.2 mEnd . . . 1278
12.321 gdcm::Type Class Reference . . . 1278
12.321.1 Detailed Description . . . 1279
12.321.2 Member Enumeration Documentation . . . 1279
12.321.2.1 TypeType . . . 1279
12.321.3 Constructor & Destructor Documentation . . . 1280
12.321.3.1 Type() . . . 1280
12.321.4 Member Function Documentation . . . 1280
12.321.4.1 GetTypeString() . . . 1280
12.321.4.2 GetTypeType() . . . 1280
12.321.4.3 operator TypeType() . . . 1280
12.321.5 Friends And Related Symbol Documentation . . . 1280
12.321.5.1 operator<< . . . 1280
12.322 gdcm::UI Struct Reference . . . 1281
12.322.1 Friends And Related Symbol Documentation . . . 1281
12.322.1.1 operator<< . . . 1281
12.322.2 Member Data Documentation . . . 1281
12.322.2.1 Internal . . . 1281
12.323 gdcm::UIDGenerator Class Reference . . . 1281
12.323.1 Detailed Description . . . 1282
12.323.2 Constructor & Destructor Documentation . . . 1282
12.323.2.1 UIDGenerator() . . . 1282
12.323.3 Member Function Documentation . . . 1283
12.323.3.1 Generate() . . . 1283
12.323.3.2 GenerateUUID() . . . 1283
12.323.3.3 GetGDCMUID() . . . 1283
12.323.3.4 GetRoot() . . . 1283
12.323.3.5 IsValid() . . . 1283
12.323.3.6 SetRoot() . . . 1284
12.324 gdcm::UIDs Class Reference . . . 1284
12.324.1 Detailed Description . . . 1300
12.324.2 Member Typedef Documentation . . . 1300
12.324.2.1 TransferSyntaxStringsType . . . 1300
12.324.3 Member Enumeration Documentation . . . 1300
12.324.3.1 TSName . . . 1300

12.324.3.2 TSType . . . 1300
12.324.4 Member Function Documentation . . . 1300
12.324.4.1 GetName() . . . 1300
12.324.4.2 GetNumberOfTransferSyntaxStrings() . . . 1301
12.324.4.3 GetString() . . . 1301
12.324.4.4 GetTransferSyntaxString() . . . 1301
12.324.4.5 GetTransferSyntaxStrings() . . . 1301
12.324.4.6 GetUIDName() . . . 1301
12.324.4.7 GetUIDString() . . . 1301
12.324.4.8 operator TSType() . . . 1301
12.324.4.9 SetFromUID() . . . 1302
12.325 gdcm::network::ULAction Class Reference . . . 1302
12.325.1 Detailed Description . . . 1304
12.325.2 Constructor & Destructor Documentation . . . 1304
12.325.2.1 ULAction() [1/2] . . . 1304
12.325.2.2 ∼ULAction() . . . 1304
12.325.2.3 ULAction() [2/2] . . . 1305
12.325.3 Member Function Documentation . . . 1305
12.325.3.1 operator=() . . . 1305
12.325.3.2 PerformAction() . . . 1305
12.326 gdcm::network::ULActionAA1 Class Reference . . . 1306
12.326.1 Member Function Documentation . . . 1307
12.326.1.1 PerformAction() . . . 1307
12.327 gdcm::network::ULActionAA2 Class Reference . . . 1307
12.327.1 Member Function Documentation . . . 1308
12.327.1.1 PerformAction() . . . 1308
12.328 gdcm::network::ULActionAA3 Class Reference . . . 1308
12.328.1 Member Function Documentation . . . 1309
12.328.1.1 PerformAction() . . . 1309
12.329 gdcm::network::ULActionAA4 Class Reference . . . 1310
12.329.1 Member Function Documentation . . . 1311
12.329.1.1 PerformAction() . . . 1311
12.330 gdcm::network::ULActionAA5 Class Reference . . . 1311
12.330.1 Member Function Documentation . . . 1312
12.330.1.1 PerformAction() . . . 1312
12.331 gdcm::network::ULActionAA6 Class Reference . . . 1312
12.331.1 Member Function Documentation . . . 1313
12.331.1.1 PerformAction() . . . 1313
12.332 gdcm::network::ULActionAA7 Class Reference . . . 1314

12.332.1 Member Function Documentation . . . 1315
12.332.1.1 PerformAction() . . . 1315
12.333 gdcm::network::ULActionAA8 Class Reference . . . 1315
12.333.1 Member Function Documentation . . . 1316
12.333.1.1 PerformAction() . . . 1316
12.334 gdcm::network::ULActionAE1 Class Reference . . . 1316
12.334.1 Member Function Documentation . . . 1317
12.334.1.1 PerformAction() . . . 1317
12.335 gdcm::network::ULActionAE2 Class Reference . . . 1318
12.335.1 Member Function Documentation . . . 1319
12.335.1.1 PerformAction() . . . 1319
12.336 gdcm::network::ULActionAE3 Class Reference . . . 1319
12.336.1 Member Function Documentation . . . 1320
12.336.1.1 PerformAction() . . . 1320
12.337 gdcm::network::ULActionAE4 Class Reference . . . 1320
12.337.1 Member Function Documentation . . . 1321
12.337.1.1 PerformAction() . . . 1321
12.338 gdcm::network::ULActionAE5 Class Reference . . . 1322
12.338.1 Member Function Documentation . . . 1323
12.338.1.1 PerformAction() . . . 1323
12.339 gdcm::network::ULActionAE6 Class Reference . . . 1323
12.339.1 Member Function Documentation . . . 1324
12.339.1.1 PerformAction() . . . 1324
12.340 gdcm::network::ULActionAE7 Class Reference . . . 1324
12.340.1 Member Function Documentation . . . 1325
12.340.1.1 PerformAction() . . . 1325
12.341 gdcm::network::ULActionAE8 Class Reference . . . 1326
12.341.1 Member Function Documentation . . . 1327
12.341.1.1 PerformAction() . . . 1327
12.342 gdcm::network::ULActionAR1 Class Reference . . . 1327
12.342.1 Member Function Documentation . . . 1328
12.342.1.1 PerformAction() . . . 1328
12.343 gdcm::network::ULActionAR10 Class Reference . . . 1328
12.343.1 Member Function Documentation . . . 1329
12.343.1.1 PerformAction() . . . 1329
12.344 gdcm::network::ULActionAR2 Class Reference . . . 1330
12.344.1 Member Function Documentation . . . 1331
12.344.1.1 PerformAction() . . . 1331
12.345 gdcm::network::ULActionAR3 Class Reference . . . 1331

12.345.1 Member Function Documentation . . . 1332
12.345.1.1 PerformAction() . . . 1332
12.346 gdcm::network::ULActionAR4 Class Reference . . . 1332
12.346.1 Member Function Documentation . . . 1333
12.346.1.1 PerformAction() . . . 1333
12.347 gdcm::network::ULActionAR5 Class Reference . . . 1334
12.347.1 Member Function Documentation . . . 1335
12.347.1.1 PerformAction() . . . 1335
12.348 gdcm::network::ULActionAR6 Class Reference . . . 1335
12.348.1 Member Function Documentation . . . 1336
12.348.1.1 PerformAction() . . . 1336
12.349 gdcm::network::ULActionAR7 Class Reference . . . 1336
12.349.1 Member Function Documentation . . . 1337
12.349.1.1 PerformAction() . . . 1337
12.350 gdcm::network::ULActionAR8 Class Reference . . . 1338
12.350.1 Member Function Documentation . . . 1339
12.350.1.1 PerformAction() . . . 1339
12.351 gdcm::network::ULActionAR9 Class Reference . . . 1339
12.351.1 Member Function Documentation . . . 1340
12.351.1.1 PerformAction() . . . 1340
12.352 gdcm::network::ULActionDT1 Class Reference . . . 1340
12.352.1 Member Function Documentation . . . 1341
12.352.1.1 PerformAction() . . . 1341
12.353 gdcm::network::ULActionDT2 Class Reference . . . 1342
12.353.1 Member Function Documentation . . . 1343
12.353.1.1 PerformAction() . . . 1343
12.354 gdcm::network::ULBasicCallback Class Reference . . . 1343
12.354.1 Detailed Description . . . 1345
12.354.2 Constructor & Destructor Documentation . . . 1345
12.354.2.1 ULBasicCallback() . . . 1345
12.354.2.2 ~ULBasicCallback() . . . 1345
12.354.3 Member Function Documentation . . . 1345
12.354.3.1 GetDataSets() . . . 1345
12.354.3.2 GetResponses() . . . 1345
12.354.3.3 HandleDataSet() . . . 1345
12.354.3.4 HandleResponse() . . . 1345
12.355 gdcm::network::ULConnection Class Reference . . . 1346
12.355.1 Detailed Description . . . 1347
12.355.2 Constructor & Destructor Documentation . . . 1347

12.355.2.1 ULConnection() [1/2] . . . 1347
12.355.2.2 ~ULConnection() . . . 1347
12.355.2.3 ULConnection() [2/2] . . . 1347
12.355.3 Member Function Documentation . . . 1347
12.355.3.1 AddAcceptedPresentationContext() . . . 1347
12.355.3.2 FindContext() . . . 1348
12.355.3.3 GetAcceptedPresentationContexts() [1/2] . . . 1348
12.355.3.4 GetAcceptedPresentationContexts() [2/2] . . . 1348
12.355.3.5 GetConnectionInfo() . . . 1348
12.355.3.6 GetMaxPDUSize() . . . 1348
12.355.3.7 GetPresentationContextACByID() . . . 1348
12.355.3.8 GetPresentationContextIDFromPresentationContext() . . . 1348
12.355.3.9 GetPresentationContextRQByID() . . . 1348
12.355.3.10 GetPresentationContexts() . . . 1349
12.355.3.11 GetProtocol() . . . 1349
12.355.3.12 GetState() . . . 1349
12.355.3.13 GetTimer() . . . 1349
12.355.3.14 InitializeConnection() . . . 1349
12.355.3.15 InitializeIncomingConnection() . . . 1349
12.355.3.16 operator=() . . . 1349
12.355.3.17 SetMaxPDUSize() . . . 1349
12.355.3.18 SetPresentationContexts() [1/2] . . . 1350
12.355.3.19 SetPresentationContexts() [2/2] . . . 1350
12.355.3.20 SetState() . . . 1350
12.355.3.21 StopProtocol() . . . 1350
12.355.4 Friends And Related Symbol Documentation . . . 1350
12.355.4.1 ULActionAE6 . . . 1350
12.355.4.2 ULConnectionManager . . . 1350
12.356 gdcm::network::ULConnectionCallback Class Reference . . . 1351
12.356.1 Detailed Description . . . 1351
12.356.2 Constructor & Destructor Documentation . . . 1352
12.356.2.1 ULConnectionCallback() . . . 1352
12.356.2.2 ~ULConnectionCallback() . . . 1352
12.356.3 Member Function Documentation . . . 1352
12.356.3.1 DataSetHandled() . . . 1352
12.356.3.2 DataSetHandles() . . . 1352
12.356.3.3 HandleDataSet() . . . 1352
12.356.3.4 HandleResponse() . . . 1352
12.356.3.5 ResetHandledDataSet() . . . 1352

12.356.3.6 SetImplicitFlag() . . . 1353
12.356.4 Member Data Documentation . . . 1353
12.356.4.1 mImplicit . . . 1353
12.357 gdcm::network::ULConnectionInfo Class Reference . . . 1353
12.357.1 Detailed Description . . . 1353
12.357.2 Constructor & Destructor Documentation . . . 1354
12.357.2.1 ULConnectionInfo() . . . 1354
12.357.3 Member Function Documentation . . . 1354
12.357.3.1 GetCalledAETitle() . . . 1354
12.357.3.2 GetCalledComputerName() . . . 1354
12.357.3.3 GetCalledIPAddress() . . . 1354
12.357.3.4 GetCalledIPPort() . . . 1354
12.357.3.5 GetCallingAETitle() . . . 1354
12.357.3.6 GetMaxPDULength() . . . 1354
12.357.3.7 Initialize() . . . 1354
12.357.3.8 SetMaxPDULength() . . . 1355
12.358 gdcm::network::ULConnectionManager Class Reference . . . 1355
12.358.1 Detailed Description . . . 1358
12.358.2 Constructor & Destructor Documentation . . . 1358
12.358.2.1 ULConnectionManager() [1/2] . . . 1358
12.358.2.2 ULConnectionManager() [2/2] . . . 1358
12.358.2.3 ∼ULConnectionManager() . . . 1358
12.358.3 Member Function Documentation . . . 1358
12.358.3.1 BreakConnection() . . . 1358
12.358.3.2 BreakConnectionNow() . . . 1358
12.358.3.3 EstablishConnection() . . . 1359
12.358.3.4 EstablishConnectionMove() . . . 1359
12.358.3.5 RunEventLoop() . . . 1359
12.358.3.6 RunMoveEventLoop() . . . 1359
12.358.3.7 SendEcho() . . . 1359
12.358.3.8 SendFind() [1/2] . . . 1360
12.358.3.9 SendFind() [2/2] . . . 1360
12.358.3.10 SendMove() [1/2] . . . 1360
12.358.3.11 SendMove() [2/2] . . . 1360
12.358.3.12 SendNAction() [1/2] . . . 1360
12.358.3.13 SendNAction() [2/2] . . . 1360
12.358.3.14 SendNCreate() [1/2] . . . 1360
12.358.3.15 SendNCreate() [2/2] . . . 1361
12.358.3.16 SendNDelete() [1/2] . . . 1361

12.358.3.17 SendNDelete() [2/2] . . . 1361
12.358.3.18 SendNEventReport() [1/2] . . . 1361
12.358.3.19 SendNEventReport() [2/2] . . . 1361
12.358.3.20 SendNGet() [1/2] . . . 1361
12.358.3.21 SendNGet() [2/2] . . . 1361
12.358.3.22 SendNSet() [1/2] . . . 1361
12.358.3.23 SendNSet() [2/2] . . . 1362
12.358.3.24 SendStore() [1/2] . . . 1362
12.358.3.25 SendStore() [2/2] . . . 1362
12.358.4 Member Data Documentation . . . 1362
12.358.4.1 mConnection . . . 1362
12.358.4.2 mSecondaryConnection . . . 1362
12.358.4.3 mTransitions . . . 1362
12.359 gdcm::network::ULEvent Class Reference . . . 1363
12.359.1 Detailed Description . . . 1363
12.359.2 Constructor & Destructor Documentation . . . 1363
12.359.2.1 ULEvent() [1/2] . . . 1363
12.359.2.2 ULEvent() [2/2] . . . 1363
12.359.2.3 ∼ULEvent() . . . 1364
12.359.3 Member Function Documentation . . . 1364
12.359.3.1 GetDataSetPos() . . . 1364
12.359.3.2 GetEvent() . . . 1364
12.359.3.3 GetIStream() . . . 1364
12.359.3.4 GetPDUs() . . . 1364
12.359.3.5 SetEvent() . . . 1364
12.359.3.6 SetPDU() . . . 1364
12.360 gdcm::network::ULTransitionTable Class Reference . . . 1364
12.360.1 Detailed Description . . . 1365
12.360.2 Constructor & Destructor Documentation . . . 1365
12.360.2.1 ULTransitionTable() . . . 1365
12.360.3 Member Function Documentation . . . 1365
12.360.3.1 HandleEvent() . . . 1365
12.360.3.2 PrintTable() . . . 1365
12.361 gdcm::network::ULWritingCallback Class Reference . . . 1366
12.361.1 Constructor & Destructor Documentation . . . 1367
12.361.1.1 ULWritingCallback() . . . 1367
12.361.1.2 ∼ULWritingCallback() . . . 1367
12.361.2 Member Function Documentation . . . 1367
12.361.2.1 HandleDataSet() . . . 1367

12.361.2.2 HandleResponse() . . . 1367
12.361.2.3 SetDirectory() . . . 1368
12.362 gdcm::UNExplicitDataElement Class Reference . . . 1368
12.362.1 Detailed Description . . . 1371
12.362.2 Member Function Documentation . . . 1371
12.362.2.1 GetLength() . . . 1371
12.362.2.2 Read() . . . 1371
12.362.2.3 ReadPreValue() . . . 1371
12.362.2.4 ReadValue() . . . 1371
12.362.2.5 ReadWithLength() . . . 1372
12.363 gdcm::UNExplicitImplicitDataElement Class Reference . . . 1372
12.363.1 Detailed Description . . . 1375
12.363.2 Member Function Documentation . . . 1375
12.363.2.1 GetLength() . . . 1375
12.363.2.2 Read() . . . 1375
12.363.2.3 ReadPreValue() . . . 1375
12.363.2.4 ReadValue() . . . 1375
12.364 gdcm::Unpacker12Bits Class Reference . . . 1376
12.364.1 Detailed Description . . . 1376
12.364.2 Member Function Documentation . . . 1376
12.364.2.1 Pack() . . . 1376
12.364.2.2 Unpack() . . . 1377
12.365 gdcm::Usage Class Reference . . . 1377
12.365.1 Detailed Description . . . 1378
12.365.2 Member Enumeration Documentation . . . 1378
12.365.2.1 UsageType . . . 1378
12.365.3 Constructor & Destructor Documentation . . . 1378
12.365.3.1 Usage() . . . 1378
12.365.4 Member Function Documentation . . . 1379
12.365.4.1 GetUsageString() . . . 1379
12.365.4.2 GetUsageType() . . . 1379
12.365.4.3 operator UsageType() . . . 1379
12.365.5 Friends And Related Symbol Documentation . . . 1379
12.365.5.1 operator<< . . . 1379
12.366 gdcm::UserEvent Class Reference . . . 1380
12.367 gdcm::network::UserInformation Class Reference . . . 1381
12.367.1 Detailed Description . . . 1381
12.367.2 Constructor & Destructor Documentation . . . 1382
12.367.2.1 UserInformation() [1/2] . . . 1382

12.367.2.2 ~UserInformation() . . . 1382
12.367.2.3 UserInformation() [2/2] . . . 1382
12.367.3 Member Function Documentation . . . 1382
12.367.3.1 AddRoleSelectionSub() . . . 1382
12.367.3.2 AddSOPClassExtendedNegociationSub() . . . 1382
12.367.3.3 GetMaximumLengthSub() [1/2] . . . 1382
12.367.3.4 GetMaximumLengthSub() [2/2] . . . 1382
12.367.3.5 operator=() . . . 1383
12.367.3.6 Print() . . . 1383
12.367.3.7 Read() . . . 1383
12.367.3.8 Size() . . . 1383
12.367.3.9 Write() . . . 1383
12.368 gdcm::UUIDGenerator Class Reference . . . 1383
12.368.1 Detailed Description . . . 1384
12.368.2 Member Function Documentation . . . 1384
12.368.2.1 Generate() . . . 1384
12.368.2.2 IsValid() . . . 1384
12.369 gdcm::Validate Class Reference . . . 1384
12.369.1 Detailed Description . . . 1385
12.369.2 Constructor & Destructor Documentation . . . 1385
12.369.2.1 Validate() . . . 1385
12.369.2.2 ~Validate() . . . 1386
12.369.3 Member Function Documentation . . . 1386
12.369.3.1 GetValidatedFile() . . . 1386
12.369.3.2 SetFile() . . . 1386
12.369.3.3 Validation() . . . 1386
12.369.4 Member Data Documentation . . . 1386
12.369.4.1 F . . . 1386
12.369.4.2 V . . . 1386
12.370 gdcm::Value Class Reference . . . 1387
12.370.1 Detailed Description . . . 1388
12.370.2 Constructor & Destructor Documentation . . . 1388
12.370.2.1 Value() . . . 1388
12.370.2.2 ~Value() . . . 1388
12.370.3 Member Function Documentation . . . 1389
12.370.3.1 Clear() . . . 1389
12.370.3.2 GetLength() . . . 1389
12.370.3.3 operator==() . . . 1389
12.370.3.4 SetLength() . . . 1389

12.370.3.5 SetLengthOnly() . . . 1389
12.370.4 Friends And Related Symbol Documentation . . . 1390
12.370.4.1 DataElement . . . 1390
12.371 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference . . . 1390
12.371.1 Detailed Description . . . 1390
12.371.2 Member Function Documentation . . . 1390
12.371.2.1 Read() . . . 1390
12.371.2.2 Write() . . . 1391
12.372 gdcm::MrProtocol::Vector3 Struct Reference . . . 1391
12.372.1 Member Data Documentation . . . 1391
12.372.1.1 dCor . . . 1391
12.372.1.2 dSag . . . 1391
12.372.1.3 dTra . . . 1391
12.373 gdcm::Version Class Reference . . . 1392
12.373.1 Detailed Description . . . 1392
12.373.2 Constructor & Destructor Documentation . . . 1392
12.373.2.1 Version() . . . 1392
12.373.2.2 ~Version() . . . 1392
12.373.3 Member Function Documentation . . . 1393
12.373.3.1 GetBuildVersion() . . . 1393
12.373.3.2 GetMajorVersion() . . . 1393
12.373.3.3 GetMinorVersion() . . . 1393
12.373.3.4 GetVersion() . . . 1393
12.373.3.5 Print() . . . 1393
12.373.4 Friends And Related Symbol Documentation . . . 1393
12.373.4.1 operator<< . . . 1393
12.374 gdcm::VL Class Reference . . . 1394
12.374.1 Detailed Description . . . 1395
12.374.2 Member Typedef Documentation . . . 1395
12.374.2.1 Type . . . 1395
12.374.3 Constructor & Destructor Documentation . . . 1395
12.374.3.1 VL() . . . 1395
12.374.4 Member Function Documentation . . . 1395
12.374.4.1 GetLength() . . . 1395
12.374.4.2 GetVL16Max() . . . 1396
12.374.4.3 GetVL32Max() . . . 1396
12.374.4.4 IsOdd() . . . 1396
12.374.4.5 IsUndefined() . . . 1396
12.374.4.6 operator uint32_t() . . . 1396

12.374.4.7 operator++() [1/2] . . . 1396
12.374.4.8 operator++() [2/2] . . . 1396
12.374.4.9 operator+=() . . . 1397
12.374.4.10 Read() . . . 1397
12.374.4.11 Read16() . . . 1397
12.374.4.12 SetToUndefined() . . . 1397
12.374.4.13 Write() . . . 1397
12.374.4.14 Write16() . . . 1397
12.374.5 Friends And Related Symbol Documentation . . . 1398
12.374.5.1 operator<< . . . 1398
12.375 gdcm::VM Class Reference . . . 1398
12.375.1 Detailed Description . . . 1399
12.375.2 Member Enumeration Documentation . . . 1400
12.375.2.1 VMType . . . 1400
12.375.3 Constructor & Destructor Documentation . . . 1401
12.375.3.1 VM() . . . 1401
12.375.4 Member Function Documentation . . . 1401
12.375.4.1 Compatible() . . . 1401
12.375.4.2 GetIndex() . . . 1401
12.375.4.3 GetLength() . . . 1402
12.375.4.4 GetNumberOfElementsFromArray() . . . 1402
12.375.4.5 GetVMString() . . . 1402
12.375.4.6 GetVMType() . . . 1402
12.375.4.7 GetVMTypeFromLength() . . . 1402
12.375.4.8 IsValid() . . . 1402
12.375.4.9 operator VMType() . . . 1402
12.375.5 Friends And Related Symbol Documentation . . . 1403
12.375.5.1 operator<< . . . 1403
12.376 gdcm::VMToLength< T > Struct Template Reference . . . 1403
12.377 gdcm::VR Class Reference . . . 1403
12.377.1 Detailed Description . . . 1405
12.377.2 Member Enumeration Documentation . . . 1405
12.377.2.1 VRType . . . 1405
12.377.3 Constructor & Destructor Documentation . . . 1407
12.377.3.1 VR() . . . 1407
12.377.4 Member Function Documentation . . . 1407
12.377.4.1 CanDisplay() . . . 1407
12.377.4.2 Compatible() . . . 1407
12.377.4.3 GetLength() [1/2] . . . 1407

12.377.4.4 GetLength() [2/2] . . . 1408
12.377.4.5 GetSize() . . . 1408
12.377.4.6 GetSizeof() . . . 1408
12.377.4.7 GetVRString() . . . 1408
12.377.4.8 GetVRStringFromFile() . . . 1408
12.377.4.9 GetVRType() . . . 1408
12.377.4.10 GetVRTypeFromFile() . . . 1408
12.377.4.11 IsASCII() . . . 1409
12.377.4.12 IsASCII2() . . . 1409
12.377.4.13 IsBinary() . . . 1409
12.377.4.14 IsBinary2() . . . 1409
12.377.4.15 IsDual() . . . 1409
12.377.4.16 IsSwap() . . . 1409
12.377.4.17 IsValid() [1/2] . . . 1409
12.377.4.18 IsValid() [2/2] . . . 1409
12.377.4.19 IsVRFile() . . . 1410
12.377.4.20 operator VRType() . . . 1410
12.377.4.21 Read() . . . 1410
12.377.4.22 Write() . . . 1410
12.377.5 Friends And Related Symbol Documentation . . . 1410
12.377.5.1 operator<< . . . 1410
12.378 gdcm::VR16ExplicitDataElement Class Reference . . . 1411
12.378.1 Detailed Description . . . 1413
12.378.2 Member Function Documentation . . . 1413
12.378.2.1 GetLength() . . . 1413
12.378.2.2 Read() . . . 1414
12.378.2.3 ReadPreValue() . . . 1414
12.378.2.4 ReadValue() . . . 1414
12.378.2.5 ReadWithLength() . . . 1414
12.379 gdcm::VRToEncoding< T > Struct Template Reference . . . 1414
12.380 gdcm::VRToType< T > Struct Template Reference . . . 1415
12.380.1 Detailed Description . . . 1415
12.381 gdcm::VRVLSize< T > Class Template Reference . . . 1415
12.382 gdcm::VRVLSize< 0 > Class Reference . . . 1416
12.382.1 Member Function Documentation . . . 1416
12.382.1.1 Read() . . . 1416
12.382.1.2 Write() . . . 1417
12.383 gdcm::VRVLSize< 1 > Class Reference . . . 1417
12.383.1 Member Function Documentation . . . 1418

12.383.1.1 Read() . . . 1418
12.383.1.2 Write() . . . 1418
12.384 vtkGDCMImageReader Class Reference . . . 1418
12.384.1 Detailed Description . . . 1421
12.384.2 Constructor & Destructor Documentation . . . 1421
12.384.2.1 vtkGDCMImageReader() . . . 1421
12.384.2.2 ~vtkGDCMImageReader() . . . 1421
12.384.3 Member Function Documentation . . . 1421
12.384.3.1 CanReadFile() . . . 1421
12.384.3.2 ExecuteData() . . . 1422
12.384.3.3 ExecuteInformation() . . . 1422
12.384.3.4 FillMedicalImageInformation() . . . 1422
12.384.3.5 GetDescriptiveName() . . . 1422
12.384.3.6 GetFileExtensions() . . . 1422
12.384.3.7 GetIconImage() . . . 1422
12.384.3.8 GetOverlay() . . . 1422
12.384.3.9 LoadSingleFile() . . . 1422
12.384.3.10 New() . . . 1423
12.384.3.11 PrintSelf() . . . 1423
12.384.3.12 RequestDataCompat() . . . 1423
12.384.3.13 RequestInformationCompat() . . . 1423
12.384.3.14 SetCurve() . . . 1423
12.384.3.15 SetFileNames() . . . 1424
12.384.3.16 SetFilePattern() . . . 1424
12.384.3.17 SetFilePrefix() . . . 1424
12.384.3.18 SetMedicalImageProperties() . . . 1424
12.384.3.19 vtkBooleanMacro() [1/5] . . . 1424
12.384.3.20 vtkBooleanMacro() [2/5] . . . 1424
12.384.3.21 vtkBooleanMacro() [3/5] . . . 1425
12.384.3.22 vtkBooleanMacro() [4/5] . . . 1425
12.384.3.23 vtkBooleanMacro() [5/5] . . . 1425
12.384.3.24 vtkGetMacro() [1/11] . . . 1425
12.384.3.25 vtkGetMacro() [2/11] . . . 1425
12.384.3.26 vtkGetMacro() [3/11] . . . 1426
12.384.3.27 vtkGetMacro() [4/11] . . . 1426
12.384.3.28 vtkGetMacro() [5/11] . . . 1426
12.384.3.29 vtkGetMacro() [6/11] . . . 1426
12.384.3.30 vtkGetMacro() [7/11] . . . 1426
12.384.3.31 vtkGetMacro() [8/11] . . . 1427

12.384.3.32 vtkGetMacro() [9/11] . . . 1427
12.384.3.33 vtkGetMacro() [10/11] . . . 1427
12.384.3.34 vtkGetMacro() [11/11] . . . 1427
12.384.3.35 vtkGetObjectMacro() [1/4] . . . 1427
12.384.3.36 vtkGetObjectMacro() [2/4] . . . 1427
12.384.3.37 vtkGetObjectMacro() [3/4] . . . 1428
12.384.3.38 vtkGetObjectMacro() [4/4] . . . 1428
12.384.3.39 vtkGetStringMacro() [1/2] . . . 1428
12.384.3.40 vtkGetStringMacro() [2/2] . . . 1428
12.384.3.41 vtkGetVector3Macro() . . . 1428
12.384.3.42 vtkGetVector6Macro() . . . 1428
12.384.3.43 vtkSetMacro() [1/4] . . . 1429
12.384.3.44 vtkSetMacro() [2/4] . . . 1429
12.384.3.45 vtkSetMacro() [3/4] . . . 1429
12.384.3.46 vtkSetMacro() [4/4] . . . 1429
12.384.3.47 vtkSetVector6Macro() . . . 1429
12.384.3.48 vtkTypeMacro() . . . 1430
12.384.4 Member Data Documentation . . . 1430
12.384.4.1 ApplyInverseVideo . . . 1430
12.384.4.2 ApplyLookupTable . . . 1430
12.384.4.3 ApplyPlanarConfiguration . . . 1430
12.384.4.4 ApplyShiftScale . . . 1430
12.384.4.5 ApplyYBRToRGB . . . 1430
12.384.4.6 Curve . . . 1430
12.384.4.7 DirectionCosines . . . 1431
12.384.4.8 FileNames . . . 1431
12.384.4.9 ForceRescale . . . 1431
12.384.4.10 IconDataScalarType . . . 1431
12.384.4.11 IconImageDataExtent . . . 1431
12.384.4.12 IconNumberOfScalarComponents . . . 1431
12.384.4.13 ImageFormat . . . 1431
12.384.4.14 ImageOrientationPatient . . . 1431
12.384.4.15 ImagePositionPatient . . . 1432
12.384.4.16 LoadIconImage . . . 1432
12.384.4.17 LoadOverlays . . . 1432
12.384.4.18 LossyFlag . . . 1432
12.384.4.19 MedicalImageProperties . . . 1432
12.384.4.20 NumberOfIconImages . . . 1432
12.384.4.21 NumberOfOverlays . . . 1432

12.384.4.22 PlanarConfiguration . . . 1433
12.384.4.23 Scale . . . 1433
12.384.4.24 Shift . . . 1433
12.385 vtkGDCMImageReader2 Class Reference . . . 1433
12.385.1 Detailed Description . . . 1436
12.385.2 Constructor & Destructor Documentation . . . 1436
12.385.2.1 vtkGDCMImageReader2() . . . 1436
12.385.2.2 ∼vtkGDCMImageReader2() . . . 1436
12.385.3 Member Function Documentation . . . 1436
12.385.3.1 CanReadFile() . . . 1436
12.385.3.2 FillMedicalImageInformation() . . . 1436
12.385.3.3 GetDescriptiveName() . . . 1436
12.385.3.4 GetFileExtensions() . . . 1437
12.385.3.5 GetIconImage() . . . 1437
12.385.3.6 GetIconImagePort() . . . 1437
12.385.3.7 GetOverlay() . . . 1437
12.385.3.8 GetOverlayPort() . . . 1437
12.385.3.9 LoadSingleFile() . . . 1437
12.385.3.10 New() . . . 1437
12.385.3.11 PrintSelf() . . . 1438
12.385.3.12 ProcessRequest() . . . 1438
12.385.3.13 RequestData() . . . 1438
12.385.3.14 RequestDataCompat() . . . 1438
12.385.3.15 RequestInformation() . . . 1438
12.385.3.16 RequestInformationCompat() . . . 1439
12.385.3.17 SetCurve() . . . 1439
12.385.3.18 SetFilePattern() . . . 1439
12.385.3.19 SetFilePrefix() . . . 1439
12.385.3.20 SetMedicalImageProperties() . . . 1439
12.385.3.21 vtkBooleanMacro() [1/5] . . . 1439
12.385.3.22 vtkBooleanMacro() [2/5] . . . 1440
12.385.3.23 vtkBooleanMacro() [3/5] . . . 1440
12.385.3.24 vtkBooleanMacro() [4/5] . . . 1440
12.385.3.25 vtkBooleanMacro() [5/5] . . . 1440
12.385.3.26 vtkGetMacro() [1/11] . . . 1440
12.385.3.27 vtkGetMacro() [2/11] . . . 1441
12.385.3.28 vtkGetMacro() [3/11] . . . 1441
12.385.3.29 vtkGetMacro() [4/11] . . . 1441
12.385.3.30 vtkGetMacro() [5/11] . . . 1441

12.385.3.31 vtkGetMacro() [6/11] 1441
12.385.3.32 vtkGetMacro() [7/11] 1442
12.385.3.33 vtkGetMacro() [8/11] 1442
12.385.3.34 vtkGetMacro() [9/11] 1442
12.385.3.35 vtkGetMacro() [10/11] 1442
12.385.3.36 vtkGetMacro() [11/11] 1442
12.385.3.37 vtkGetObjectMacro() [1/2] 1442
12.385.3.38 vtkGetObjectMacro() [2/2] 1443
12.385.3.39 vtkGetStringMacro() [1/2] 1443
12.385.3.40 vtkGetStringMacro() [2/2] 1443
12.385.3.41 vtkGetVector3Macro() 1443
12.385.3.42 vtkGetVector6Macro() 1443
12.385.3.43 vtkSetMacro() [1/4] 1444
12.385.3.44 vtkSetMacro() [2/4] 1444
12.385.3.45 vtkSetMacro() [3/4] 1444
12.385.3.46 vtkSetMacro() [4/4] 1444
12.385.3.47 vtkSetVector6Macro() 1444
12.385.3.48 vtkTypeMacro() 1445
12.385.4 Member Data Documentation 1445
12.385.4.1 ApplyInverseVideo 1445
12.385.4.2 ApplyLookupTable 1445
12.385.4.3 ApplyPlanarConfiguration 1445
12.385.4.4 ApplyShiftScale 1445
12.385.4.5 ApplyYBRToRGB 1445
12.385.4.6 Curve 1445
12.385.4.7 DirectionCosines 1446
12.385.4.8 ForceRescale 1446
12.385.4.9 IconDataScalarType 1446
12.385.4.10 IconImageDataExtent 1446
12.385.4.11 IconNumberOfScalarComponents 1446
12.385.4.12 ImageFormat 1446
12.385.4.13 ImageOrientationPatient 1446
12.385.4.14 ImagePositionPatient 1446
12.385.4.15 LoadIconImage 1447
12.385.4.16 LoadOverlays 1447
12.385.4.17 LossyFlag 1447
12.385.4.18 NumberOfIconImages 1447
12.385.4.19 NumberOfOverlays 1447
12.385.4.20 PlanarConfiguration 1447

12.385.4.21 Scale . . . 1447
12.385.4.22 Shift . . . 1448
12.386 vtkGDCMImageWriter Class Reference . . . 1448
12.386.1 Detailed Description . . . 1450
12.386.2 Member Enumeration Documentation . . . 1450
12.386.2.1 CompressionTypes . . . 1450
12.386.3 Constructor & Destructor Documentation . . . 1450
12.386.3.1 vtkGDCMImageWriter() . . . 1450
12.386.3.2 ~vtkGDCMImageWriter() . . . 1450
12.386.4 Member Function Documentation . . . 1451
12.386.4.1 GetDescriptiveName() . . . 1451
12.386.4.2 GetFileExtensions() . . . 1451
12.386.4.3 GetFileName() . . . 1451
12.386.4.4 New() . . . 1451
12.386.4.5 PrintSelf() . . . 1451
12.386.4.6 SetDirectionCosines() . . . 1451
12.386.4.7 SetDirectionCosinesFromImageOrientationPatient() . . . 1452
12.386.4.8 SetFileNames() . . . 1452
12.386.4.9 SetMedicalImageProperties() . . . 1452
12.386.4.10 vtkBooleanMacro() [1/2] . . . 1452
12.386.4.11 vtkBooleanMacro() [2/2] . . . 1452
12.386.4.12 vtkGetMacro() [1/7] . . . 1452
12.386.4.13 vtkGetMacro() [2/7] . . . 1453
12.386.4.14 vtkGetMacro() [3/7] . . . 1453
12.386.4.15 vtkGetMacro() [4/7] . . . 1453
12.386.4.16 vtkGetMacro() [5/7] . . . 1453
12.386.4.17 vtkGetMacro() [6/7] . . . 1453
12.386.4.18 vtkGetMacro() [7/7] . . . 1453
12.386.4.19 vtkGetObjectMacro() [1/3] . . . 1453
12.386.4.20 vtkGetObjectMacro() [2/3] . . . 1454
12.386.4.21 vtkGetObjectMacro() [3/3] . . . 1454
12.386.4.22 vtkGetStringMacro() [1/2] . . . 1454
12.386.4.23 vtkGetStringMacro() [2/2] . . . 1454
12.386.4.24 vtkSetMacro() [1/7] . . . 1454
12.386.4.25 vtkSetMacro() [2/7] . . . 1454
12.386.4.26 vtkSetMacro() [3/7] . . . 1454
12.386.4.27 vtkSetMacro() [4/7] . . . 1455
12.386.4.28 vtkSetMacro() [5/7] . . . 1455
12.386.4.29 vtkSetMacro() [6/7] . . . 1455

12.386.4.30 vtkSetMacro() [7/7] . . . 1455
12.386.4.31 vtkSetStringMacro() [1/2] . . . 1455
12.386.4.32 vtkSetStringMacro() [2/2] . . . 1455
12.386.4.33 vtkTypeMacro() . . . 1455
12.386.4.34 Write() . . . 1456
12.386.4.35 WriteGDCMData() . . . 1456
12.386.4.36 WriteSlice() . . . 1456
12.387 vtkGDCMMedicalImageProperties Class Reference . . . 1456
12.387.1 Constructor & Destructor Documentation . . . 1457
12.387.1.1 vtkGDCMMedicalImageProperties() . . . 1457
12.387.1.2 ∼vtkGDCMMedicalImageProperties() . . . 1458
12.387.2 Member Function Documentation . . . 1458
12.387.2.1 Clear() . . . 1458
12.387.2.2 GetFile() . . . 1458
12.387.2.3 New() . . . 1458
12.387.2.4 PrintSelf() . . . 1458
12.387.2.5 PushBackFile() . . . 1458
12.387.2.6 vtkTypeMacro() . . . 1458
12.387.3 Friends And Related Symbol Documentation . . . 1459
12.387.3.1 vtkGDCMImageReader . . . 1459
12.387.3.2 vtkGDCMImageReader2 . . . 1459
12.387.3.3 vtkGDCMImageWriter . . . 1459
12.388 vtkGDCMPolyDataReader Class Reference . . . 1459
12.388.1 Detailed Description . . . 1461
12.388.2 Constructor & Destructor Documentation . . . 1461
12.388.2.1 vtkGDCMPolyDataReader() . . . 1461
12.388.2.2 ∼vtkGDCMPolyDataReader() . . . 1461
12.388.3 Member Function Documentation . . . 1461
12.388.3.1 FillMedicalImageInformation() . . . 1461
12.388.3.2 New() . . . 1461
12.388.3.3 PrintSelf() . . . 1462
12.388.3.4 RequestData() . . . 1462
12.388.3.5 RequestData_HemodynamicWaveformStorage() . . . 1462
12.388.3.6 RequestData_RTStructureSetStorage() . . . 1462
12.388.3.7 RequestInformation() . . . 1462
12.388.3.8 RequestInformation_HemodynamicWaveformStorage() . . . 1462
12.388.3.9 RequestInformation_RTStructureSetStorage() . . . 1462
12.388.3.10 vtkGetObjectMacro() [1/2] . . . 1463
12.388.3.11 vtkGetObjectMacro() [2/2] . . . 1463

12.388.3.12 vtkGetStringMacro() . . . 1463
12.388.3.13 vtkSetStringMacro() . . . 1463
12.388.3.14 vtkTypeMacro() . . . 1463
12.388.4 Member Data Documentation . . . 1463
12.388.4.1 FileName . . . 1463
12.388.4.2 MedicalImageProperties . . . 1464
12.388.4.3 RTStructSetProperties . . . 1464
12.389 vtkGDCMPolyDataWriter Class Reference . . . 1464
12.389.1 Detailed Description . . . 1466
12.389.2 Constructor & Destructor Documentation . . . 1466
12.389.2.1 vtkGDCMPolyDataWriter() . . . 1466
12.389.2.2 ∼vtkGDCMPolyDataWriter() . . . 1466
12.389.3 Member Function Documentation . . . 1466
12.389.3.1 InitializeRTStructSet() . . . 1466
12.389.3.2 New() . . . 1466
12.389.3.3 PrintSelf() . . . 1467
12.389.3.4 SetMedicalImageProperties() . . . 1467
12.389.3.5 SetNumberOfInputPorts() . . . 1467
12.389.3.6 SetRTStructSetProperties() . . . 1467
12.389.3.7 vtkTypeMacro() . . . 1467
12.389.3.8 WriteData() . . . 1468
12.389.3.9 WriteRTSTRUCTData() . . . 1468
12.389.3.10 WriteRTSTRUCTInfo() . . . 1468
12.389.4 Member Data Documentation . . . 1468
12.389.4.1 MedicalImageProperties . . . 1468
12.389.4.2 RTStructSetProperties . . . 1468
12.390 vtkGDCMTesting Class Reference . . . 1469
12.390.1 Detailed Description . . . 1470
12.390.2 Member Typedef Documentation . . . 1470
12.390.2.1 MD5MetaImagesType . . . 1470
12.390.3 Constructor & Destructor Documentation . . . 1470
12.390.3.1 vtkGDCMTesting() . . . 1470
12.390.3.2 ∼vtkGDCMTesting() . . . 1470
12.390.4 Member Function Documentation . . . 1471
12.390.4.1 GetGDCMDataRoot() . . . 1471
12.390.4.2 GetMD5MetaImage() . . . 1471
12.390.4.3 GetMHDMD5FromFile() . . . 1471
12.390.4.4 GetNumberOfMD5MetaImages() . . . 1471
12.390.4.5 GetRAWMD5FromFile() . . . 1471

12.390.4.6 GetVTKDataRoot() . . . 1471
12.390.4.7 New() . . . 1472
12.390.4.8 PrintSelf() . . . 1472
12.390.4.9 vtkTypeMacro() . . . 1472
12.391 vtkGDCMThreadedImageReader Class Reference . . . 1472
12.391.1 Constructor & Destructor Documentation . . . 1475
12.391.1.1 vtkGDCMThreadedImageReader() . . . 1475
12.391.1.2 ~vtkGDCMThreadedImageReader() . . . 1475
12.391.2 Member Function Documentation . . . 1475
12.391.2.1 ExecuteData() . . . 1475
12.391.2.2 ExecuteInformation() . . . 1476
12.391.2.3 New() . . . 1476
12.391.2.4 PrintSelf() . . . 1476
12.391.2.5 ReadFiles() . . . 1476
12.391.2.6 RequestDataCompat() . . . 1476
12.391.2.7 vtkBooleanMacro() . . . 1476
12.391.2.8 vtkGetMacro() . . . 1476
12.391.2.9 vtkSetMacro() [1/3] . . . 1477
12.391.2.10 vtkSetMacro() [2/3] . . . 1477
12.391.2.11 vtkSetMacro() [3/3] . . . 1477
12.391.2.12 vtkTypeMacro() . . . 1477
12.392 vtkGDCMThreadedImageReader2 Class Reference . . . 1478
12.392.1 Constructor & Destructor Documentation . . . 1479
12.392.1.1 vtkGDCMThreadedImageReader2() . . . 1479
12.392.1.2 ~vtkGDCMThreadedImageReader2() . . . 1479
12.392.2 Member Function Documentation . . . 1480
12.392.2.1 GetFileName() . . . 1480
12.392.2.2 New() . . . 1480
12.392.2.3 PrintSelf() . . . 1480
12.392.2.4 RequestInformation() . . . 1480
12.392.2.5 SetFileName() . . . 1480
12.392.2.6 SetFileNames() . . . 1480
12.392.2.7 SplitExtent() . . . 1480
12.392.2.8 ThreadedRequestData() . . . 1481
12.392.2.9 vtkBooleanMacro() [1/3] . . . 1481
12.392.2.10 vtkBooleanMacro() [2/3] . . . 1481
12.392.2.11 vtkBooleanMacro() [3/3] . . . 1481
12.392.2.12 vtkGetMacro() [1/8] . . . 1481
12.392.2.13 vtkGetMacro() [2/8] . . . 1481

12.392.2.14 vtkGetMacro() [3/8] . . . 1482
12.392.2.15 vtkGetMacro() [4/8] . . . 1482
12.392.2.16 vtkGetMacro() [5/8] . . . 1482
12.392.2.17 vtkGetMacro() [6/8] . . . 1482
12.392.2.18 vtkGetMacro() [7/8] . . . 1482
12.392.2.19 vtkGetMacro() [8/8] . . . 1482
12.392.2.20 vtkGetObjectMacro() . . . 1482
12.392.2.21 vtkGetVector3Macro() [1/2] . . . 1483
12.392.2.22 vtkGetVector3Macro() [2/2] . . . 1483
12.392.2.23 vtkGetVector6Macro() . . . 1483
12.392.2.24 vtkSetMacro() [1/7] . . . 1483
12.392.2.25 vtkSetMacro() [2/7] . . . 1483
12.392.2.26 vtkSetMacro() [3/7] . . . 1483
12.392.2.27 vtkSetMacro() [4/7] . . . 1483
12.392.2.28 vtkSetMacro() [5/7] . . . 1484
12.392.2.29 vtkSetMacro() [6/7] . . . 1484
12.392.2.30 vtkSetMacro() [7/7] . . . 1484
12.392.2.31 vtkSetVector3Macro() [1/2] . . . 1484
12.392.2.32 vtkSetVector3Macro() [2/2] . . . 1484
12.392.2.33 vtkSetVector6Macro() . . . 1484
12.392.2.34 vtkTypeMacro() . . . 1484
12.393 vtkImageColorViewer Class Reference . . . 1485
12.393.1 Detailed Description . . . 1487
12.393.2 Member Enumeration Documentation . . . 1487
12.393.2.1 anonymous enum . . . 1487
12.393.3 Constructor & Destructor Documentation . . . 1488
12.393.3.1 vtkImageColorViewer() . . . 1488
12.393.3.2 ~vtkImageColorViewer() . . . 1488
12.393.4 Member Function Documentation . . . 1488
12.393.4.1 AddInput() . . . 1488
12.393.4.2 AddInputConnection() . . . 1488
12.393.4.3 GetColorLevel() . . . 1488
12.393.4.4 GetColorWindow() . . . 1488
12.393.4.5 GetInput() . . . 1488
12.393.4.6 GetOffScreenRendering() . . . 1489
12.393.4.7 GetOverlayVisibility() . . . 1489
12.393.4.8 GetPosition() . . . 1489
12.393.4.9 GetSize() . . . 1489
12.393.4.10 GetSliceMax() . . . 1489

12.393.4.11 GetSliceMin() . . . 1489
12.393.4.12 GetSliceRange() [1/3] . . . 1489
12.393.4.13 GetSliceRange() [2/3] . . . 1489
12.393.4.14 GetSliceRange() [3/3] . . . 1489
12.393.4.15 GetWindowName() . . . 1490
12.393.4.16 InstallPipeline() . . . 1490
12.393.4.17 New() . . . 1490
12.393.4.18 PrintSelf() . . . 1490
12.393.4.19 Render() . . . 1490
12.393.4.20 SetColorLevel() . . . 1490
12.393.4.21 SetColorWindow() . . . 1490
12.393.4.22 SetDisplayId() . . . 1491
12.393.4.23 SetInput() . . . 1491
12.393.4.24 SetInputConnection() . . . 1491
12.393.4.25 SetOffScreenRendering() . . . 1491
12.393.4.26 SetOverlayVisibility() . . . 1491
12.393.4.27 SetParentId() . . . 1491
12.393.4.28 SetPosition() [1/2] . . . 1491
12.393.4.29 SetPosition() [2/2] . . . 1492
12.393.4.30 SetRenderer() . . . 1492
12.393.4.31 SetRenderWindow() . . . 1492
12.393.4.32 SetSize() [1/2] . . . 1492
12.393.4.33 SetSize() [2/2] . . . 1492
12.393.4.34 SetSlice() . . . 1492
12.393.4.35 SetSliceOrientation() . . . 1493
12.393.4.36 SetSliceOrientationToXY() . . . 1493
12.393.4.37 SetSliceOrientationToXZ() . . . 1493
12.393.4.38 SetSliceOrientationToYZ() . . . 1493
12.393.4.39 SetupInteractor() . . . 1493
12.393.4.40 SetWindowId() . . . 1493
12.393.4.41 UnInstallPipeline() . . . 1493
12.393.4.42 UpdateDisplayExtent() . . . 1494
12.393.4.43 UpdateOrientation() . . . 1494
12.393.4.44 vtkBooleanMacro() . . . 1494
12.393.4.45 vtkGetMacro() [1/2] . . . 1494
12.393.4.46 vtkGetMacro() [2/2] . . . 1494
12.393.4.47 vtkGetObjectMacro() [1/5] . . . 1494
12.393.4.48 vtkGetObjectMacro() [2/5] . . . 1494
12.393.4.49 vtkGetObjectMacro() [3/5] . . . 1495

12.393.4.50 vtkGetObjectMacro() [4/5] . . . 1495
12.393.4.51 vtkGetObjectMacro() [5/5] . . . 1495
12.393.4.52 vtkTypeMacro() . . . 1495
12.393.5 Friends And Related Symbol Documentation . . . 1495
12.393.5.1 vtkImageColorViewerCallback . . . 1495
12.393.6 Member Data Documentation . . . 1496
12.393.6.1 FirstRender . . . 1496
12.393.6.2 ImageActor . . . 1496
12.393.6.3 Interactor . . . 1496
12.393.6.4 InteractorStyle . . . 1496
12.393.6.5 OverlayImageActor . . . 1496
12.393.6.6 Renderer . . . 1496
12.393.6.7 RenderWindow . . . 1496
12.393.6.8 Slice . . . 1497
12.393.6.9 SliceOrientation . . . 1497
12.393.6.10 WindowLevel . . . 1497
12.394 vtkImageMapToColors16 Class Reference . . . 1497
12.394.1 Constructor & Destructor Documentation . . . 1499
12.394.1.1 vtkImageMapToColors16() . . . 1499
12.394.1.2 ∼vtkImageMapToColors16() . . . 1499
12.394.2 Member Function Documentation . . . 1499
12.394.2.1 GetMTime() . . . 1499
12.394.2.2 New() . . . 1499
12.394.2.3 PrintSelf() . . . 1499
12.394.2.4 RequestData() . . . 1500
12.394.2.5 RequestInformation() . . . 1500
12.394.2.6 SetLookupTable() . . . 1500
12.394.2.7 SetOutputFormatToLuminance() . . . 1500
12.394.2.8 SetOutputFormatToLuminanceAlpha() . . . 1500
12.394.2.9 SetOutputFormatToRGB() . . . 1500
12.394.2.10 SetOutputFormatToRGBA() . . . 1500
12.394.2.11 ThreadedRequestData() . . . 1501
12.394.2.12 vtkBooleanMacro() . . . 1501
12.394.2.13 vtkGetMacro() [1/3] . . . 1501
12.394.2.14 vtkGetMacro() [2/3] . . . 1501
12.394.2.15 vtkGetMacro() [3/3] . . . 1501
12.394.2.16 vtkGetObjectMacro() . . . 1502
12.394.2.17 vtkSetMacro() [1/3] . . . 1502
12.394.2.18 vtkSetMacro() [2/3] . . . 1502

12.394.2.19 vtkSetMacro() [3/3] . . . 1502
12.394.2.20 vtkTypeMacro() . . . 1502
12.394.3 Member Data Documentation . . . 1502
12.394.3.1 ActiveComponent . . . 1502
12.394.3.2 DataWasPassed . . . 1503
12.394.3.3 LookupTable . . . 1503
12.394.3.4 OutputFormat . . . 1503
12.394.3.5 PassAlphaToOutput . . . 1503
12.395 vtkImageMapToWindowLevelColors2 Class Reference . . . 1503
12.395.1 Constructor & Destructor Documentation . . . 1505
12.395.1.1 vtkImageMapToWindowLevelColors2() . . . 1505
12.395.1.2 ~vtkImageMapToWindowLevelColors2() . . . 1505
12.395.2 Member Function Documentation . . . 1505
12.395.2.1 New() . . . 1505
12.395.2.2 PrintSelf() . . . 1505
12.395.2.3 RequestData() . . . 1505
12.395.2.4 RequestInformation() . . . 1505
12.395.2.5 ThreadedRequestData() . . . 1506
12.395.2.6 vtkGetMacro() [1/2] . . . 1506
12.395.2.7 vtkGetMacro() [2/2] . . . 1506
12.395.2.8 vtkSetMacro() [1/2] . . . 1506
12.395.2.9 vtkSetMacro() [2/2] . . . 1506
12.395.2.10 vtkTypeMacro() . . . 1507
12.395.3 Member Data Documentation . . . 1507
12.395.3.1 Level . . . 1507
12.395.3.2 Window . . . 1507
12.396 vtkImagePlanarComponentsToComponents Class Reference . . . 1507
12.396.1 Constructor & Destructor Documentation . . . 1508
12.396.1.1 vtkImagePlanarComponentsToComponents() . . . 1508
12.396.1.2 ~vtkImagePlanarComponentsToComponents() . . . 1508
12.396.2 Member Function Documentation . . . 1509
12.396.2.1 New() . . . 1509
12.396.2.2 PrintSelf() . . . 1509
12.396.2.3 RequestData() . . . 1509
12.396.2.4 vtkTypeMacro() . . . 1509
12.397 vtkImageRGBToYBR Class Reference . . . 1510
12.397.1 Constructor & Destructor Documentation . . . 1511
12.397.1.1 vtkImageRGBToYBR() . . . 1511
12.397.1.2 ~vtkImageRGBToYBR() . . . 1511

12.397.2 Member Function Documentation . . . 1511
12.397.2.1 New() . . . 1511
12.397.2.2 PrintSelf() . . . 1511
12.397.2.3 ThreadedExecute() . . . 1511
12.397.2.4 vtkTypeMacro() . . . 1512
12.398 vtkImageYBRToRGB Class Reference . . . 1512
12.398.1 Constructor & Destructor Documentation . . . 1513
12.398.1.1 vtkImageYBRToRGB() . . . 1513
12.398.1.2 ∼vtkImageYBRToRGB() . . . 1513
12.398.2 Member Function Documentation . . . 1513
12.398.2.1 New() . . . 1513
12.398.2.2 PrintSelf() . . . 1513
12.398.2.3 ThreadedExecute() . . . 1514
12.398.2.4 vtkTypeMacro() . . . 1514
12.399 vtkLookupTable16 Class Reference . . . 1514
12.399.1 Constructor & Destructor Documentation . . . 1515
12.399.1.1 vtkLookupTable16() . . . 1515
12.399.1.2 ∼vtkLookupTable16() . . . 1516
12.399.2 Member Function Documentation . . . 1516
12.399.2.1 Build() . . . 1516
12.399.2.2 GetPointer() . . . 1516
12.399.2.3 MapScalarsThroughTable2() . . . 1516
12.399.2.4 New() . . . 1516
12.399.2.5 PrintSelf() . . . 1516
12.399.2.6 SetNumberOfTableValues() . . . 1517
12.399.2.7 vtkTypeMacro() . . . 1517
12.399.2.8 WritePointer() . . . 1517
12.399.3 Member Data Documentation . . . 1517
12.399.3.1 Table16 . . . 1517
12.400 vtkRTStructSetProperties Class Reference . . . 1518
12.400.1 Detailed Description . . . 1520
12.400.2 Constructor & Destructor Documentation . . . 1520
12.400.2.1 vtkRTStructSetProperties() . . . 1520
12.400.2.2 ∼vtkRTStructSetProperties() . . . 1520
12.400.3 Member Function Documentation . . . 1520
12.400.3.1 AddContourReferencedFrameOfReference() . . . 1520
12.400.3.2 AddReferencedFrameOfReference() . . . 1520
12.400.3.3 AddStructureSetROI() . . . 1521
12.400.3.4 AddStructureSetROIObservation() . . . 1521

12.400.3.5 Clear() 1521
12.400.3.6 DeepCopy() 1521
12.400.3.7 GetContourReferencedFrameOfReferenceClassUID() 1521
12.400.3.8 GetContourReferencedFrameOfReferenceInstanceUID() 1521
12.400.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2] 1521
12.400.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2] 1522
12.400.3.11 GetNumberOfReferencedFrameOfReferences() 1522
12.400.3.12 GetNumberOfStructureSetROIs() 1522
12.400.3.13 GetReferencedFrameOfReferenceClassUID() 1522
12.400.3.14 GetReferencedFrameOfReferenceInstanceUID() 1522
12.400.3.15 GetStructureSetObservationNumber() 1522
12.400.3.16 GetStructureSetROIDescription() 1522
12.400.3.17 GetStructureSetROIGenerationAlgorithm() 1522
12.400.3.18 GetStructureSetROIName() 1523
12.400.3.19 GetStructureSetROINumber() 1523
12.400.3.20 GetStructureSetROIObservationLabel() 1523
12.400.3.21 GetStructureSetROIRefFrameRefUID() 1523
12.400.3.22 GetStructureSetRTROIInterpretedType() 1523
12.400.3.23 New() 1523
12.400.3.24 PrintSelf() 1523
12.400.3.25 vtkGetStringMacro() [1/9] 1524
12.400.3.26 vtkGetStringMacro() [2/9] 1524
12.400.3.27 vtkGetStringMacro() [3/9] 1524
12.400.3.28 vtkGetStringMacro() [4/9] 1524
12.400.3.29 vtkGetStringMacro() [5/9] 1524
12.400.3.30 vtkGetStringMacro() [6/9] 1524
12.400.3.31 vtkGetStringMacro() [7/9] 1525
12.400.3.32 vtkGetStringMacro() [8/9] 1525
12.400.3.33 vtkGetStringMacro() [9/9] 1525
12.400.3.34 vtkSetStringMacro() [1/9] 1525
12.400.3.35 vtkSetStringMacro() [2/9] 1525
12.400.3.36 vtkSetStringMacro() [3/9] 1525
12.400.3.37 vtkSetStringMacro() [4/9] 1526
12.400.3.38 vtkSetStringMacro() [5/9] 1526
12.400.3.39 vtkSetStringMacro() [6/9] 1526
12.400.3.40 vtkSetStringMacro() [7/9] 1526
12.400.3.41 vtkSetStringMacro() [8/9] 1526
12.400.3.42 vtkSetStringMacro() [9/9] 1526
12.400.3.43 vtkTypeMacro() 1527

12.400.4 Member Data Documentation . . . 1527
12.400.4.1 Internals . . . 1527
12.400.4.2 ReferenceFrameOfReferenceUID . . . 1527
12.400.4.3 ReferenceSeriesInstanceUID . . . 1527
12.400.4.4 SeriesInstanceUID . . . 1527
12.400.4.5 SOPInstanceUID . . . 1527
12.400.4.6 StructureSetDate . . . 1527
12.400.4.7 StructureSetLabel . . . 1528
12.400.4.8 StructureSetName . . . 1528
12.400.4.9 StructureSetTime . . . 1528
12.400.4.10 StudyInstanceUID . . . 1528
12.401 gdcm::Waveform Class Reference . . . 1528
12.401.1 Detailed Description . . . 1528
12.401.2 Constructor & Destructor Documentation . . . 1529
12.401.2.1 Waveform() . . . 1529
12.402 gdcm::WLMFindQuery Class Reference . . . 1529
12.402.1 Detailed Description . . . 1532
12.402.2 Constructor & Destructor Documentation . . . 1532
12.402.2.1 WLMFindQuery() . . . 1532
12.402.3 Member Function Documentation . . . 1532
12.402.3.1 GetAbstractSyntaxUID() . . . 1532
12.402.3.2 GetTagListByLevel() . . . 1532
12.402.3.3 GetValidDataSet() . . . 1532
12.402.3.4 InitializeDataSet() . . . 1532
12.402.3.5 ValidateQuery() . . . 1533
12.402.4 Friends And Related Symbol Documentation . . . 1533
12.402.4.1 QueryFactory . . . 1533
12.403 gdcm::Writer Class Reference . . . 1533
12.403.1 Detailed Description . . . 1535
12.403.2 Constructor & Destructor Documentation . . . 1536
12.403.2.1 Writer() . . . 1536
12.403.2.2 ~Writer() . . . 1536
12.403.3 Member Function Documentation . . . 1536
12.403.3.1 CheckFileMetaInformationOff() . . . 1536
12.403.3.2 CheckFileMetaInformationOn() . . . 1536
12.403.3.3 GetCheckFileMetaInformation() . . . 1536
12.403.3.4 GetFile() . . . 1536
12.403.3.5 GetStreamPtr() . . . 1537
12.403.3.6 SetCheckFileMetaInformation() . . . 1537

12.403.3.7 SetFile() . . . 1537
12.403.3.8 SetFileName() . . . 1537
12.403.3.9 SetStream() . . . 1538
12.403.3.10 SetWriteDataSetOnly() . . . 1538
12.403.3.11 Write() . . . 1538
12.403.4 Friends And Related Symbol Documentation . . . 1538
12.403.4.1 StreamImageWriter . . . 1538
12.403.5 Member Data Documentation . . . 1538
12.403.5.1 Ofstream . . . 1538
12.403.5.2 Stream . . . 1539
12.404 gdcm::XMLDictReader Class Reference . . . 1539
12.404.1 Detailed Description . . . 1540
12.404.2 Constructor & Destructor Documentation . . . 1540
12.404.2.1 XMLDictReader() . . . 1540
12.404.2.2 ∼XMLDictReader() . . . 1541
12.404.3 Member Function Documentation . . . 1541
12.404.3.1 CharacterDataHandler() . . . 1541
12.404.3.2 EndElement() . . . 1541
12.404.3.3 GetDict() . . . 1541
12.404.3.4 HandleDescription() . . . 1541
12.404.3.5 HandleEntry() . . . 1541
12.404.3.6 StartElement() . . . 1541
12.405 gdcm::XMLPrinter Class Reference . . . 1542
12.405.1 Member Enumeration Documentation . . . 1543
12.405.1.1 PrintStyles . . . 1543
12.405.2 Constructor & Destructor Documentation . . . 1543
12.405.2.1 XMLPrinter() . . . 1543
12.405.2.2 ∼XMLPrinter() . . . 1543
12.405.3 Member Function Documentation . . . 1543
12.405.3.1 GetPrintStyle() . . . 1543
12.405.3.2 HandleBulkData() . . . 1543
12.405.3.3 Print() . . . 1544
12.405.3.4 PrintDataElement() . . . 1544
12.405.3.5 PrintDataSet() . . . 1544
12.405.3.6 PrintSQ() . . . 1544
12.405.3.7 SetFile() . . . 1544
12.405.3.8 SetStyle() . . . 1544
12.405.4 Member Data Documentation . . . 1545
12.405.4.1 F . . . 1545

12.405.4.2 PrintStyle . . . 1545
12.406 gdcm::XMLPrivateDictReader Class Reference . . . 1545
12.406.1 Detailed Description . . . 1547
12.406.2 Constructor & Destructor Documentation . . . 1547
12.406.2.1 XMLPrivateDictReader() . . . 1547
12.406.2.2 ∼XMLPrivateDictReader() . . . 1547
12.406.3 Member Function Documentation . . . 1547
12.406.3.1 CharacterDataHandler() . . . 1547
12.406.3.2 EndElement() . . . 1547
12.406.3.3 GetPrivateDict() . . . 1547
12.406.3.4 HandleDescription() . . . 1548
12.406.3.5 HandleEntry() . . . 1548
12.406.3.6 StartElement() . . . 1548

13 File Documentation **1549**
13.1 README.txt File Reference . . . 1549
13.2 TestsList.txt File Reference . . . 1549
13.3 gdcmASN1.h File Reference . . . 1549
13.4 gdcmASN1.h . . . 1550
13.5 gdcmBase64.h File Reference . . . 1551
13.6 gdcmBase64.h . . . 1551
13.7 gdcmBoxRegion.h File Reference . . . 1552
13.8 gdcmBoxRegion.h . . . 1553
13.9 gdcmByteSwap.h File Reference . . . 1553
13.10 gdcmByteSwap.h . . . 1554
13.11 gdcmCAPICryptoFactory.h File Reference . . . 1555
13.12 gdcmCAPICryptoFactory.h . . . 1556
13.13 gdcmCAPICryptographicMessageSyntax.h File Reference . . . 1556
13.14 gdcmCAPICryptographicMessageSyntax.h . . . 1557
13.15 gdcmCommand.h File Reference . . . 1559
13.16 gdcmCommand.h . . . 1560
13.17 gdcmCryptoFactory.h File Reference . . . 1562
13.18 gdcmCryptoFactory.h . . . 1563
13.19 gdcmCryptographicMessageSyntax.h File Reference . . . 1564
13.20 gdcmCryptographicMessageSyntax.h . . . 1565
13.21 gdcmDataEvent.h File Reference . . . 1566
13.22 gdcmDataEvent.h . . . 1567
13.23 gdcmDeflateStream.h File Reference . . . 1568
13.24 gdcmDeflateStream.h . . . 1568

13.25 gdcmDirectory.h File Reference . . . 1568
13.26 gdcmDirectory.h . . . 1569
13.27 gdcmDummyValueGenerator.h File Reference . . . 1571
13.28 gdcmDummyValueGenerator.h . . . 1571
13.29 gdcmEvent.h File Reference . . . 1572
13.29.1 Macro Definition Documentation . . . 1573
13.29.1.1 gdcmEventMacro . . . 1573
13.30 gdcmEvent.h . . . 1574
13.31 gdcmException.h File Reference . . . 1575
13.32 gdcmException.h . . . 1576
13.33 gdcmFilename.h File Reference . . . 1577
13.34 gdcmFilename.h . . . 1578
13.35 gdcmFileNameEvent.h File Reference . . . 1578
13.36 gdcmFileNameEvent.h . . . 1579
13.37 gdcmFilenameGenerator.h File Reference . . . 1580
13.38 gdcmFilenameGenerator.h . . . 1581
13.39 gdcmLegacyMacro.h File Reference . . . 1581
13.39.1 Macro Definition Documentation . . . 1582
13.39.1.1 GDCM_LEGACY . . . 1582
13.39.1.2 GDCM_LEGACY_BODY . . . 1582
13.39.1.3 GDCM_LEGACY_REPLACED_BODY . . . 1583
13.39.1.4 GDCM_NOOP_STATEMENT . . . 1583
13.40 gdcmLegacyMacro.h . . . 1583
13.41 gdcmMD5.h File Reference . . . 1584
13.42 gdcmMD5.h . . . 1585
13.43 gdcmObject.h File Reference . . . 1585
13.44 gdcmObject.h . . . 1586
13.45 gdcmOpenSSLCryptoFactory.h File Reference . . . 1588
13.46 gdcmOpenSSLCryptoFactory.h . . . 1589
13.47 gdcmOpenSSLCryptographicMessageSyntax.h File Reference . . . 1589
13.48 gdcmOpenSSLCryptographicMessageSyntax.h . . . 1591
13.49 gdcmOpenSSLP7CryptoFactory.h File Reference . . . 1591
13.50 gdcmOpenSSLP7CryptoFactory.h . . . 1592
13.51 gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference . . . 1593
13.52 gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1594
13.53 gdcmProgressEvent.h File Reference . . . 1595
13.54 gdcmProgressEvent.h . . . 1596
13.55 gdcmRegion.h File Reference . . . 1596
13.56 gdcmRegion.h . . . 1598

13.57 gdcmSHA1.h File Reference . . . 1599
13.58 gdcmSHA1.h . . . 1599
13.59 gdcmSmartPointer.h File Reference . . . 1600
13.60 gdcmSmartPointer.h . . . 1601
13.61 gdcmStaticAssert.h File Reference . . . 1602
13.61.1 Macro Definition Documentation . . . 1603
13.61.1.1 GDCM_DO_JOIN . . . 1603
13.61.1.2 GDCM_DO_JOIN2 . . . 1603
13.61.1.3 GDCM_JOIN . . . 1603
13.61.1.4 GDCM_STATIC_ASSERT . . . 1603
13.62 gdcmStaticAssert.h . . . 1604
13.63 gdcmString.h File Reference . . . 1604
13.64 gdcmString.h . . . 1606
13.65 gdcmSubject.h File Reference . . . 1608
13.66 gdcmSubject.h . . . 1608
13.67 gdcmSwapCode.h File Reference . . . 1609
13.68 gdcmSwapCode.h . . . 1610
13.69 gdcmSwapper.h File Reference . . . 1611
13.70 gdcmSwapper.h . . . 1612
13.71 gdcmSystem.h File Reference . . . 1614
13.72 gdcmSystem.h . . . 1614
13.73 gdcmTerminal.h File Reference . . . 1616
13.74 gdcmTerminal.h . . . 1617
13.75 gdcmTestDriver.h File Reference . . . 1618
13.76 gdcmTestDriver.h . . . 1618
13.77 gdcmTesting.h File Reference . . . 1619
13.78 gdcmTesting.h . . . 1619
13.79 gdcmTrace.h File Reference . . . 1621
13.79.1 Macro Definition Documentation . . . 1622
13.79.1.1 GDCM_FUNCTION . . . 1622
13.79.1.2 gdcmAssertAlwaysMacro . . . 1622
13.79.1.3 gdcmAssertMacro . . . 1623
13.79.1.4 gdcmDebugMacro . . . 1623
13.79.1.5 gdcmErrorMacro . . . 1624
13.79.1.6 gdcmWarningMacro . . . 1624
13.80 gdcmTrace.h . . . 1625
13.81 gdcmTypes.h File Reference . . . 1627
13.82 gdcmTypes.h . . . 1627
13.83 gdcmUnpacker12Bits.h File Reference . . . 1628

13.84 gdcmUnpacker12Bits.h . . . 1629
13.85 gdcmVersion.h File Reference . . . 1629
13.86 gdcmVersion.h . . . 1630
13.87 gdcmWin32.h File Reference . . . 1630
13.87.1 Macro Definition Documentation . . . 1631
13.87.1.1 GDCM_EXPORT . . . 1631
13.88 gdcmWin32.h . . . 1631
13.89 gdcmCSAHeaderDict.h File Reference . . . 1632
13.90 gdcmCSAHeaderDict.h . . . 1633
13.91 gdcmCSAHeaderDictEntry.h File Reference . . . 1635
13.92 gdcmCSAHeaderDictEntry.h . . . 1636
13.93 gdcmDict.h File Reference . . . 1638
13.94 gdcmDict.h . . . 1639
13.95 gdcmDictConverter.h File Reference . . . 1643
13.96 gdcmDictConverter.h . . . 1644
13.97 gdcmDictEntry.h File Reference . . . 1645
13.98 gdcmDictEntry.h . . . 1646
13.99 gdcmDicts.h File Reference . . . 1648
13.100 gdcmDicts.h . . . 1649
13.101 gdcmGlobal.h File Reference . . . 1650
13.102 gdcmGlobal.h . . . 1651
13.103 gdcmGroupDict.h File Reference . . . 1652
13.104 gdcmGroupDict.h . . . 1653
13.105 gdcmSOPClassUIDToIOD.h File Reference . . . 1654
13.106 gdcmSOPClassUIDToIOD.h . . . 1654
13.107 gdcmUIDs.h File Reference . . . 1655
13.108 gdcmUIDs.h . . . 1656
13.109 gdcmAttribute.h File Reference . . . 1669
13.110 gdcmAttribute.h . . . 1670
13.111 gdcmBasicOffsetTable.h File Reference . . . 1683
13.112 gdcmBasicOffsetTable.h . . . 1684
13.113 gdcmByteBuffer.h File Reference . . . 1686
13.114 gdcmByteBuffer.h . . . 1687
13.115 gdcmByteSwapFilter.h File Reference . . . 1689
13.116 gdcmByteSwapFilter.h . . . 1689
13.117 gdcmByteValue.h File Reference . . . 1690
13.118 gdcmByteValue.h . . . 1691
13.119 gdcmCodeString.h File Reference . . . 1694
13.120 gdcmCodeString.h . . . 1695

13.121 gdcmCP246ExplicitDataElement.h File Reference . . . 1696
13.122 gdcmCP246ExplicitDataElement.h . . . 1696
13.123 gdcmCSAElement.h File Reference . . . 1697
13.124 gdcmCSAElement.h . . . 1699
13.125 gdcmCSAHeader.h File Reference . . . 1701
13.126 gdcmCSAHeader.h . . . 1701
13.127 gdcmDataElement.h File Reference . . . 1703
13.128 gdcmDataElement.h . . . 1704
13.129 gdcmDataSet.h File Reference . . . 1707
13.130 gdcmDataSet.h . . . 1708
13.131 gdcmDataSetEvent.h File Reference . . . 1711
13.132 gdcmDataSetEvent.h . . . 1712
13.133 gdcmElement.h File Reference . . . 1713
13.134 gdcmElement.h . . . 1714
13.135 gdcmExplicitDataElement.h File Reference . . . 1725
13.136 gdcmExplicitDataElement.h . . . 1726
13.137 gdcmExplicitImplicitDataElement.h File Reference . . . 1727
13.138 gdcmExplicitImplicitDataElement.h . . . 1728
13.139 gdcmFile.h File Reference . . . 1729
13.140 gdcmFile.h . . . 1730
13.141 gdcmFileMetaInformation.h File Reference . . . 1730
13.142 gdcmFileMetaInformation.h . . . 1732
13.143 gdcmFileSet.h File Reference . . . 1733
13.144 gdcmFileSet.h . . . 1735
13.145 gdcmFragment.h File Reference . . . 1735
13.146 gdcmFragment.h . . . 1737
13.147 gdcmImplicitDataElement.h File Reference . . . 1740
13.148 gdcmImplicitDataElement.h . . . 1741
13.149 gdcmItem.h File Reference . . . 1741
13.150 gdcmItem.h . . . 1743
13.151 gdcmLO.h File Reference . . . 1747
13.152 gdcmLO.h . . . 1747
13.153 gdcmMediaStorage.h File Reference . . . 1748
13.154 gdcmMediaStorage.h . . . 1749
13.155 gdcmMrProtocol.h File Reference . . . 1752
13.156 gdcmMrProtocol.h . . . 1753
13.157 gdcmParseException.h File Reference . . . 1754
13.158 gdcmParseException.h . . . 1755
13.159 gdcmParser.h File Reference . . . 1756

13.160 gdcmParser.h . . . 1757
13.161 gdcmPDBElement.h File Reference . . . 1759
13.162 gdcmPDBElement.h . . . 1760
13.163 gdcmPDBHeader.h File Reference . . . 1761
13.164 gdcmPDBHeader.h . . . 1762
13.165 gdcmPreamble.h File Reference . . . 1763
13.166 gdcmPreamble.h . . . 1764
13.167 gdcmPrivateTag.h File Reference . . . 1765
13.168 gdcmPrivateTag.h . . . 1766
13.169 gdcmReader.h File Reference . . . 1767
13.170 gdcmReader.h . . . 1768
13.171 gdcmSequenceOfFragments.h File Reference . . . 1769
13.172 gdcmSequenceOfFragments.h . . . 1770
13.173 gdcmSequenceOfItems.h File Reference . . . 1774
13.174 gdcmSequenceOfItems.h . . . 1775
13.175 gdcmTag.h File Reference . . . 1778
13.176 gdcmTag.h . . . 1780
13.177 gdcmTagToVR.h File Reference . . . 1783
13.178 gdcmTagToVR.h . . . 1783
13.179 gdcmTransferSyntax.h File Reference . . . 1784
13.180 gdcmTransferSyntax.h . . . 1785
13.181 gdcmUNExplicitDataElement.h File Reference . . . 1786
13.182 gdcmUNExplicitDataElement.h . . . 1787
13.183 gdcmUNExplicitImplicitDataElement.h File Reference . . . 1788
13.184 gdcmUNExplicitImplicitDataElement.h . . . 1789
13.185 gdcmValue.h File Reference . . . 1789
13.186 gdcmValue.h . . . 1790
13.187 gdcmValueIO.h File Reference . . . 1791
13.188 gdcmValueIO.h . . . 1792
13.189 gdcmVL.h File Reference . . . 1792
13.190 gdcmVL.h . . . 1793
13.191 gdcmVM.h File Reference . . . 1795
13.191.1 Macro Definition Documentation . . . 1796
13.191.1.1 TYPETOLENGTH . . . 1796
13.192 gdcmVM.h . . . 1797
13.193 gdcmVR.h File Reference . . . 1798
13.193.1 Macro Definition Documentation . . . 1800
13.193.1.1 TYPETOENCODING . . . 1800
13.193.1.2 VRTypeTemplateCase . . . 1800

13.194 gdcmVR.h . . . 1801
13.195 gdcmVR16ExplicitDataElement.h File Reference . . . 1805
13.196 gdcmVR16ExplicitDataElement.h . . . 1806
13.197 gdcmWriter.h File Reference . . . 1807
13.198 gdcmWriter.h . . . 1808
13.199 gdcmDefinedTerms.h File Reference . . . 1809
13.200 gdcmDefinedTerms.h . . . 1810
13.201 gdcmDefs.h File Reference . . . 1810
13.202 gdcmDefs.h . . . 1812
13.203 gdcmEnumeratedValues.h File Reference . . . 1813
13.204 gdcmEnumeratedValues.h . . . 1813
13.205 gdcmIOD.h File Reference . . . 1814
13.206 gdcmIOD.h . . . 1815
13.207 gdcmIODEntry.h File Reference . . . 1817
13.208 gdcmIODEntry.h . . . 1819
13.209 gdcmIODs.h File Reference . . . 1819
13.210 gdcmIODs.h . . . 1821
13.211 gdcmMacro.h File Reference . . . 1822
13.212 gdcmMacro.h . . . 1823
13.213 gdcmMacroEntry.h File Reference . . . 1825
13.213.1 Macro Definition Documentation . . . 1826
13.213.1.1 GDCMMACROENTRY_H . . . 1826
13.214 gdcmMacroEntry.h . . . 1827
13.215 gdcmMacros.h File Reference . . . 1828
13.216 gdcmMacros.h . . . 1829
13.217 gdcmModule.h File Reference . . . 1830
13.218 gdcmModule.h . . . 1832
13.219 gdcmModuleEntry.h File Reference . . . 1833
13.220 gdcmModuleEntry.h . . . 1835
13.221 gdcmModules.h File Reference . . . 1836
13.222 gdcmModules.h . . . 1837
13.223 gdcmNestedModuleEntries.h File Reference . . . 1838
13.224 gdcmNestedModuleEntries.h . . . 1840
13.225 gdcmPatient.h File Reference . . . 1840
13.226 gdcmPatient.h . . . 1841
13.227 gdcmSeries.h File Reference . . . 1842
13.228 gdcmSeries.h . . . 1843
13.229 gdcmStudy.h File Reference . . . 1843
13.230 gdcmStudy.h . . . 1845

13.231 gdcmTable.h File Reference . . . 1845
13.232 gdcmTable.h . . . 1846
13.233 gdcmTableEntry.h File Reference . . . 1847
13.234 gdcmTableEntry.h . . . 1849
13.235 gdcmTableReader.h File Reference . . . 1849
13.236 gdcmTableReader.h . . . 1851
13.237 gdcmType.h File Reference . . . 1852
13.238 gdcmType.h . . . 1853
13.239 gdcmUsage.h File Reference . . . 1854
13.240 gdcmUsage.h . . . 1857
13.241 gdcmXMLDictReader.h File Reference . . . 1857
13.242 gdcmXMLDictReader.h . . . 1858
13.243 gdcmXMLPrivateDictReader.h File Reference . . . 1859
13.244 gdcmXMLPrivateDictReader.h . . . 1860
13.245 gdcmAnonymizeEvent.h File Reference . . . 1860
13.246 gdcmAnonymizeEvent.h . . . 1862
13.247 gdcmAnonymizer.h File Reference . . . 1862
13.248 gdcmAnonymizer.h . . . 1863
13.249 gdcmApplicationEntity.h File Reference . . . 1864
13.250 gdcmApplicationEntity.h . . . 1865
13.251 gdcmAudioCodec.h File Reference . . . 1866
13.252 gdcmAudioCodec.h . . . 1867
13.253 gdcmBitmap.h File Reference . . . 1867
13.254 gdcmBitmap.h . . . 1868
13.255 gdcmBitmapToBitmapFilter.h File Reference . . . 1871
13.256 gdcmBitmapToBitmapFilter.h . . . 1872
13.257 gdcmCleaner.h File Reference . . . 1872
13.258 gdcmCleaner.h . . . 1873
13.259 gdcmCodec.h File Reference . . . 1874
13.260 gdcmCodec.h . . . 1875
13.261 gdcmCoder.h File Reference . . . 1876
13.262 gdcmCoder.h . . . 1877
13.263 gdcmConstCharWrapper.h File Reference . . . 1877
13.264 gdcmConstCharWrapper.h . . . 1878
13.265 gdcmCurve.h File Reference . . . 1878
13.266 gdcmCurve.h . . . 1880
13.267 gdcmDataSetHelper.h File Reference . . . 1881
13.268 gdcmDataSetHelper.h . . . 1881
13.269 gdcmDecoder.h File Reference . . . 1882

13.270 gdcmDecoder.h . . . 1883
13.271 gdcmDeltaEncodingCodec.h File Reference . . . 1884
13.272 gdcmDeltaEncodingCodec.h . . . 1884
13.273 gdcmDICOMDIR.h File Reference . . . 1885
13.274 gdcmDICOMDIR.h . . . 1886
13.275 gdcmDICOMDIRGenerator.h File Reference . . . 1886
13.276 gdcmDICOMDIRGenerator.h . . . 1887
13.277 gdcmDictPrinter.h File Reference . . . 1888
13.278 gdcmDictPrinter.h . . . 1889
13.279 gdcmDirectionCosines.h File Reference . . . 1889
13.280 gdcmDirectionCosines.h . . . 1890
13.281 gdcmDirectoryHelper.h File Reference . . . 1891
13.282 gdcmDirectoryHelper.h . . . 1891
13.283 gdcmDPath.h File Reference . . . 1892
13.284 gdcmDPath.h . . . 1893
13.285 gdcmDumper.h File Reference . . . 1894
13.286 gdcmDumper.h . . . 1895
13.287 gdcmEmptyMaskGenerator.h File Reference . . . 1896
13.288 gdcmEmptyMaskGenerator.h . . . 1896
13.289 gdcmEncapsulatedDocument.h File Reference . . . 1897
13.290 gdcmEncapsulatedDocument.h . . . 1898
13.291 gdcmEquipmentManufacturer.h File Reference . . . 1898
13.292 gdcmEquipmentManufacturer.h . . . 1899
13.293 gdcmFiducials.h File Reference . . . 1900
13.294 gdcmFiducials.h . . . 1900
13.295 gdcmFileAnonymizer.h File Reference . . . 1901
13.296 gdcmFileAnonymizer.h . . . 1902
13.297 gdcmFileChangeTransferSyntax.h File Reference . . . 1902
13.298 gdcmFileChangeTransferSyntax.h . . . 1903
13.299 gdcmFileDecompressLookupTable.h File Reference . . . 1904
13.300 gdcmFileDecompressLookupTable.h . . . 1905
13.301 gdcmFileDerivation.h File Reference . . . 1906
13.302 gdcmFileDerivation.h . . . 1906
13.303 gdcmFileExplicitFilter.h File Reference . . . 1908
13.304 gdcmFileExplicitFilter.h . . . 1908
13.305 gdcmFileStreamer.h File Reference . . . 1909
13.306 gdcmFileStreamer.h . . . 1910
13.307 gdcmIconImage.h File Reference . . . 1911
13.308 gdcmIconImage.h . . . 1912

13.309 gdcmIconImageFilter.h File Reference . . . 1913
13.310 gdcmIconImageFilter.h . . . 1914
13.311 gdcmIconImageGenerator.h File Reference . . . 1915
13.312 gdcmIconImageGenerator.h . . . 1916
13.313 gdcmImage.h File Reference . . . 1916
13.314 gdcmImage.h . . . 1918
13.315 gdcmImageApplyLookupTable.h File Reference . . . 1919
13.316 gdcmImageApplyLookupTable.h . . . 1919
13.317 gdcmImageChangePhotometricInterpretation.h File Reference . . . 1920
13.318 gdcmImageChangePhotometricInterpretation.h . . . 1921
13.319 gdcmImageChangePlanarConfiguration.h File Reference . . . 1923
13.320 gdcmImageChangePlanarConfiguration.h . . . 1923
13.321 gdcmImageChangeTransferSyntax.h File Reference . . . 1924
13.322 gdcmImageChangeTransferSyntax.h . . . 1925
13.323 gdcmImageCodec.h File Reference . . . 1926
13.324 gdcmImageCodec.h . . . 1927
13.325 gdcmImageConverter.h File Reference . . . 1929
13.326 gdcmImageConverter.h . . . 1930
13.327 gdcmImageFragmentSplitter.h File Reference . . . 1931
13.328 gdcmImageFragmentSplitter.h . . . 1931
13.329 gdcmImageHelper.h File Reference . . . 1932
13.330 gdcmImageHelper.h . . . 1933
13.331 gdcmImageReader.h File Reference . . . 1934
13.332 gdcmImageReader.h . . . 1936
13.333 gdcmImageRegionReader.h File Reference . . . 1936
13.334 gdcmImageRegionReader.h . . . 1937
13.335 gdcmImageToImageFilter.h File Reference . . . 1938
13.336 gdcmImageToImageFilter.h . . . 1939
13.337 gdcmImageWriter.h File Reference . . . 1939
13.338 gdcmImageWriter.h . . . 1940
13.339 gdcmIPPSorter.h File Reference . . . 1941
13.340 gdcmIPPSorter.h . . . 1942
13.341 gdcmJPEG12Codec.h File Reference . . . 1943
13.342 gdcmJPEG12Codec.h . . . 1943
13.343 gdcmJPEG16Codec.h File Reference . . . 1944
13.344 gdcmJPEG16Codec.h . . . 1945
13.345 gdcmJPEG2000Codec.h File Reference . . . 1946
13.346 gdcmJPEG2000Codec.h . . . 1946
13.347 gdcmJPEG8Codec.h File Reference . . . 1948

13.348 gdcmJPEG8Codec.h . . . 1948
13.349 gdcmJPEGCodec.h File Reference . . . 1949
13.350 gdcmJPEGCodec.h . . . 1950
13.351 gdcmJPEGLSCodec.h File Reference . . . 1952
13.352 gdcmJPEGLSCodec.h . . . 1952
13.353 gdcmJSON.h File Reference . . . 1953
13.354 gdcmJSON.h . . . 1954
13.355 gdcmKAKADUCodec.h File Reference . . . 1955
13.356 gdcmKAKADUCodec.h . . . 1956
13.357 gdcmLookupTable.h File Reference . . . 1956
13.358 gdcmLookupTable.h . . . 1957
13.359 gdcmMEC_MR3.h File Reference . . . 1959
13.360 gdcmMEC_MR3.h . . . 1960
13.361 gdcmMeshPrimitive.h File Reference . . . 1960
13.362 gdcmMeshPrimitive.h . . . 1961
13.363 gdcmOrientation.h File Reference . . . 1963
13.364 gdcmOrientation.h . . . 1963
13.365 gdcmOverlay.h File Reference . . . 1964
13.366 gdcmOverlay.h . . . 1965
13.367 gdcmPDFCodec.h File Reference . . . 1967
13.368 gdcmPDFCodec.h . . . 1968
13.369 gdcmPersonName.h File Reference . . . 1968
13.370 gdcmPersonName.h . . . 1969
13.371 gdcmPGXCodec.h File Reference . . . 1970
13.372 gdcmPGXCodec.h . . . 1971
13.373 gdcmPhotometricInterpretation.h File Reference . . . 1971
13.374 gdcmPhotometricInterpretation.h . . . 1972
13.375 gdcmPixelFormat.h File Reference . . . 1973
13.376 gdcmPixelFormat.h . . . 1975
13.377 gdcmPixmap.h File Reference . . . 1977
13.378 gdcmPixmap.h . . . 1978
13.379 gdcmPixmapReader.h File Reference . . . 1980
13.380 gdcmPixmapReader.h . . . 1981
13.381 gdcmPixmapToPixmapFilter.h File Reference . . . 1982
13.382 gdcmPixmapToPixmapFilter.h . . . 1982
13.383 gdcmPixmapWriter.h File Reference . . . 1983
13.384 gdcmPixmapWriter.h . . . 1984
13.385 gdcmPNMCodec.h File Reference . . . 1985
13.386 gdcmPNMCodec.h . . . 1986

13.387 gdcmPrinter.h File Reference . . . 1986
13.388 gdcmPrinter.h . . . 1988
13.389 gdcmPVRGCodec.h File Reference . . . 1989
13.390 gdcmPVRGCodec.h . . . 1990
13.391 gdcmRAWCodec.h File Reference . . . 1991
13.392 gdcmRAWCodec.h . . . 1991
13.393 gdcmRescaler.h File Reference . . . 1992
13.394 gdcmRescaler.h . . . 1993
13.395 gdcmRLECodec.h File Reference . . . 1994
13.396 gdcmRLECodec.h . . . 1994
13.397 gdcmScanner.h File Reference . . . 1995
13.398 gdcmScanner.h . . . 1996
13.399 gdcmScanner2.h File Reference . . . 1998
13.400 gdcmScanner2.h . . . 1999
13.401 gdcmSegment.h File Reference . . . 2001
13.402 gdcmSegment.h . . . 2003
13.403 gdcmSegmentedPaletteColorLookupTable.h File Reference . . . 2005
13.404 gdcmSegmentedPaletteColorLookupTable.h . . . 2005
13.405 gdcmSegmentHelper.h File Reference . . . 2006
13.406 gdcmSegmentHelper.h . . . 2007
13.407 gdcmSegmentReader.h File Reference . . . 2008
13.408 gdcmSegmentReader.h . . . 2010
13.409 gdcmSegmentWriter.h File Reference . . . 2010
13.410 gdcmSegmentWriter.h . . . 2012
13.411 gdcmSerieHelper.h File Reference . . . 2012
13.412 gdcmSerieHelper.h . . . 2014
13.413 gdcmSimpleSubjectWatcher.h File Reference . . . 2015
13.414 gdcmSimpleSubjectWatcher.h . . . 2016
13.415 gdcmSorter.h File Reference . . . 2017
13.416 gdcmSorter.h . . . 2019
13.417 gdcmSpacing.h File Reference . . . 2020
13.418 gdcmSpacing.h . . . 2020
13.419 gdcmSpectroscopy.h File Reference . . . 2021
13.420 gdcmSpectroscopy.h . . . 2022
13.421 gdcmSplitMosaicFilter.h File Reference . . . 2022
13.422 gdcmSplitMosaicFilter.h . . . 2023
13.423 gdcmStreamImageReader.h File Reference . . . 2025
13.424 gdcmStreamImageReader.h . . . 2025
13.425 gdcmStreamImageWriter.h File Reference . . . 2026

13.426 gdcmStreamImageWriter.h . . . 2027
13.427 gdcmStrictScanner.h File Reference . . . 2028
13.428 gdcmStrictScanner.h . . . 2029
13.429 gdcmStrictScanner2.h File Reference . . . 2031
13.430 gdcmStrictScanner2.h . . . 2032
13.431 gdcmStringFilter.h File Reference . . . 2034
13.432 gdcmStringFilter.h . . . 2035
13.433 gdcmSurface.h File Reference . . . 2036
13.434 gdcmSurface.h . . . 2037
13.435 gdcmSurfaceHelper.h File Reference . . . 2040
13.436 gdcmSurfaceHelper.h . . . 2041
13.437 gdcmSurfaceReader.h File Reference . . . 2043
13.438 gdcmSurfaceReader.h . . . 2044
13.439 gdcmSurfaceWriter.h File Reference . . . 2045
13.440 gdcmSurfaceWriter.h . . . 2046
13.441 gdcmTagPath.h File Reference . . . 2046
13.442 gdcmTagPath.h . . . 2047
13.443 gdcmUIDGenerator.h File Reference . . . 2048
13.444 gdcmUIDGenerator.h . . . 2049
13.445 gdcmUUIDGenerator.h File Reference . . . 2050
13.446 gdcmUUIDGenerator.h . . . 2050
13.447 gdcmValidate.h File Reference . . . 2051
13.448 gdcmValidate.h . . . 2052
13.449 gdcmWaveform.h File Reference . . . 2052
13.450 gdcmWaveform.h . . . 2053
13.451 gdcmXMLPrinter.h File Reference . . . 2053
13.452 gdcmXMLPrinter.h . . . 2054
13.453 gdcmAAbortPDU.h File Reference . . . 2056
13.454 gdcmAAbortPDU.h . . . 2057
13.455 gdcmAAssociateACPDU.h File Reference . . . 2057
13.456 gdcmAAssociateACPDU.h . . . 2058
13.457 gdcmAAssociateRJPDU.h File Reference . . . 2060
13.458 gdcmAAssociateRJPDU.h . . . 2060
13.459 gdcmAAssociateRQPDU.h File Reference . . . 2061
13.460 gdcmAAssociateRQPDU.h . . . 2062
13.461 gdcmAbstractSyntax.h File Reference . . . 2064
13.462 gdcmAbstractSyntax.h . . . 2065
13.463 gdcmApplicationContext.h File Reference . . . 2066
13.464 gdcmApplicationContext.h . . . 2067

13.465 gdcmAReleaseRPPDU.h File Reference . . . 2067
13.466 gdcmAReleaseRPPDU.h . . . 2068
13.467 gdcmAReleaseRQPDU.h File Reference . . . 2069
13.468 gdcmAReleaseRQPDU.h . . . 2070
13.469 gdcmARTIMTimer.h File Reference . . . 2070
13.470 gdcmARTIMTimer.h . . . 2071
13.471 gdcmAsynchronousOperationsWindowSub.h File Reference . . . 2072
13.472 gdcmAsynchronousOperationsWindowSub.h . . . 2072
13.473 gdcmBaseCompositeMessage.h File Reference . . . 2073
13.474 gdcmBaseCompositeMessage.h . . . 2074
13.475 gdcmBaseNormalizedMessage.h File Reference . . . 2075
13.476 gdcmBaseNormalizedMessage.h . . . 2076
13.477 gdcmBasePDU.h File Reference . . . 2076
13.478 gdcmBasePDU.h . . . 2077
13.479 gdcmBaseQuery.h File Reference . . . 2078
13.480 gdcmBaseQuery.h . . . 2079
13.481 gdcmBaseRootQuery.h File Reference . . . 2080
13.482 gdcmBaseRootQuery.h . . . 2081
13.483 gdcmCEchoMessages.h File Reference . . . 2083
13.484 gdcmCEchoMessages.h . . . 2083
13.485 gdcmCFindMessages.h File Reference . . . 2084
13.486 gdcmCFindMessages.h . . . 2085
13.487 gdcmCMoveMessages.h File Reference . . . 2085
13.488 gdcmCMoveMessages.h . . . 2086
13.489 gdcmCommandDataSet.h File Reference . . . 2087
13.490 gdcmCommandDataSet.h . . . 2088
13.491 gdcmCompositeMessageFactory.h File Reference . . . 2089
13.492 gdcmCompositeMessageFactory.h . . . 2089
13.493 gdcmCompositeNetworkFunctions.h File Reference . . . 2090
13.494 gdcmCompositeNetworkFunctions.h . . . 2091
13.495 gdcmCStoreMessages.h File Reference . . . 2092
13.496 gdcmCStoreMessages.h . . . 2092
13.497 gdcmDIMSE.h File Reference . . . 2093
13.498 gdcmDIMSE.h . . . 2094
13.499 gdcmFindPatientRootQuery.h File Reference . . . 2095
13.500 gdcmFindPatientRootQuery.h . . . 2096
13.501 gdcmFindStudyRootQuery.h File Reference . . . 2097
13.502 gdcmFindStudyRootQuery.h . . . 2097
13.503 gdcmImplementationClassUIDSub.h File Reference . . . 2098

13.504 gdcmImplementationClassUIDSub.h . . . 2099
13.505 gdcmImplementationUIDSub.h File Reference . . . 2100
13.506 gdcmImplementationUIDSub.h . . . 2101
13.507 gdcmImplementationVersionNameSub.h File Reference . . . 2101
13.508 gdcmImplementationVersionNameSub.h . . . 2102
13.509 gdcmMaximumLengthSub.h File Reference . . . 2103
13.510 gdcmMaximumLengthSub.h . . . 2104
13.511 gdcmModalityPerformedProcedureStepCreateQuery.h File Reference . . . 2105
13.512 gdcmModalityPerformedProcedureStepCreateQuery.h . . . 2105
13.513 gdcmModalityPerformedProcedureStepSetQuery.h File Reference . . . 2106
13.514 gdcmModalityPerformedProcedureStepSetQuery.h . . . 2107
13.515 gdcmMovePatientRootQuery.h File Reference . . . 2107
13.516 gdcmMovePatientRootQuery.h . . . 2108
13.517 gdcmMoveStudyRootQuery.h File Reference . . . 2109
13.518 gdcmMoveStudyRootQuery.h . . . 2109
13.519 gdcmNActionMessages.h File Reference . . . 2110
13.520 gdcmNActionMessages.h . . . 2111
13.521 gdcmNCreateMessages.h File Reference . . . 2111
13.522 gdcmNCreateMessages.h . . . 2112
13.523 gdcmNDeleteMessages.h File Reference . . . 2113
13.524 gdcmNDeleteMessages.h . . . 2113
13.525 gdcmNetworkEvents.h File Reference . . . 2114
13.526 gdcmNetworkEvents.h . . . 2115
13.527 gdcmNetworkStateID.h File Reference . . . 2116
13.528 gdcmNetworkStateID.h . . . 2117
13.529 gdcmNEventReportMessages.h File Reference . . . 2118
13.530 gdcmNEventReportMessages.h . . . 2119
13.531 gdcmNGetMessages.h File Reference . . . 2119
13.532 gdcmNGetMessages.h . . . 2120
13.533 gdcmNormalizedMessageFactory.h File Reference . . . 2120
13.534 gdcmNormalizedMessageFactory.h . . . 2121
13.535 gdcmNormalizedNetworkFunctions.h File Reference . . . 2122
13.536 gdcmNormalizedNetworkFunctions.h . . . 2123
13.537 gdcmNSetMessages.h File Reference . . . 2124
13.538 gdcmNSetMessages.h . . . 2124
13.539 gdcmPDataTFPDU.h File Reference . . . 2125
13.540 gdcmPDataTFPDU.h . . . 2126
13.541 gdcmPDUFactory.h File Reference . . . 2127
13.542 gdcmPDUFactory.h . . . 2127

13.543 gdcmPresentationContext.h File Reference 2128
13.544 gdcmPresentationContext.h 2130
13.545 gdcmPresentationContextAC.h File Reference 2130
13.546 gdcmPresentationContextAC.h 2132
13.547 gdcmPresentationContextGenerator.h File Reference 2132
13.548 gdcmPresentationContextGenerator.h 2133
13.549 gdcmPresentationContextRQ.h File Reference 2134
13.550 gdcmPresentationContextRQ.h 2135
13.551 gdcmPresentationDataValue.h File Reference 2136
13.552 gdcmPresentationDataValue.h 2137
13.553 gdcmQueryBase.h File Reference 2138
13.554 gdcmQueryBase.h 2140
13.555 gdcmQueryFactory.h File Reference 2141
13.556 gdcmQueryFactory.h 2142
13.557 gdcmQueryImage.h File Reference 2142
13.558 gdcmQueryImage.h 2143
13.559 gdcmQueryPatient.h File Reference 2144
13.560 gdcmQueryPatient.h 2145
13.561 gdcmQuerySeries.h File Reference 2146
13.562 gdcmQuerySeries.h 2146
13.563 gdcmQueryStudy.h File Reference 2147
13.564 gdcmQueryStudy.h 2148
13.565 gdcmRoleSelectionSub.h File Reference 2149
13.566 gdcmRoleSelectionSub.h 2149
13.567 gdcmServiceClassApplicationInformation.h File Reference 2150
13.568 gdcmServiceClassApplicationInformation.h 2151
13.569 gdcmServiceClassUser.h File Reference 2152
13.570 gdcmServiceClassUser.h 2153
13.571 gdcmSOPClassExtendedNegociationSub.h File Reference 2154
13.572 gdcmSOPClassExtendedNegociationSub.h 2155
13.573 gdcmTransferSyntaxSub.h File Reference 2155
13.574 gdcmTransferSyntaxSub.h 2157
13.575 gdcmULAction.h File Reference 2157
13.576 gdcmULAction.h 2158
13.577 gdcmULActionAA.h File Reference 2159
13.578 gdcmULActionAA.h 2160
13.579 gdcmULActionAE.h File Reference 2161
13.580 gdcmULActionAE.h 2162
13.581 gdcmULActionAR.h File Reference 2163

13.582 gdcmULActionAR.h . . . 2164
13.583 gdcmULActionDT.h File Reference . . . 2166
13.584 gdcmULActionDT.h . . . 2166
13.585 gdcmULBasicCallback.h File Reference . . . 2167
13.586 gdcmULBasicCallback.h . . . 2168
13.587 gdcmULConnection.h File Reference . . . 2168
13.588 gdcmULConnection.h . . . 2169
13.589 gdcmULConnectionCallback.h File Reference . . . 2171
13.590 gdcmULConnectionCallback.h . . . 2172
13.591 gdcmULConnectionInfo.h File Reference . . . 2172
13.592 gdcmULConnectionInfo.h . . . 2174
13.593 gdcmULConnectionManager.h File Reference . . . 2174
13.594 gdcmULConnectionManager.h . . . 2175
13.595 gdcmULEvent.h File Reference . . . 2177
13.596 gdcmULEvent.h . . . 2178
13.597 gdcmULTransitionTable.h File Reference . . . 2179
13.598 gdcmULTransitionTable.h . . . 2180
13.599 gdcmULWritingCallback.h File Reference . . . 2182
13.600 gdcmULWritingCallback.h . . . 2182
13.601 gdcmUserInformation.h File Reference . . . 2183
13.602 gdcmUserInformation.h . . . 2184
13.603 gdcmWLMFindQuery.h File Reference . . . 2185
13.604 gdcmWLMFindQuery.h . . . 2186
13.605 vtkGDCMImageReader.h File Reference . . . 2186
13.605.1 Macro Definition Documentation . . . 2188
13.605.1.1 VTK_CMYK . . . 2188
13.605.1.2 VTK_INVERSE_LUMINANCE . . . 2188
13.605.1.3 VTK_LOOKUP_TABLE . . . 2188
13.605.1.4 VTK_YBR . . . 2188
13.606 vtkGDCMImageReader.h . . . 2188
13.607 vtkGDCMImageReader2.h File Reference . . . 2192
13.607.1 Macro Definition Documentation . . . 2193
13.607.1.1 VTK_CMYK . . . 2193
13.607.1.2 VTK_INVERSE_LUMINANCE . . . 2193
13.607.1.3 VTK_LOOKUP_TABLE . . . 2193
13.607.1.4 VTK_YBR . . . 2193
13.608 vtkGDCMImageReader2.h . . . 2194
13.609 vtkGDCMImageWriter.h File Reference . . . 2197
13.610 vtkGDCMImageWriter.h . . . 2198

13.611 vtkGDCMMedicalImageProperties.h File Reference . . . 2200
13.612 vtkGDCMMedicalImageProperties.h . . . 2201
13.613 vtkGDCMPolyDataReader.h File Reference . . . 2206
13.614 vtkGDCMPolyDataReader.h . . . 2206
13.615 vtkGDCMPolyDataWriter.h File Reference . . . 2207
13.616 vtkGDCMPolyDataWriter.h . . . 2208
13.617 vtkGDCMTesting.h File Reference . . . 2209
13.618 vtkGDCMTesting.h . . . 2210
13.619 vtkGDCMThreadedImageReader.h File Reference . . . 2211
13.620 vtkGDCMThreadedImageReader.h . . . 2211
13.621 vtkGDCMThreadedImageReader2.h File Reference . . . 2213
13.622 vtkGDCMThreadedImageReader2.h . . . 2213
13.623 vtkImageColorViewer.h File Reference . . . 2215
13.624 vtkImageColorViewer.h . . . 2216
13.625 vtkImageMapToColors16.h File Reference . . . 2219
13.626 vtkImageMapToColors16.h . . . 2219
13.627 vtkImageMapToWindowLevelColors2.h File Reference . . . 2221
13.628 vtkImageMapToWindowLevelColors2.h . . . 2221
13.629 vtkImagePlanarComponentsToComponents.h File Reference . . . 2223
13.630 vtkImagePlanarComponentsToComponents.h . . . 2223
13.631 vtkImageRGBToYBR.h File Reference . . . 2224
13.632 vtkImageRGBToYBR.h . . . 2225
13.633 vtkImageYBRToRGB.h File Reference . . . 2226
13.634 vtkImageYBRToRGB.h . . . 2226
13.635 vtkLookupTable16.h File Reference . . . 2227
13.636 vtkLookupTable16.h . . . 2228
13.637 vtkRTStructSetProperties.h File Reference . . . 2229
13.638 vtkRTStructSetProperties.h . . . 2230
13.639 gdcmPythonFilter.h File Reference . . . 2231
13.640 gdcmPythonFilter.h . . . 2232

14 Examples **2235**

14.1 TestByteSwap.cxx . . . 2235
14.2 PatchFile.cxx . . . 2237
14.3 SimplePrint.cs . . . 2238
14.4 TestReader.cxx . . . 2240
14.5 TestReader.py . . . 2241
14.6 DecompressJPEGFile.cs . . . 2241
14.7 ManipulateFile.cs . . . 2242

14.8 ClinicalTrialIdentificationWorkflow.cs . . . 2243
14.9 GenerateDICOMDIR.cs . . . 2246
14.10 GenFakeImage.cxx . . . 2247
14.11 ReformatFile.cs . . . 2249
14.12 DecompressImage.cs . . . 2250
14.13 StandardizeFiles.cs . . . 2251
14.14 ScanDirectory.cs . . . 2253
14.15 BasicAnonymizer.cs . . . 2254
14.16 BasicImageAnonymizer.cs . . . 2256
14.17 Cleaner.cs . . . 2257
14.18 CompressLossyJPEG.cs . . . 2258
14.19 DecompressImageMultiframe.cs . . . 2259
14.20 DumpCSA.cs . . . 2261
14.21 ExtractEncapsulatedFile.cs . . . 2262
14.22 ExtractImageRegion.cs . . . 2263
14.23 ExtractImageRegionWithLUT.cs . . . 2265
14.24 ExtractOneFrame.cs . . . 2266
14.25 FileAnonymize.cs . . . 2267
14.26 FileChangeTS.cs . . . 2268
14.27 FileChangeTSLossy.cs . . . 2270
14.28 FileStreaming.cs . . . 2273
14.29 GetArray.cs . . . 2274
14.30 MpegVideoInfo.cs . . . 2275
14.31 NewSequence.cs . . . 2279
14.32 RescaleImage.cs . . . 2280
14.33 SendFileSCU.cs . . . 2281
14.34 SimplePrintPatientName.cs . . . 2282
14.35 SortImage2.cs . . . 2283
14.36 CStoreQtProgress.cxx . . . 2283
14.37 ChangePrivateTags.cxx . . . 2285
14.38 ChangeSequenceUltrasound.cxx . . . 2286
14.39 CheckBigEndianBug.cxx . . . 2288
14.40 ClinicalTrialAnnotate.cxx . . . 2289
14.41 CompressImage.cxx . . . 2290
14.42 ConvertToQImage.cxx . . . 2291
14.43 CreateARGBImage.cxx . . . 2293
14.44 CreateCMYKImage.cxx . . . 2294
14.45 CreateJPIPDataSet.cxx . . . 2295
14.46 DeriveSeries.cxx . . . 2296

14.47 DiffFile.cxx . . . 2297
14.48 DiscriminateVolume.cxx . . . 2299
14.49 DumpADAC.cxx . . . 2302
14.50 DumpExamCard.cxx . . . 2307
14.51 DumpGEMSMovieGroup.cxx . . . 2315
14.52 DumpImageHeaderInfo.cxx . . . 2321
14.53 DumpPhilipsECHO.cxx . . . 2324
14.54 DumpSiemensBase64.cxx . . . 2329
14.55 DumpToSQLITE3.cxx . . . 2330
14.56 DumpToshibaDTI.cxx . . . 2332
14.57 DumpToshibaDTI2.cxx . . . 2334
14.58 DumpVisusChange.cxx . . . 2335
14.59 DuplicatePCDE.cxx . . . 2337
14.60 ELSCINT1WaveToText.cxx . . . 2340
14.61 EmptyMask.cxx . . . 2341
14.62 EncapsulateFileInRawData.cxx . . . 2342
14.63 ExtractEncryptedContent.cxx . . . 2343
14.64 ExtractIconFromFile.cxx . . . 2344
14.65 Extracting_All_Resolution.cxx . . . 2345
14.66 Fake_Image_Using_Stream_Image_Writer.cxx . . . 2351
14.67 FixBrokenJ2K.cxx . . . 2354
14.68 FixJAIBugJPEGLS.cxx . . . 2356
14.69 FixOrientation.cxx . . . 2359
14.70 GenAllVR.cxx . . . 2360
14.71 GenFakeIdentifyFile.cxx . . . 2362
14.72 GenLongSeqs.cxx . . . 2365
14.73 GenSeqs.cxx . . . 2366
14.74 GenerateStandardSOPClasses.cxx . . . 2367
14.75 GetJPEGSamplePrecision.cxx . . . 2368
14.76 GetSequenceUltrasound.cxx . . . 2370
14.77 GetSubSequenceData.cxx . . . 2372
14.78 HelloVizWorld.cxx . . . 2374
14.79 HelloWorld.cxx . . . 2375
14.80 LargeVRDSExplicit.cxx . . . 2376
14.81 MakeTemplate.cxx . . . 2379
14.82 MergeTwoFiles.cxx . . . 2379
14.83 MrProtocol.cxx . . . 2381
14.84 PrintLUT.cxx . . . 2387
14.85 PublicDict.cxx . . . 2388

14.86 QIDO-RS.cxx . . . 2389
14.87 ReadAndDumpDICOMDIR.cxx . . . 2390
14.88 ReadAndDumpDICOMDIR2.cxx . . . 2393
14.89 ReadAndPrintAttributes.cxx . . . 2398
14.90 ReadExplicitLengthSQIVR.cxx . . . 2399
14.91 ReadGEMSSDO.cxx . . . 2400
14.92 ReadMultiTimesException.cxx . . . 2402
14.93 ReadUTF8QtDir.cxx . . . 2403
14.94 SimpleScanner.cxx . . . 2404
14.95 SortImage.cxx . . . 2406
14.96 StreamImageReaderTest.cxx . . . 2408
14.97 TemplateEmptyImage.cxx . . . 2411
14.98 TraverseModules.cxx . . . 2413
14.99 VolumeSorter.cxx . . . 2414
14.100 csa2img.cxx . . . 2416
14.101 iU22tomultisc.cxx . . . 2418
14.102 pmsct_rgb1.cxx . . . 2420
14.103 rle2img.cxx . . . 2423
14.104 uid_unique.cxx . . . 2426
14.105 DecompressImage.java . . . 2427
14.106 DecompressPixmap.java . . . 2427
14.107 ExtractImageRegion.java . . . 2428
14.108 FileAnonymize.java . . . 2429
14.109 HelloSimple.java . . . 2430
14.110 ReadFiles.java . . . 2431
14.111 ScanDirectory.java . . . 2432
14.112 SimplePrint.java . . . 2436
14.113 AddPrivateAttribute.py . . . 2437
14.114 ConvertMPL.py . . . 2437
14.115 ConvertNumpy.py . . . 2438
14.116 ConvertPIL.py . . . 2439
14.117 CreateRAWStorage.py . . . 2440
14.118 DecompressImage.py . . . 2442
14.119 DumbAnonymizer.py . . . 2443
14.120 ExtractImageRegion.py . . . 2445
14.121 FindAllPatientName.py . . . 2446
14.122 FixCommaBug.py . . . 2446
14.123 GetPortionCSAHeader.py . . . 2447
14.124 HelloWorld.py . . . 2448

14.125 ManipulateFile.py . . . 2448
14.126 ManipulateSequence.py . . . 2450
14.127 MergeFile.py . . . 2451
14.128 NewSequence.py . . . 2451
14.129 PhilipsPrivateRescaleInterceptSlope.py . . . 2452
14.130 PlaySound.py . . . 2453
14.131 PrivateDict.py . . . 2454
14.132 ReWriteSCAsMR.py . . . 2454
14.133 ReadAndDumpDICOMDIR.py . . . 2455
14.134 RemovePrivateTags.py . . . 2458
14.135 ScanDirectory.py . . . 2458
14.136 SortImage.py . . . 2459
14.137 WriteBuffer.py . . . 2459
14.138 HelloActiviz.cs . . . 2460
14.139 HelloActiviz2.cs . . . 2462
14.140 HelloActiviz3.cs . . . 2463
14.141 HelloActiviz4.cs . . . 2464
14.142 HelloActiviz5.cs . . . 2464
14.143 HelloVTKWorld.cs . . . 2466
14.144 HelloVTKWorld2.cs . . . 2467
14.145 MetaImageMD5Activiz.cs . . . 2467
14.146 RefCounting.cs . . . 2469
14.147 Compute3DSpacing.cxx . . . 2469
14.148 Convert16BitsTo8Bits.cxx . . . 2471
14.149 ConvertMultiFrameToSingleFrame.cxx . . . 2472
14.150 ConvertRGBToLuminance.cxx . . . 2473
14.151 ConvertSingleBitTo8Bits.cxx . . . 2474
14.152 CreateFakePET.cxx . . . 2475
14.153 CreateFakeRTDOSE.cxx . . . 2476
14.154 GenerateRTSTRUCT.cxx . . . 2478
14.155 MagnifyFile.cxx . . . 2481
14.156 gdcmorthoplanes.cxx . . . 2482
14.157 gdcmreslice.cxx . . . 2488
14.158 gdcmrtionplan.cxx . . . 2490
14.159 gdcmrtplan.cxx . . . 2494
14.160 gdcmscene.cxx . . . 2498
14.161 gdcmtexture.cxx . . . 2500
14.162 gdcmvolume.cxx . . . 2502
14.163 offscreenimage.cxx . . . 2503

14.164 reslicesphere.cxx 2504
14.165 rtstructapp.cxx 2512
14.166 threadgdcm.cxx 2514
14.167 AWTMedical3.java 2517
14.168 HelloVTKWorld.java 2521
14.169 MIPViewer.java 2523
14.170 MPRViewer.java 2525
14.171 MPRViewer2.java 2527
14.172 ReadSeriesIntoVTK.java 2531
14.173 CastConvertPhilips.py 2533
14.174 headsq2dcm.py 2535

Index **2537**

Chapter 1

GDCM Documentation

This is the developers documentation.

A PDF version of this doxygen documentation can be found here:

`http://gdcm.sourceforge.net/3.0/gdcm-3.0.24.pdf`

A tarball version of this HTML doxygen documentation can be found here:

`http://gdcm.sourceforge.net/3.0/gdcm-3.0.24-doc.tar.gz`

Author

Mathieu Malaterre

Chapter 2

Todo List

Class gdcm::CSAHeader

MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

Class gdcm::network::ApplicationContext

Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

Class gdcm::Overlay

Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differentiate them (other than the group tag).

Class gdcm::SequenceOfFragments

I do not enforce that Sequence of Fragments ends with a SQ end del

Class gdcm::TransferSyntax

: The implementation is completely retarded -> see gdcm::UIDs for a replacement We need: IsSupported We need preprocess of raw/xml file We need GetFullName()

Member gdcm::UIDGenerator::IsValid (const char *uid)

: Move that in DataStructureAndEncoding (see FileMetaInformation::CheckFileMetaInformation)

Chapter 3

Deprecated List

Member gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType inRootType, EQueryLevel in↩ QueryLevel, const KeyValuePairArrayType &keys, EQueryType queryType=eFind)

Member gdcm::FileSet::AddFile (File const &)

. Does nothing

Member gdcm::TransferSyntax::GetSwapCode () const

Return the SwapCode associated with the Transfer Syntax. Be careful with the special GE private syntax the DataSet is written in little endian but the Pixel Data is in Big Endian.

Chapter 4

Bug List

Class gdcm::DICOMDIRGenerator

: There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the Scanner does not allow us See PS 3.11 / Table D.3-2 STD-GEN Additional DICOMDIR Keys

Member gdcm::FileStreamer::StartGroupDataElement (const PrivateTag &pt, size_t maxsizede=0, uint8_↩t startoffset=0)

maxsizede should be a value lower than the actual total size of the buffer to be copied

Class gdcm::IPPSorter

There are currently a couple of bugs in this implementation:

Chapter 5

Directory Hierarchy

5.1 Directories

Common . . . 57
- gdcmASN1.h . . . 1549
- gdcmBase64.h . . . 1551
- gdcmBoxRegion.h . . . 1552
- gdcmByteSwap.h . . . 1553
- gdcmCAPICryptoFactory.h . . . 1555
- gdcmCAPICryptographicMessageSyntax.h . . . 1556
- gdcmCommand.h . . . 1559
- gdcmCryptoFactory.h . . . 1562
- gdcmCryptographicMessageSyntax.h . . . 1564
- gdcmDataEvent.h . . . 1566
- gdcmDeflateStream.h . . . 1568
- gdcmDirectory.h . . . 1568
- gdcmDummyValueGenerator.h . . . 1571
- gdcmEvent.h . . . 1572
- gdcmException.h . . . 1575
- gdcmFilename.h . . . 1577
- gdcmFileNameEvent.h . . . 1578
- gdcmFilenameGenerator.h . . . 1580
- gdcmLegacyMacro.h . . . 1581
- gdcmMD5.h . . . 1584
- gdcmObject.h . . . 1585
- gdcmOpenSSLCryptoFactory.h . . . 1588
- gdcmOpenSSLCryptographicMessageSyntax.h . . . 1589
- gdcmOpenSSLP7CryptoFactory.h . . . 1591
- gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1593
- gdcmProgressEvent.h . . . 1595
- gdcmRegion.h . . . 1596
- gdcmSHA1.h . . . 1599
- gdcmSmartPointer.h . . . 1600
- gdcmStaticAssert.h . . . 1602
- gdcmString.h . . . 1604
- gdcmSubject.h . . . 1608
- gdcmSwapCode.h . . . 1609

gdcmSwapper.h . . . 1611
gdcmSystem.h . . . 1614
gdcmTerminal.h . . . 1616
gdcmTestDriver.h . . . 1618
gdcmTesting.h . . . 1619
gdcmTrace.h . . . 1621
gdcmTypes.h . . . 1627
gdcmUnpacker12Bits.h . . . 1628
gdcmVersion.h . . . 1629
gdcmWin32.h . . . 1630
DataDictionary . . . 59
gdcmCSAHeaderDict.h . . . 1632
gdcmCSAHeaderDictEntry.h . . . 1635
gdcmDict.h . . . 1638
gdcmDictConverter.h . . . 1643
gdcmDictEntry.h . . . 1645
gdcmDicts.h . . . 1648
gdcmGlobal.h . . . 1650
gdcmGroupDict.h . . . 1652
gdcmSOPClassUIDToIOD.h . . . 1654
gdcmUIDs.h . . . 1655
DataStructureAndEncodingDefinition . . . 60
gdcmAttribute.h . . . 1669
gdcmBasicOffsetTable.h . . . 1683
gdcmByteBuffer.h . . . 1686
gdcmByteSwapFilter.h . . . 1689
gdcmByteValue.h . . . 1690
gdcmCodeString.h . . . 1694
gdcmCP246ExplicitDataElement.h . . . 1696
gdcmCSAElement.h . . . 1697
gdcmCSAHeader.h . . . 1701
gdcmDataElement.h . . . 1703
gdcmDataSet.h . . . 1707
gdcmDataSetEvent.h . . . 1711
gdcmElement.h . . . 1713
gdcmExplicitDataElement.h . . . 1725
gdcmExplicitImplicitDataElement.h . . . 1727
gdcmFile.h . . . 1729
gdcmFileMetaInformation.h . . . 1730
gdcmFileSet.h . . . 1733
gdcmFragment.h . . . 1735
gdcmImplicitDataElement.h . . . 1740
gdcmItem.h . . . 1741
gdcmLO.h . . . 1747
gdcmMediaStorage.h . . . 1748
gdcmMrProtocol.h . . . 1752
gdcmParseException.h . . . 1754
gdcmParser.h . . . 1756
gdcmPDBElement.h . . . 1759
gdcmPDBHeader.h . . . 1761
gdcmPreamble.h . . . 1763
gdcmPrivateTag.h . . . 1765
gdcmReader.h . . . 1767
gdcmSequenceOfFragments.h . . . 1769

gdcmSequenceOfItems.h . . . 1774
gdcmTag.h . . . 1778
gdcmTagToVR.h . . . 1783
gdcmTransferSyntax.h . . . 1784
gdcmUNExplicitDataElement.h . . . 1786
gdcmUNExplicitImplicitDataElement.h . . . 1788
gdcmValue.h . . . 1789
gdcmValueIO.h . . . 1791
gdcmVL.h . . . 1792
gdcmVM.h . . . 1795
gdcmVR.h . . . 1798
gdcmVR16ExplicitDataElement.h . . . 1805
gdcmWriter.h . . . 1807
InformationObjectDefinition . . . 61
gdcmDefinedTerms.h . . . 1809
gdcmDefs.h . . . 1810
gdcmEnumeratedValues.h . . . 1813
gdcmIOD.h . . . 1814
gdcmIODEntry.h . . . 1817
gdcmIODs.h . . . 1819
gdcmMacro.h . . . 1822
gdcmMacroEntry.h . . . 1825
gdcmMacros.h . . . 1828
gdcmModule.h . . . 1830
gdcmModuleEntry.h . . . 1833
gdcmModules.h . . . 1836
gdcmNestedModuleEntries.h . . . 1838
gdcmPatient.h . . . 1840
gdcmSeries.h . . . 1842
gdcmStudy.h . . . 1843
gdcmTable.h . . . 1845
gdcmTableEntry.h . . . 1847
gdcmTableReader.h . . . 1849
gdcmType.h . . . 1852
gdcmUsage.h . . . 1854
gdcmXMLDictReader.h . . . 1857
gdcmXMLPrivateDictReader.h . . . 1859
MediaStorageAndFileFormat . . . 62
gdcmAnonymizeEvent.h . . . 1860
gdcmAnonymizer.h . . . 1862
gdcmApplicationEntity.h . . . 1864
gdcmAudioCodec.h . . . 1866
gdcmBitmap.h . . . 1867
gdcmBitmapToBitmapFilter.h . . . 1871
gdcmCleaner.h . . . 1872
gdcmCodec.h . . . 1874
gdcmCoder.h . . . 1876
gdcmConstCharWrapper.h . . . 1877
gdcmCurve.h . . . 1878
gdcmDataSetHelper.h . . . 1881
gdcmDecoder.h . . . 1882
gdcmDeltaEncodingCodec.h . . . 1884
gdcmDICOMDIR.h . . . 1885
gdcmDICOMDIRGenerator.h . . . 1886

gdcmDictPrinter.h . . . 1888
gdcmDirectionCosines.h . . . 1889
gdcmDirectoryHelper.h . . . 1891
gdcmDPath.h . . . 1892
gdcmDumper.h . . . 1894
gdcmEmptyMaskGenerator.h . . . 1896
gdcmEncapsulatedDocument.h . . . 1897
gdcmEquipmentManufacturer.h . . . 1898
gdcmFiducials.h . . . 1900
gdcmFileAnonymizer.h . . . 1901
gdcmFileChangeTransferSyntax.h . . . 1902
gdcmFileDecompressLookupTable.h . . . 1904
gdcmFileDerivation.h . . . 1906
gdcmFileExplicitFilter.h . . . 1908
gdcmFileStreamer.h . . . 1909
gdcmIconImage.h . . . 1911
gdcmIconImageFilter.h . . . 1913
gdcmIconImageGenerator.h . . . 1915
gdcmImage.h . . . 1916
gdcmImageApplyLookupTable.h . . . 1919
gdcmImageChangePhotometricInterpretation.h . . . 1920
gdcmImageChangePlanarConfiguration.h . . . 1923
gdcmImageChangeTransferSyntax.h . . . 1924
gdcmImageCodec.h . . . 1926
gdcmImageConverter.h . . . 1929
gdcmImageFragmentSplitter.h . . . 1931
gdcmImageHelper.h . . . 1932
gdcmImageReader.h . . . 1934
gdcmImageRegionReader.h . . . 1936
gdcmImageToImageFilter.h . . . 1938
gdcmImageWriter.h . . . 1939
gdcmIPPSorter.h . . . 1941
gdcmJPEG12Codec.h . . . 1943
gdcmJPEG16Codec.h . . . 1944
gdcmJPEG2000Codec.h . . . 1946
gdcmJPEG8Codec.h . . . 1948
gdcmJPEGCodec.h . . . 1949
gdcmJPEGLSCodec.h . . . 1952
gdcmJSON.h . . . 1953
gdcmKAKADUCodec.h . . . 1955
gdcmLookupTable.h . . . 1956
gdcmMEC_MR3.h . . . 1959
gdcmMeshPrimitive.h . . . 1960
gdcmOrientation.h . . . 1963
gdcmOverlay.h . . . 1964
gdcmPDFCodec.h . . . 1967
gdcmPersonName.h . . . 1968
gdcmPGXCodec.h . . . 1970
gdcmPhotometricInterpretation.h . . . 1971
gdcmPixelFormat.h . . . 1973
gdcmPixmap.h . . . 1977
gdcmPixmapReader.h . . . 1980
gdcmPixmapToPixmapFilter.h . . . 1982
gdcmPixmapWriter.h . . . 1983

gdcmPNMCodec.h . . . 1985
gdcmPrinter.h . . . 1986
gdcmPVRGCodec.h . . . 1989
gdcmRAWCodec.h . . . 1991
gdcmRescaler.h . . . 1992
gdcmRLECodec.h . . . 1994
gdcmScanner.h . . . 1995
gdcmScanner2.h . . . 1998
gdcmSegment.h . . . 2001
gdcmSegmentedPaletteColorLookupTable.h . . . 2005
gdcmSegmentHelper.h . . . 2006
gdcmSegmentReader.h . . . 2008
gdcmSegmentWriter.h . . . 2010
gdcmSerieHelper.h . . . 2012
gdcmSimpleSubjectWatcher.h . . . 2015
gdcmSorter.h . . . 2017
gdcmSpacing.h . . . 2020
gdcmSpectroscopy.h . . . 2021
gdcmSplitMosaicFilter.h . . . 2022
gdcmStreamImageReader.h . . . 2025
gdcmStreamImageWriter.h . . . 2026
gdcmStrictScanner.h . . . 2028
gdcmStrictScanner2.h . . . 2031
gdcmStringFilter.h . . . 2034
gdcmSurface.h . . . 2036
gdcmSurfaceHelper.h . . . 2040
gdcmSurfaceReader.h . . . 2043
gdcmSurfaceWriter.h . . . 2045
gdcmTagPath.h . . . 2046
gdcmUIDGenerator.h . . . 2048
gdcmUUIDGenerator.h . . . 2050
gdcmValidate.h . . . 2051
gdcmWaveform.h . . . 2052
gdcmXMLPrinter.h . . . 2053
MessageExchangeDefinition . . . 65
gdcmAAbortPDU.h . . . 2056
gdcmAAssociateACPDU.h . . . 2057
gdcmAAssociateRJPDU.h . . . 2060
gdcmAAssociateRQPDU.h . . . 2061
gdcmAbstractSyntax.h . . . 2064
gdcmApplicationContext.h . . . 2066
gdcmAReleaseRPPDU.h . . . 2067
gdcmAReleaseRQPDU.h . . . 2069
gdcmARTIMTimer.h . . . 2070
gdcmAsynchronousOperationsWindowSub.h . . . 2072
gdcmBaseCompositeMessage.h . . . 2073
gdcmBaseNormalizedMessage.h . . . 2075
gdcmBasePDU.h . . . 2076
gdcmBaseQuery.h . . . 2078
gdcmBaseRootQuery.h . . . 2080
gdcmCEchoMessages.h . . . 2083
gdcmCFindMessages.h . . . 2084
gdcmCMoveMessages.h . . . 2085
gdcmCommandDataSet.h . . . 2087

gdcmCompositeMessageFactory.h . . . 2089
gdcmCompositeNetworkFunctions.h . . . 2090
gdcmCStoreMessages.h . . . 2092
gdcmDIMSE.h . . . 2093
gdcmFindPatientRootQuery.h . . . 2095
gdcmFindStudyRootQuery.h . . . 2097
gdcmImplementationClassUIDSub.h . . . 2098
gdcmImplementationUIDSub.h . . . 2100
gdcmImplementationVersionNameSub.h . . . 2101
gdcmMaximumLengthSub.h . . . 2103
gdcmModalityPerformedProcedureStepCreateQuery.h . . . 2105
gdcmModalityPerformedProcedureStepSetQuery.h . . . 2106
gdcmMovePatientRootQuery.h . . . 2107
gdcmMoveStudyRootQuery.h . . . 2109
gdcmNActionMessages.h . . . 2110
gdcmNCreateMessages.h . . . 2111
gdcmNDeleteMessages.h . . . 2113
gdcmNetworkEvents.h . . . 2114
gdcmNetworkStateID.h . . . 2116
gdcmNEventReportMessages.h . . . 2118
gdcmNGetMessages.h . . . 2119
gdcmNormalizedMessageFactory.h . . . 2120
gdcmNormalizedNetworkFunctions.h . . . 2122
gdcmNSetMessages.h . . . 2124
gdcmPDataTFPDU.h . . . 2125
gdcmPDUFactory.h . . . 2127
gdcmPresentationContext.h . . . 2128
gdcmPresentationContextAC.h . . . 2130
gdcmPresentationContextGenerator.h . . . 2132
gdcmPresentationContextRQ.h . . . 2134
gdcmPresentationDataValue.h . . . 2136
gdcmQueryBase.h . . . 2138
gdcmQueryFactory.h . . . 2141
gdcmQueryImage.h . . . 2142
gdcmQueryPatient.h . . . 2144
gdcmQuerySeries.h . . . 2146
gdcmQueryStudy.h . . . 2147
gdcmRoleSelectionSub.h . . . 2149
gdcmServiceClassApplicationInformation.h . . . 2150
gdcmServiceClassUser.h . . . 2152
gdcmSOPClassExtendedNegociationSub.h . . . 2154
gdcmTransferSyntaxSub.h . . . 2155
gdcmULAction.h . . . 2157
gdcmULActionAA.h . . . 2159
gdcmULActionAE.h . . . 2161
gdcmULActionAR.h . . . 2163
gdcmULActionDT.h . . . 2166
gdcmULBasicCallback.h . . . 2167
gdcmULConnection.h . . . 2168
gdcmULConnectionCallback.h . . . 2171
gdcmULConnectionInfo.h . . . 2172
gdcmULConnectionManager.h . . . 2174
gdcmULEvent.h . . . 2177
gdcmULTransitionTable.h . . . 2179

gdcmULWritingCallback.h . . . 2182
gdcmUserInformation.h . . . 2183
gdcmWLMFindQuery.h . . . 2185
Python . . . 67
gdcmPythonFilter.h . . . 2231
Source . . . 67
Common . . . 57
gdcmASN1.h . . . 1549
gdcmBase64.h . . . 1551
gdcmBoxRegion.h . . . 1552
gdcmByteSwap.h . . . 1553
gdcmCAPICryptoFactory.h . . . 1555
gdcmCAPICryptographicMessageSyntax.h . . . 1556
gdcmCommand.h . . . 1559
gdcmCryptoFactory.h . . . 1562
gdcmCryptographicMessageSyntax.h . . . 1564
gdcmDataEvent.h . . . 1566
gdcmDeflateStream.h . . . 1568
gdcmDirectory.h . . . 1568
gdcmDummyValueGenerator.h . . . 1571
gdcmEvent.h . . . 1572
gdcmException.h . . . 1575
gdcmFilename.h . . . 1577
gdcmFileNameEvent.h . . . 1578
gdcmFilenameGenerator.h . . . 1580
gdcmLegacyMacro.h . . . 1581
gdcmMD5.h . . . 1584
gdcmObject.h . . . 1585
gdcmOpenSSLCryptoFactory.h . . . 1588
gdcmOpenSSLCryptographicMessageSyntax.h . . . 1589
gdcmOpenSSLP7CryptoFactory.h . . . 1591
gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1593
gdcmProgressEvent.h . . . 1595
gdcmRegion.h . . . 1596
gdcmSHA1.h . . . 1599
gdcmSmartPointer.h . . . 1600
gdcmStaticAssert.h . . . 1602
gdcmString.h . . . 1604
gdcmSubject.h . . . 1608
gdcmSwapCode.h . . . 1609
gdcmSwapper.h . . . 1611
gdcmSystem.h . . . 1614
gdcmTerminal.h . . . 1616
gdcmTestDriver.h . . . 1618
gdcmTesting.h . . . 1619
gdcmTrace.h . . . 1621
gdcmTypes.h . . . 1627
gdcmUnpacker12Bits.h . . . 1628
gdcmVersion.h . . . 1629
gdcmWin32.h . . . 1630
DataDictionary . . . 59
gdcmCSAHeaderDict.h . . . 1632
gdcmCSAHeaderDictEntry.h . . . 1635

gdcmDict.h . . . 1638
gdcmDictConverter.h . . . 1643
gdcmDictEntry.h . . . 1645
gdcmDicts.h . . . 1648
gdcmGlobal.h . . . 1650
gdcmGroupDict.h . . . 1652
gdcmSOPClassUIDToIOD.h . . . 1654
gdcmUIDs.h . . . 1655
DataStructureAndEncodingDefinition . . . 60
gdcmAttribute.h . . . 1669
gdcmBasicOffsetTable.h . . . 1683
gdcmByteBuffer.h . . . 1686
gdcmByteSwapFilter.h . . . 1689
gdcmByteValue.h . . . 1690
gdcmCodeString.h . . . 1694
gdcmCP246ExplicitDataElement.h . . . 1696
gdcmCSAElement.h . . . 1697
gdcmCSAHeader.h . . . 1701
gdcmDataElement.h . . . 1703
gdcmDataSet.h . . . 1707
gdcmDataSetEvent.h . . . 1711
gdcmElement.h . . . 1713
gdcmExplicitDataElement.h . . . 1725
gdcmExplicitImplicitDataElement.h . . . 1727
gdcmFile.h . . . 1729
gdcmFileMetaInformation.h . . . 1730
gdcmFileSet.h . . . 1733
gdcmFragment.h . . . 1735
gdcmImplicitDataElement.h . . . 1740
gdcmItem.h . . . 1741
gdcmLO.h . . . 1747
gdcmMediaStorage.h . . . 1748
gdcmMrProtocol.h . . . 1752
gdcmParseException.h . . . 1754
gdcmParser.h . . . 1756
gdcmPDBElement.h . . . 1759
gdcmPDBHeader.h . . . 1761
gdcmPreamble.h . . . 1763
gdcmPrivateTag.h . . . 1765
gdcmReader.h . . . 1767
gdcmSequenceOfFragments.h . . . 1769
gdcmSequenceOfItems.h . . . 1774
gdcmTag.h . . . 1778
gdcmTagToVR.h . . . 1783
gdcmTransferSyntax.h . . . 1784
gdcmUNExplicitDataElement.h . . . 1786
gdcmUNExplicitImplicitDataElement.h . . . 1788
gdcmValue.h . . . 1789
gdcmValueIO.h . . . 1791
gdcmVL.h . . . 1792
gdcmVM.h . . . 1795
gdcmVR.h . . . 1798
gdcmVR16ExplicitDataElement.h . . . 1805
gdcmWriter.h . . . 1807

InformationObjectDefinition . . . 61
gdcmDefinedTerms.h . . . 1809
gdcmDefs.h . . . 1810
gdcmEnumeratedValues.h . . . 1813
gdcmIOD.h . . . 1814
gdcmIODEntry.h . . . 1817
gdcmIODs.h . . . 1819
gdcmMacro.h . . . 1822
gdcmMacroEntry.h . . . 1825
gdcmMacros.h . . . 1828
gdcmModule.h . . . 1830
gdcmModuleEntry.h . . . 1833
gdcmModules.h . . . 1836
gdcmNestedModuleEntries.h . . . 1838
gdcmPatient.h . . . 1840
gdcmSeries.h . . . 1842
gdcmStudy.h . . . 1843
gdcmTable.h . . . 1845
gdcmTableEntry.h . . . 1847
gdcmTableReader.h . . . 1849
gdcmType.h . . . 1852
gdcmUsage.h . . . 1854
gdcmXMLDictReader.h . . . 1857
gdcmXMLPrivateDictReader.h . . . 1859
MediaStorageAndFileFormat . . . 62
gdcmAnonymizeEvent.h . . . 1860
gdcmAnonymizer.h . . . 1862
gdcmApplicationEntity.h . . . 1864
gdcmAudioCodec.h . . . 1866
gdcmBitmap.h . . . 1867
gdcmBitmapToBitmapFilter.h . . . 1871
gdcmCleaner.h . . . 1872
gdcmCodec.h . . . 1874
gdcmCoder.h . . . 1876
gdcmConstCharWrapper.h . . . 1877
gdcmCurve.h . . . 1878
gdcmDataSetHelper.h . . . 1881
gdcmDecoder.h . . . 1882
gdcmDeltaEncodingCodec.h . . . 1884
gdcmDICOMDIR.h . . . 1885
gdcmDICOMDIRGenerator.h . . . 1886
gdcmDictPrinter.h . . . 1888
gdcmDirectionCosines.h . . . 1889
gdcmDirectoryHelper.h . . . 1891
gdcmDPath.h . . . 1892
gdcmDumper.h . . . 1894
gdcmEmptyMaskGenerator.h . . . 1896
gdcmEncapsulatedDocument.h . . . 1897
gdcmEquipmentManufacturer.h . . . 1898
gdcmFiducials.h . . . 1900
gdcmFileAnonymizer.h . . . 1901
gdcmFileChangeTransferSyntax.h . . . 1902
gdcmFileDecompressLookupTable.h . . . 1904
gdcmFileDerivation.h . . . 1906

gdcmFileExplicitFilter.h . . . 1908
gdcmFileStreamer.h . . . 1909
gdcmIconImage.h . . . 1911
gdcmIconImageFilter.h . . . 1913
gdcmIconImageGenerator.h . . . 1915
gdcmImage.h . . . 1916
gdcmImageApplyLookupTable.h . . . 1919
gdcmImageChangePhotometricInterpretation.h . . . 1920
gdcmImageChangePlanarConfiguration.h . . . 1923
gdcmImageChangeTransferSyntax.h . . . 1924
gdcmImageCodec.h . . . 1926
gdcmImageConverter.h . . . 1929
gdcmImageFragmentSplitter.h . . . 1931
gdcmImageHelper.h . . . 1932
gdcmImageReader.h . . . 1934
gdcmImageRegionReader.h . . . 1936
gdcmImageToImageFilter.h . . . 1938
gdcmImageWriter.h . . . 1939
gdcmIPPSorter.h . . . 1941
gdcmJPEG12Codec.h . . . 1943
gdcmJPEG16Codec.h . . . 1944
gdcmJPEG2000Codec.h . . . 1946
gdcmJPEG8Codec.h . . . 1948
gdcmJPEGCodec.h . . . 1949
gdcmJPEGLSCodec.h . . . 1952
gdcmJSON.h . . . 1953
gdcmKAKADUCodec.h . . . 1955
gdcmLookupTable.h . . . 1956
gdcmMEC_MR3.h . . . 1959
gdcmMeshPrimitive.h . . . 1960
gdcmOrientation.h . . . 1963
gdcmOverlay.h . . . 1964
gdcmPDFCodec.h . . . 1967
gdcmPersonName.h . . . 1968
gdcmPGXCodec.h . . . 1970
gdcmPhotometricInterpretation.h . . . 1971
gdcmPixelFormat.h . . . 1973
gdcmPixmap.h . . . 1977
gdcmPixmapReader.h . . . 1980
gdcmPixmapToPixmapFilter.h . . . 1982
gdcmPixmapWriter.h . . . 1983
gdcmPNMCodec.h . . . 1985
gdcmPrinter.h . . . 1986
gdcmPVRGCodec.h . . . 1989
gdcmRAWCodec.h . . . 1991
gdcmRescaler.h . . . 1992
gdcmRLECodec.h . . . 1994
gdcmScanner.h . . . 1995
gdcmScanner2.h . . . 1998
gdcmSegment.h . . . 2001
gdcmSegmentedPaletteColorLookupTable.h . . . 2005
gdcmSegmentHelper.h . . . 2006
gdcmSegmentReader.h . . . 2008
gdcmSegmentWriter.h . . . 2010

gdcmSerieHelper.h . . . 2012
gdcmSimpleSubjectWatcher.h . . . 2015
gdcmSorter.h . . . 2017
gdcmSpacing.h . . . 2020
gdcmSpectroscopy.h . . . 2021
gdcmSplitMosaicFilter.h . . . 2022
gdcmStreamImageReader.h . . . 2025
gdcmStreamImageWriter.h . . . 2026
gdcmStrictScanner.h . . . 2028
gdcmStrictScanner2.h . . . 2031
gdcmStringFilter.h . . . 2034
gdcmSurface.h . . . 2036
gdcmSurfaceHelper.h . . . 2040
gdcmSurfaceReader.h . . . 2043
gdcmSurfaceWriter.h . . . 2045
gdcmTagPath.h . . . 2046
gdcmUIDGenerator.h . . . 2048
gdcmUUIDGenerator.h . . . 2050
gdcmValidate.h . . . 2051
gdcmWaveform.h . . . 2052
gdcmXMLPrinter.h . . . 2053
MessageExchangeDefinition . . . 65
gdcmAAbortPDU.h . . . 2056
gdcmAAssociateACPDU.h . . . 2057
gdcmAAssociateRJPDU.h . . . 2060
gdcmAAssociateRQPDU.h . . . 2061
gdcmAbstractSyntax.h . . . 2064
gdcmApplicationContext.h . . . 2066
gdcmAReleaseRPPDU.h . . . 2067
gdcmAReleaseRQPDU.h . . . 2069
gdcmARTIMTimer.h . . . 2070
gdcmAsynchronousOperationsWindowSub.h . . . 2072
gdcmBaseCompositeMessage.h . . . 2073
gdcmBaseNormalizedMessage.h . . . 2075
gdcmBasePDU.h . . . 2076
gdcmBaseQuery.h . . . 2078
gdcmBaseRootQuery.h . . . 2080
gdcmCEchoMessages.h . . . 2083
gdcmCFindMessages.h . . . 2084
gdcmCMoveMessages.h . . . 2085
gdcmCommandDataSet.h . . . 2087
gdcmCompositeMessageFactory.h . . . 2089
gdcmCompositeNetworkFunctions.h . . . 2090
gdcmCStoreMessages.h . . . 2092
gdcmDIMSE.h . . . 2093
gdcmFindPatientRootQuery.h . . . 2095
gdcmFindStudyRootQuery.h . . . 2097
gdcmImplementationClassUIDSub.h . . . 2098
gdcmImplementationUIDSub.h . . . 2100
gdcmImplementationVersionNameSub.h . . . 2101
gdcmMaximumLengthSub.h . . . 2103
gdcmModalityPerformedProcedureStepCreateQuery.h . . . 2105
gdcmModalityPerformedProcedureStepSetQuery.h . . . 2106
gdcmMovePatientRootQuery.h . . . 2107

gdcmMoveStudyRootQuery.h . . . 2109
gdcmNActionMessages.h . . . 2110
gdcmNCreateMessages.h . . . 2111
gdcmNDeleteMessages.h . . . 2113
gdcmNetworkEvents.h . . . 2114
gdcmNetworkStateID.h . . . 2116
gdcmNEventReportMessages.h . . . 2118
gdcmNGetMessages.h . . . 2119
gdcmNormalizedMessageFactory.h . . . 2120
gdcmNormalizedNetworkFunctions.h . . . 2122
gdcmNSetMessages.h . . . 2124
gdcmPDataTFPDU.h . . . 2125
gdcmPDUFactory.h . . . 2127
gdcmPresentationContext.h . . . 2128
gdcmPresentationContextAC.h . . . 2130
gdcmPresentationContextGenerator.h . . . 2132
gdcmPresentationContextRQ.h . . . 2134
gdcmPresentationDataValue.h . . . 2136
gdcmQueryBase.h . . . 2138
gdcmQueryFactory.h . . . 2141
gdcmQueryImage.h . . . 2142
gdcmQueryPatient.h . . . 2144
gdcmQuerySeries.h . . . 2146
gdcmQueryStudy.h . . . 2147
gdcmRoleSelectionSub.h . . . 2149
gdcmServiceClassApplicationInformation.h . . . 2150
gdcmServiceClassUser.h . . . 2152
gdcmSOPClassExtendedNegociationSub.h . . . 2154
gdcmTransferSyntaxSub.h . . . 2155
gdcmULAction.h . . . 2157
gdcmULActionAA.h . . . 2159
gdcmULActionAE.h . . . 2161
gdcmULActionAR.h . . . 2163
gdcmULActionDT.h . . . 2166
gdcmULBasicCallback.h . . . 2167
gdcmULConnection.h . . . 2168
gdcmULConnectionCallback.h . . . 2171
gdcmULConnectionInfo.h . . . 2172
gdcmULConnectionManager.h . . . 2174
gdcmULEvent.h . . . 2177
gdcmULTransitionTable.h . . . 2179
gdcmULWritingCallback.h . . . 2182
gdcmUserInformation.h . . . 2183
gdcmWLMFindQuery.h . . . 2185
Utilities . . . 68
VTK . . . 68
vtkGDCMImageReader.h . . . 2186
vtkGDCMImageReader2.h . . . 2192
vtkGDCMImageWriter.h . . . 2197
vtkGDCMMedicalImageProperties.h . . . 2200
vtkGDCMPolyDataReader.h . . . 2206
vtkGDCMPolyDataWriter.h . . . 2207
vtkGDCMTesting.h . . . 2209
vtkGDCMThreadedImageReader.h . . . 2211

vtkGDCMThreadedImageReader2.h . . . 2213
vtkImageColorViewer.h . . . 2215
vtkImageMapToColors16.h . . . 2219
vtkImageMapToWindowLevelColors2.h . . . 2221
vtkImagePlanarComponentsToComponents.h . . . 2223
vtkImageRGBToYBR.h . . . 2224
vtkImageYBRToRGB.h . . . 2226
vtkLookupTable16.h . . . 2227
vtkRTStructSetProperties.h . . . 2229
VTK . . . 68
vtkGDCMImageReader.h . . . 2186
vtkGDCMImageReader2.h . . . 2192
vtkGDCMImageWriter.h . . . 2197
vtkGDCMMedicalImageProperties.h . . . 2200
vtkGDCMPolyDataReader.h . . . 2206
vtkGDCMPolyDataWriter.h . . . 2207
vtkGDCMTesting.h . . . 2209
vtkGDCMThreadedImageReader.h . . . 2211
vtkGDCMThreadedImageReader2.h . . . 2213
vtkImageColorViewer.h . . . 2215
vtkImageMapToColors16.h . . . 2219
vtkImageMapToWindowLevelColors2.h . . . 2221
vtkImagePlanarComponentsToComponents.h . . . 2223
vtkImageRGBToYBR.h . . . 2224
vtkImageYBRToRGB.h . . . 2226
vtkLookupTable16.h . . . 2227
vtkRTStructSetProperties.h . . . 2229
Wrapping . . . 69
Python . . . 67
gdcmPythonFilter.h . . . 2231

Chapter 6

Namespace Index

6.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gdcm 71
gdcm::network 102
gdcm::SegmentHelper 109
gdcm::terminal
Class for Terminal 109

Chapter 7

Hierarchical Index

7.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

gdcm::network::AbstractSyntax . . . 129
gdcm::network::ApplicationContext . . . 145
gdcm::ApplicationEntity . . . 147
gdcm::network::ARTIMTimer . . . 154
gdcm::ASN1 . . . 155
gdcm::network::AsynchronousOperationsWindowSub . . . 157
gdcm::Attribute< Group, Element, TVR, TVM > . . . 158
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 168
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 187
  gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 176
  gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 181
  gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 202
    gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 195
  gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 214
    gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 207
gdcm::Base64 . . . 222
gdcm::network::BaseCompositeMessage . . . 225
  gdcm::network::CEchoRQ . . . 290
  gdcm::network::CEchoRSP . . . 292
  gdcm::network::CFindCancelRQ . . . 294
  gdcm::network::CFindRQ . . . 295
  gdcm::network::CFindRSP . . . 297
  gdcm::network::CMoveCancelRq . . . 305
  gdcm::network::CMoveRQ . . . 306
  gdcm::network::CMoveRSP . . . 307
  gdcm::network::CStoreRQ . . . 361
  gdcm::network::CStoreRSP . . . 363
gdcm::network::BaseNormalizedMessage . . . 227
  gdcm::network::NActionRQ . . . 845
  gdcm::network::NActionRSP . . . 846

gdcm::network::NCreateRQ . . . 848
gdcm::network::NCreateRSP . . . 849
gdcm::network::NDeleteRQ . . . 851
gdcm::network::NDeleteRSP . . . 852
gdcm::network::NEventReportRQ . . . 857
gdcm::network::NEventReportRSP . . . 859
gdcm::network::NGetRQ . . . 860
gdcm::network::NGetRSP . . . 862
gdcm::network::NSetRQ . . . 868
gdcm::network::NSetRSP . . . 869
gdcm::network::BasePDU . . . 230
gdcm::network::AAbortPDU . . . 113
gdcm::network::AAssociateACPDU . . . 116
gdcm::network::AAssociateRJPDU . . . 120
gdcm::network::AAssociateRQPDU . . . 122
gdcm::network::AReleaseRPPDU . . . 149
gdcm::network::AReleaseRQPDU . . . 151
gdcm::network::PDataTFPDU . . . 904
std::basic_string< Char >
std::string
gdcm::String<'\', 16 > . . . 1186
gdcm::String<'\', 64 > . . . 1186
gdcm::String<'\', 4294967294 > . . . 1186
gdcm::String<'\', 64, 0 > . . . 1186
gdcm::String< TDelimiter, TMaxLength, TPadChar > . . . 1186
gdcm::SegmentHelper::BasicCodedEntry . . . 242
gdcm::BitmapToBitmapFilter . . . 264
gdcm::PixmapToPixmapFilter . . . 950
gdcm::ImageToImageFilter . . . 689
gdcm::ImageApplyLookupTable . . . 637
gdcm::ImageChangePhotometricInterpretation . . . 640
gdcm::ImageChangePlanarConfiguration . . . 645
gdcm::ImageChangeTransferSyntax . . . 649
gdcm::ImageFragmentSplitter . . . 669
gdcm::ByteBuffer . . . 272
gdcm::ByteSwap< T > . . . 273
gdcm::ByteSwapFilter . . . 275
gdcm::network::CFind . . . 293
gdcm::Coder . . . 310
gdcm::Codec . . . 309
gdcm::AudioCodec . . . 220
gdcm::ImageCodec . . . 655
gdcm::DeltaEncodingCodec . . . 412
gdcm::JPEG2000Codec . . . 737
gdcm::JPEGCodec . . . 750
gdcm::JPEG12Codec . . . 727
gdcm::JPEG16Codec . . . 732
gdcm::JPEG8Codec . . . 745
gdcm::JPEGLSCodec . . . 759
gdcm::KAKADUCodec . . . 768
gdcm::PGXCodec . . . 922
gdcm::PNMCodec . . . 958
gdcm::PVRGCodec . . . 999

gdcm::RAWCodec . . . 1019
gdcm::RLECodec . . . 1039
gdcm::PDFCodec . . . 913
gdcm::CodeString . . . 312
gdcm::network::CompositeMessageFactory . . . 324
gdcm::CompositeNetworkFunctions . . . 325
gdcm::ConstCharWrapper . . . 330
gdcm::CryptoFactory . . . 335
gdcm::CAPICryptoFactory . . . 285
gdcm::OpenSSLCryptoFactory . . . 874
gdcm::OpenSSLP7CryptoFactory . . . 880
gdcm::CryptographicMessageSyntax . . . 337
gdcm::CAPICryptographicMessageSyntax . . . 287
gdcm::OpenSSLCryptographicMessageSyntax . . . 876
gdcm::OpenSSLP7CryptographicMessageSyntax . . . 882
gdcm::CSAElement . . . 341
gdcm::CSAHeader . . . 349
gdcm::CSAHeaderDict . . . 354
gdcm::CSAHeaderDictEntry . . . 357
gdcm::DataElement . . . 370
gdcm::CP246ExplicitDataElement . . . 331
gdcm::ExplicitDataElement . . . 539
gdcm::ExplicitImplicitDataElement . . . 543
gdcm::Fragment . . . 610
gdcm::BasicOffsetTable . . . 246
gdcm::ImplicitDataElement . . . 700
gdcm::Item . . . 720
gdcm::UNExplicitDataElement . . . 1368
gdcm::UNExplicitImplicitDataElement . . . 1372
gdcm::VR16ExplicitDataElement . . . 1411
gdcm::DataSet . . . 388
gdcm::CommandDataSet . . . 320
gdcm::FileMetaInformation . . . 571
gdcm::DataSetHelper . . . 404
gdcm::Decoder . . . 405
gdcm::Codec . . . 309
gdcm::DefinedTerms . . . 407
gdcm::Defs . . . 408
gdcm::DICOMDIR . . . 415
gdcm::DICOMDIRGenerator . . . 416
gdcm::Dict . . . 420
gdcm::DictConverter . . . 424
gdcm::DictEntry . . . 428
gdcm::Dicts . . . 435
gdcm::network::DIMSE . . . 439
gdcm::DirectionCosines . . . 441
gdcm::Directory . . . 445
gdcm::DirectoryHelper . . . 449
gdcm::DPath . . . 451
gdcm::DummyValueGenerator . . . 453
gdcm::Element< TVR, TVM > . . . 457
gdcm::Element< TVR, VM::VM1_n > . . . 468

gdcm::Element< TVR, VM::VM1_2 > . . . 463
gdcm::Element< TVR, VM::VM2_n > . . . 481
gdcm::Element< TVR, VM::VM2_2n > . . . 474
gdcm::Element< TVR, VM::VM3_4 > . . . 493
gdcm::Element< TVR, VM::VM3_n > . . . 498
gdcm::Element< TVR, VM::VM3_3n > . . . 486
gdcm::Element< VR::AS, VM::VM5 > . . . 504
gdcm::Element< VR::OB, VM::VM1_n > . . . 457
gdcm::Element< VR::OB, VM::VM1 > . . . 508
gdcm::Element< VR::OW, VM::VM1_n > . . . 457
gdcm::Element< VR::OW, VM::VM1 > . . . 513
gdcm::ElementDisableCombinations< TVR, TVM > . . . 518
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 519
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 520
gdcm::EmptyMaskGenerator . . . 521
gdcm::EncapsulatedDocument . . . 523
gdcm::EncodingImplementation< T > . . . 524
gdcm::EncodingImplementation< VR::VRASCII > . . . 525
gdcm::EncodingImplementation< VR::VRBINARY > . . . 527
gdcm::EnumeratedValues . . . 530
gdcm::EquipmentManufacturer . . . 531
gdcm::Event . . . 533
gdcm::AnyEvent . . . 144
gdcm::AbortEvent . . . 128
gdcm::AnonymizeEvent . . . 131
gdcm::DataEvent . . . 384
gdcm::DataSetEvent . . . 401
gdcm::EndEvent . . . 529
gdcm::ExitEvent . . . 538
gdcm::FileNameEvent . . . 584
gdcm::InitializeEvent . . . 704
gdcm::IterationEvent . . . 726
gdcm::ModifiedEvent . . . 824
gdcm::ProgressEvent . . . 995
gdcm::StartEvent . . . 1152
gdcm::UserEvent . . . 1380
gdcm::NoEvent . . . 863
std::exception
gdcm::CSAHeaderDictException . . . 361
gdcm::DataElementException . . . 384
gdcm::Exception . . . 536
gdcm::ParseException . . . 898
gdcm::Fiducials . . . 547
gdcm::FileDerivation . . . 564
gdcm::FileExplicitFilter . . . 568
gdcm::Filename . . . 581
gdcm::FilenameGenerator . . . 588
gdcm::FileSet . . . 591
gdcm::Global . . . 615
gdcm::GroupDict . . . 618
gdcm::IconImageFilter . . . 621
gdcm::IconImageGenerator . . . 624
gdcm::ignore_char . . . 627

gdcm::ImageConverter . . . 667
gdcm::ImageHelper . . . 672
gdcm::network::ImplementationClassUIDSub . . . 696
gdcm::network::ImplementationUIDSub . . . 698
gdcm::network::ImplementationVersionNameSub . . . 698
gdcm::IOD . . . 705
gdcm::IODEntry . . . 708
gdcm::IODs . . . 711
gdcm::JSON . . . 766
gdcm::Scanner2::ltstr . . . 783
gdcm::Scanner::ltstr . . . 784
gdcm::StrictScanner2::ltstr . . . 784
gdcm::StrictScanner::ltstr . . . 785
gdcm::Macro . . . 785
gdcm::Macros . . . 788
gdcm::network::MaximumLengthSub . . . 790
gdcm::MD5 . . . 792
gdcm::MEC_MR3 . . . 793
gdcm::MediaStorage . . . 794
gdcm::Module . . . 825
gdcm::ModuleEntry . . . 828
gdcm::NestedModuleEntries . . . 854
gdcm::Modules . . . 832
gdcm::MrProtocol . . . 843
gdcm::network::NormalizedMessageFactory . . . 864
gdcm::NormalizedNetworkFunctions . . . 865
gdcm::Object . . . 871
gdcm::BaseQuery . . . 232
gdcm::BaseRootQuery . . . 237
gdcm::FindPatientRootQuery . . . 602
gdcm::FindStudyRootQuery . . . 606
gdcm::MovePatientRootQuery . . . 835
gdcm::MoveStudyRootQuery . . . 839
gdcm::WLMFindQuery . . . 1529
gdcm::ModalityPerformedProcedureStepCreateQuery . . . 816
gdcm::ModalityPerformedProcedureStepSetQuery . . . 820
gdcm::Bitmap . . . 250
gdcm::Pixmap . . . 939
gdcm::Image . . . 628
gdcm::Curve . . . 364
gdcm::File . . . 548
gdcm::FileWithName . . . 599
gdcm::LookupTable . . . 776
gdcm::SegmentedPaletteColorLookupTable . . . 1077
gdcm::MeshPrimitive . . . 811
gdcm::Overlay . . . 888
gdcm::Segment . . . 1068
gdcm::Subject . . . 1195
gdcm::Anonymizer . . . 135
gdcm::Cleaner . . . 298
gdcm::Command . . . 316
gdcm::MemberCommand< SimpleSubjectWatcher > . . . 804
gdcm::SimpleMemberCommand< SimpleSubjectWatcher > . . . 1124

gdcm::MemberCommand< T > . . . 804
gdcm::SimpleMemberCommand< T > . . . 1124
gdcm::FileAnonymizer . . . 553
gdcm::FileChangeTransferSyntax . . . 557
gdcm::FileDecompressLookupTable . . . 561
gdcm::FileStreamer . . . 593
gdcm::Scanner . . . 1048
gdcm::Scanner2 . . . 1057
gdcm::ServiceClassUser . . . 1114
gdcm::StrictScanner . . . 1166
gdcm::StrictScanner2 . . . 1175
gdcm::network::ULConnectionManager . . . 1355
gdcm::Surface . . . 1198
gdcm::Value . . . 1387
gdcm::ByteValue . . . 277
gdcm::SequenceOfFragments . . . 1090
gdcm::SequenceOfItems . . . 1097
gdcm::Orientation . . . 885
gdcm::Parser . . . 900
gdcm::Patient . . . 904
gdcm::PDBElement . . . 908
gdcm::PDBHeader . . . 911
gdcm::network::PDUFactory . . . 916
gdcm::PersonName . . . 919
gdcm::PhotometricInterpretation . . . 926
gdcm::PixelFormat . . . 930
gdcm::Preamble . . . 963
gdcm::PresentationContext . . . 967
gdcm::network::PresentationContextAC . . . 970
gdcm::PresentationContextGenerator . . . 973
gdcm::network::PresentationContextRQ . . . 976
gdcm::network::PresentationDataValue . . . 979
gdcm::Printer . . . 983
gdcm::DictPrinter . . . 432
gdcm::Dumper . . . 454
gdcm::PrivateDict . . . 987
gdcm::PythonFilter . . . 1003
gdcm::QueryBase . . . 1005
gdcm::QueryImage . . . 1009
gdcm::QueryPatient . . . 1012
gdcm::QuerySeries . . . 1014
gdcm::QueryStudy . . . 1017
gdcm::QueryFactory . . . 1008
gdcm::Reader . . . 1024
gdcm::PixmapReader . . . 946
gdcm::ImageReader . . . 679
gdcm::ImageRegionReader . . . 684
gdcm::SegmentReader . . . 1080
gdcm::SurfaceReader . . . 1214
gdcm::RealWorldValueMappingContent . . . 1031
gdcm::Region . . . 1032
gdcm::BoxRegion . . . 267
gdcm::Rescaler . . . 1035

gdcm::network::RoleSelectionSub . . . 1046
gdcm::SerieHelper . . . 1106
gdcm::Series . . . 1111
gdcm::network::ServiceClassApplicationInformation . . . 1112
gdcm::SHA1 . . . 1122
gdcm::SimpleSubjectWatcher . . . 1129
gdcm::MrProtocol::Slice . . . 1132
gdcm::MrProtocol::SliceArray . . . 1133
gdcm::SmartPointer< ObjectType > . . . 1134
gdcm::network::SOPClassExtendedNegociationSub . . . 1138
gdcm::SOPClassUIDToIOD . . . 1139
gdcm::Sorter . . . 1141
  gdcm::IPPSorter . . . 714
gdcm::Spacing . . . 1146
gdcm::Spectroscopy . . . 1148
gdcm::SplitMosaicFilter . . . 1149
gdcm::static_assert_test< x > . . . 1153
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 1154
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 1154
gdcm::StreamImageReader . . . 1155
gdcm::StreamImageWriter . . . 1159
String<'\', 64 >
  gdcm::LO . . . 772
gdcm::StringFilter . . . 1190
gdcm::Study . . . 1194
gdcm::SurfaceHelper . . . 1211
gdcm::SwapCode . . . 1224
gdcm::SwapperDoOp . . . 1227
gdcm::SwapperNoOp . . . 1228
gdcm::System . . . 1228
gdcm::Table . . . 1235
gdcm::TableEntry . . . 1238
gdcm::TableReader . . . 1239
  gdcm::XMLDictReader . . . 1539
  gdcm::XMLPrivateDictReader . . . 1545
gdcm::network::TableRow . . . 1243
gdcm::Tag . . . 1244
  gdcm::PrivateTag . . . 990
gdcm::TagPath . . . 1255
gdcm::Testing . . . 1257
gdcm::Trace . . . 1263
gdcm::TransferSyntax . . . 1268
gdcm::network::TransferSyntaxSub . . . 1274
gdcm::network::Transition . . . 1276
gdcm::Type . . . 1278
gdcm::UI . . . 1281
gdcm::UIDGenerator . . . 1281
gdcm::UIDs . . . 1284
gdcm::network::ULAction . . . 1302
  gdcm::network::ULActionAA1 . . . 1306
  gdcm::network::ULActionAA2 . . . 1307
  gdcm::network::ULActionAA3 . . . 1308
  gdcm::network::ULActionAA4 . . . 1310

gdcm::network::ULActionAA5 . . . 1311
gdcm::network::ULActionAA6 . . . 1312
gdcm::network::ULActionAA7 . . . 1314
gdcm::network::ULActionAA8 . . . 1315
gdcm::network::ULActionAE1 . . . 1316
gdcm::network::ULActionAE2 . . . 1318
gdcm::network::ULActionAE3 . . . 1319
gdcm::network::ULActionAE4 . . . 1320
gdcm::network::ULActionAE5 . . . 1322
gdcm::network::ULActionAE6 . . . 1323
gdcm::network::ULActionAE7 . . . 1324
gdcm::network::ULActionAE8 . . . 1326
gdcm::network::ULActionAR1 . . . 1327
gdcm::network::ULActionAR10 . . . 1328
gdcm::network::ULActionAR2 . . . 1330
gdcm::network::ULActionAR3 . . . 1331
gdcm::network::ULActionAR4 . . . 1332
gdcm::network::ULActionAR5 . . . 1334
gdcm::network::ULActionAR6 . . . 1335
gdcm::network::ULActionAR7 . . . 1336
gdcm::network::ULActionAR8 . . . 1338
gdcm::network::ULActionAR9 . . . 1339
gdcm::network::ULActionDT1 . . . 1340
gdcm::network::ULActionDT2 . . . 1342
gdcm::network::ULConnection . . . 1346
gdcm::network::ULConnectionCallback . . . 1351
gdcm::network::ULBasicCallback . . . 1343
gdcm::network::ULWritingCallback . . . 1366
gdcm::network::ULConnectionInfo . . . 1353
gdcm::network::ULEvent . . . 1363
gdcm::network::ULTransitionTable . . . 1364
gdcm::Unpacker12Bits . . . 1376
gdcm::Usage . . . 1377
gdcm::network::UserInformation . . . 1381
gdcm::UUIDGenerator . . . 1383
gdcm::Validate . . . 1384
gdcm::ValueIO< TDE, TSwap, TType > . . . 1390
gdcm::MrProtocol::Vector3 . . . 1391
gdcm::Version . . . 1392
gdcm::VL . . . 1394
gdcm::VM . . . 1398
gdcm::VMToLength< T > . . . 1403
gdcm::VR . . . 1403
gdcm::VRToEncoding< T > . . . 1414
gdcm::VRToType< T > . . . 1415
gdcm::VRVLSize< T > . . . 1415
gdcm::VRVLSize< 0 > . . . 1416
gdcm::VRVLSize< 1 > . . . 1417
vtkImageAlgorithm
vtkImagePlanarComponentsToComponents . . . 1507
vtkImageMapToColors
vtkImageMapToWindowLevelColors2 . . . 1503
vtkImageWriter
vtkGDCMImageWriter . . . 1448

vtkLookupTable
vtkLookupTable16 1514
vtkMedicalImageProperties
vtkGDCMMedicalImageProperties 1456
vtkMedicalImageReader2
vtkGDCMImageReader 1418
vtkGDCMThreadedImageReader 1472
vtkGDCMImageReader2 1433
vtkObject
vtkGDCMTesting 1469
vtkImageColorViewer 1485
vtkRTStructSetProperties 1518
vtkPolyDataAlgorithm
vtkGDCMPolyDataReader 1459
vtkPolyDataWriter
vtkGDCMPolyDataWriter 1464
vtkThreadedImageAlgorithm
vtkGDCMThreadedImageReader2 1478
vtkImageMapToColors16 1497
vtkImageRGBToYBR 1510
vtkImageYBRToRGB 1512
gdcm::Waveform 1528
gdcm::Writer 1533
gdcm::PixmapWriter 953
gdcm::ImageWriter 692
gdcm::SegmentWriter 1084
gdcm::SurfaceWriter 1219
gdcm::XMLPrinter 1542

Chapter 8

Class Index

8.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

gdcm::network::AAbortPDU
AAbortPDU 113
gdcm::network::AAssociateACPDU
AAssociateACPDU 116
gdcm::network::AAssociateRJPDU
AAssociateRJPDU 120
gdcm::network::AAssociateRQPDU
AAssociateRQPDU 122
gdcm::AbortEvent 128
gdcm::network::AbstractSyntax
AbstractSyntax 129
gdcm::AnonymizeEvent
AnonymizeEvent 131
gdcm::Anonymizer
Anonymizer 135
gdcm::AnyEvent 144
gdcm::network::ApplicationContext
ApplicationContext 145
gdcm::ApplicationEntity
ApplicationEntity 147
gdcm::network::AReleaseRPPDU
AReleaseRPPDU 149
gdcm::network::AReleaseRQPDU
AReleaseRQPDU 151
gdcm::network::ARTIMTimer
ARTIMTimer 154
gdcm::ASN1
Class for ASN1 155
gdcm::network::AsynchronousOperationsWindowSub
AsynchronousOperationsWindowSub 157

gdcm::Attribute< Group, Element, TVR, TVM >
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary . . . 158
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 168
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 176
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 181
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 187
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 195
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 202
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 207
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 214
gdcm::AudioCodec
AudioCodec . . . 220
gdcm::Base64
Class for Base64 . . . 222
gdcm::network::BaseCompositeMessage
BaseCompositeMessage . . . 225
gdcm::network::BaseNormalizedMessage
BaseNormalizedMessage . . . 227
gdcm::network::BasePDU
BasePDU . . . 230
gdcm::BaseQuery
BaseQuery . . . 232
gdcm::BaseRootQuery
BaseRootQuery . . . 237
gdcm::SegmentHelper::BasicCodedEntry
This structure defines a basic coded entry with all of its attributes . . . 242
gdcm::BasicOffsetTable
Class to represent a BasicOffsetTable . . . 246
gdcm::Bitmap
Bitmap class . . . 250
gdcm::BitmapToBitmapFilter
BitmapToBitmapFilter class . . . 264
gdcm::BoxRegion
Class for manipulation box region . . . 267
gdcm::ByteBuffer
ByteBuffer . . . 272
gdcm::ByteSwap< T >
ByteSwap . . . 273
gdcm::ByteSwapFilter
ByteSwapFilter . . . 275
gdcm::ByteValue
Class to represent binary value (array of bytes) . . . 277
gdcm::CAPICryptoFactory . . . 285
gdcm::CAPICryptographicMessageSyntax . . . 287
gdcm::network::CEchoRQ
CEchoRQ . . . 290
gdcm::network::CEchoRSP
CEchoRSP this file defines the messages for the cecho action . . . 292
gdcm::network::CFind . . . 293
gdcm::network::CFindCancelRQ
CFindCancelRQ this file defines the messages for the cfind action . . . 294
gdcm::network::CFindRQ
CFindRQ . . . 295

gdcm::network::CFindRSP
CFindRSP this file defines the messages for the cfind action . . . 297
gdcm::Cleaner
Cleaner . . . 298
gdcm::network::CMoveCancelRq . . . 305
gdcm::network::CMoveRQ
CMoveRQ . . . 306
gdcm::network::CMoveRSP
CMoveRSP this file defines the messages for the cmove action . . . 307
gdcm::Codec
Codec class . . . 309
gdcm::Coder
Coder . . . 310
gdcm::CodeString
CodeString . . . 312
gdcm::Command
Command superclass for callback/observer methods . . . 316
gdcm::CommandDataSet
Class to represent a Command DataSet . . . 320
gdcm::network::CompositeMessageFactory
CompositeMessageFactory . . . 324
gdcm::CompositeNetworkFunctions
Composite Network Functions . . . 325
gdcm::ConstCharWrapper
Do not use me . . . 330
gdcm::CP246ExplicitDataElement
Class to read/write a DataElement as CP246Explicit Data Element . . . 331
gdcm::CryptoFactory
Class to do handle the crypto factory . . . 335
gdcm::CryptographicMessageSyntax . . . 337
gdcm::CSAElement
Class to represent a CSA Element . . . 341
gdcm::CSAHeader
Class for CSAHeader . . . 349
gdcm::CSAHeaderDict
Class to represent a map of CSAHeaderDictEntry . . . 354
gdcm::CSAHeaderDictEntry
Class to represent an Entry in the Dict . . . 357
gdcm::CSAHeaderDictException . . . 361
gdcm::network::CStoreRQ
CStoreRQ . . . 361
gdcm::network::CStoreRSP
CStoreRSP this file defines the messages for the cecho action . . . 363
gdcm::Curve
Curve class to handle element 50xx,3000 Curve Data . . . 364
gdcm::DataElement
Class to represent a Data Element either Implicit or Explicit . . . 370
gdcm::DataElementException . . . 384
gdcm::DataEvent
DataEvent . . . 384
gdcm::DataSet
Class to represent a Data Set (which contains Data Elements) . . . 388
gdcm::DataSetEvent
DataSetEvent . . . 401

gdcm::DataSetHelper
DataSetHelper (internal class, not intended for user level) . . . 404
gdcm::Decoder
Decoder . . . 405
gdcm::DefinedTerms
Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor . . . 407
gdcm::Defs
FIXME I do not like the name 'Defs' . . . 408
gdcm::DeltaEncodingCodec
DeltaEncodingCodec compression used by some private vendor . . . 412
gdcm::DICOMDIR
DICOMDIR class . . . 415
gdcm::DICOMDIRGenerator
DICOMDIRGenerator class . . . 416
gdcm::Dict
Class to represent a map of DictEntry . . . 420
gdcm::DictConverter
Class to convert a .dic file into something else: . . . 424
gdcm::DictEntry
Class to represent an Entry in the Dict . . . 428
gdcm::DictPrinter
DictPrinter class . . . 432
gdcm::Dicts
Class to manipulate the sum of knowledge (all the dict user load) . . . 435
gdcm::network::DIMSE
DIMSE . . . 439
gdcm::DirectionCosines
Class to handle DirectionCosines . . . 441
gdcm::Directory
Class for manipulation directories . . . 445
gdcm::DirectoryHelper
DirectoryHelper . . . 449
gdcm::DPath
Class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.google.com/g/comp.protocols.dicom/c/IyIH0IOBMPA` . . . 451
gdcm::DummyValueGenerator
Class for generating dummy value . . . 453
gdcm::Dumper
Codec class . . . 454
gdcm::Element< TVR, TVM >
Element class . . . 457
gdcm::Element< TVR, VM::VM1_2 > . . . 463
gdcm::Element< TVR, VM::VM1_n > . . . 468
gdcm::Element< TVR, VM::VM2_2n > . . . 474
gdcm::Element< TVR, VM::VM2_n > . . . 481

gdcm::Element< TVR, VM::VM3_3n > . . . 486
gdcm::Element< TVR, VM::VM3_4 > . . . 493
gdcm::Element< TVR, VM::VM3_n > . . . 498
gdcm::Element< VR::AS, VM::VM5 > . . . 504
gdcm::Element< VR::OB, VM::VM1 > . . . 508
gdcm::Element< VR::OW, VM::VM1 > . . . 513
gdcm::ElementDisableCombinations< TVR, TVM >
A class which is used to produce compile errors for an invalid combination of template parameters . 518
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 519
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 520
gdcm::EmptyMaskGenerator
EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory . . . 521
gdcm::EncapsulatedDocument
EncapsulatedDocument . . . 523
gdcm::EncodingImplementation< T >
EncodingImplementation . . . 524
gdcm::EncodingImplementation< VR::VRASCII > . . . 525
gdcm::EncodingImplementation< VR::VRBINARY > . . . 527
gdcm::EndEvent . . . 529
gdcm::EnumeratedValues
Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note: . . . 530
gdcm::EquipmentManufacturer . . . 531
gdcm::Event
Superclass for callback/observer methods . . . 533
gdcm::Exception
Exception . . . 536
gdcm::ExitEvent . . . 538
gdcm::ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 539
gdcm::ExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element . . . 543
gdcm::Fiducials
Fiducials . . . 547
gdcm::File
DICOM File . . . 548
gdcm::FileAnonymizer
FileAnonymizer . . . 553
gdcm::FileChangeTransferSyntax
FileChangeTransferSyntax . . . 557
gdcm::FileDecompressLookupTable
FileDecompressLookupTable class . . . 561
gdcm::FileDerivation
FileDerivation class . . . 564
gdcm::FileExplicitFilter
FileExplicitFilter class . . . 568
gdcm::FileMetaInformation
Class to represent a File Meta Information . . . 571
gdcm::Filename
Class to manipulate file name's . . . 581

gdcm::FileNameEvent
FileNameEvent . . . 584
gdcm::FilenameGenerator
FilenameGenerator . . . 588
gdcm::FileSet . . . 591
gdcm::FileStreamer
FileStreamer . . . 593
gdcm::FileWithName
FileWithName . . . 599
gdcm::FindPatientRootQuery
PatientRootQuery . . . 602
gdcm::FindStudyRootQuery
FindStudyRootQuery . . . 606
gdcm::Fragment
Class to represent a Fragment . . . 610
gdcm::Global
Global . . . 615
gdcm::GroupDict
Class to represent the mapping from group number to its abbreviation and name . . . 618
gdcm::IconImageFilter
IconImageFilter . . . 621
gdcm::IconImageGenerator
IconImageGenerator . . . 624
gdcm::ignore_char . . . 627
gdcm::Image
Image . . . 628
gdcm::ImageApplyLookupTable
ImageApplyLookupTable class . . . 637
gdcm::ImageChangePhotometricInterpretation
ImageChangePhotometricInterpretation class . . . 640
gdcm::ImageChangePlanarConfiguration
ImageChangePlanarConfiguration class . . . 645
gdcm::ImageChangeTransferSyntax
ImageChangeTransferSyntax class . . . 649
gdcm::ImageCodec
ImageCodec . . . 655
gdcm::ImageConverter
Image Converter . . . 667
gdcm::ImageFragmentSplitter
ImageFragmentSplitter class . . . 669
gdcm::ImageHelper
ImageHelper (internal class, not intended for user level) . . . 672
gdcm::ImageReader
ImageReader . . . 679
gdcm::ImageRegionReader
ImageRegionReader . . . 684
gdcm::ImageToImageFilter
ImageToImageFilter class . . . 689
gdcm::ImageWriter
ImageWriter . . . 692
gdcm::network::ImplementationClassUIDSub
ImplementationClassUIDSub . . . 696
gdcm::network::ImplementationUIDSub
ImplementationUIDSub . . . 698

gdcm::network::ImplementationVersionNameSub
ImplementationVersionNameSub 698
gdcm::ImplicitDataElement
Class to represent an *Implicit VR* Data Element 700
gdcm::InitializeEvent 704
gdcm::IOD
Class for representing a IOD 705
gdcm::IODEntry
Class for representing a IODEntry 708
gdcm::IODs
Class for representing a IODs 711
gdcm::IPPSorter
IPPSorter 714
gdcm::Item
Class to represent an Item 720
gdcm::IterationEvent 726
gdcm::JPEG12Codec
Class to do JPEG 12bits (lossy & lossless) 727
gdcm::JPEG16Codec
Class to do JPEG 16bits (lossless) 732
gdcm::JPEG2000Codec
Class to do JPEG 2000 737
gdcm::JPEG8Codec
Class to do JPEG 8bits (lossy & lossless) 745
gdcm::JPEGCodec
JPEG codec 750
gdcm::JPEGLSCodec
JPEG-LS 759
gdcm::JSON 766
gdcm::KAKADUCodec
KAKADUCodec 768
gdcm::LO
LO 772
gdcm::LookupTable
LookupTable class 776
gdcm::Scanner2::ltstr 783
gdcm::Scanner::ltstr 784
gdcm::StrictScanner2::ltstr 784
gdcm::StrictScanner::ltstr 785
gdcm::Macro
Class for representing a Macro 785
gdcm::Macros
Class for representing a Modules 788
gdcm::network::MaximumLengthSub
MaximumLengthSub 790
gdcm::MD5
Class for MD5 792
gdcm::MEC_MR3
Class for MEC_MR3 793
gdcm::MediaStorage
MediaStorage 794
gdcm::MemberCommand< T >
Command subclass that calls a pointer to a member function 804

gdcm::MeshPrimitive
This class defines surface mesh primitives . . . 811
gdcm::ModalityPerformedProcedureStepCreateQuery
ModalityPerformedProcedureStepCreateQuery . . . 816
gdcm::ModalityPerformedProcedureStepSetQuery
ModalityPerformedProcedureStepSetQuery . . . 820
gdcm::ModifiedEvent . . . 824
gdcm::Module
Class for representing a Module . . . 825
gdcm::ModuleEntry
Class for representing a ModuleEntry . . . 828
gdcm::Modules
Class for representing a Modules . . . 832
gdcm::MovePatientRootQuery
MovePatientRootQuery . . . 835
gdcm::MoveStudyRootQuery
MoveStudyRootQuery . . . 839
gdcm::MrProtocol
Class for MrProtocol . . . 843
gdcm::network::NActionRQ
NActionRQ . . . 845
gdcm::network::NActionRSP
NActionRSP this file defines the messages for the NAction action . . . 846
gdcm::network::NCreateRQ
NCreateRQ . . . 848
gdcm::network::NCreateRSP
NCreateRSP this file defines the messages for the ncreate action . . . 849
gdcm::network::NDeleteRQ
NDeleteRQ . . . 851
gdcm::network::NDeleteRSP
NDeleteRSP this file defines the messages for the ndelete action . . . 852
gdcm::NestedModuleEntries
Class for representing a NestedModuleEntries . . . 854
gdcm::network::NEventReportRQ
NEventReportRQ . . . 857
gdcm::network::NEventReportRSP
NEventReportRSP this file defines the messages for the neventreport action . . . 859
gdcm::network::NGetRQ
NGetRQ . . . 860
gdcm::network::NGetRSP
NGetRSP this file defines the messages for the nget action . . . 862
gdcm::NoEvent . . . 863
gdcm::network::NormalizedMessageFactory . . . 864
gdcm::NormalizedNetworkFunctions
Normalized Network Functions . . . 865
gdcm::network::NSetRQ
NSetRQ . . . 868
gdcm::network::NSetRSP
NSetRSP this file defines the messages for the nset action . . . 869
gdcm::Object
Object . . . 871
gdcm::OpenSSLCryptoFactory . . . 874
gdcm::OpenSSLCryptographicMessageSyntax . . . 876
gdcm::OpenSSLP7CryptoFactory . . . 880

gdcm::OpenSSLP7CryptographicMessageSyntax . . . 882
gdcm::Orientation
Class to handle Orientation . . . 885
gdcm::Overlay
Overlay class . . . 888
gdcm::ParseException
ParseException Standard exception handling object . . . 898
gdcm::Parser
Parser ala XML_Parser from expat (SAX) . . . 900
gdcm::Patient
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54 . . . 904
gdcm::network::PDataTFPDU
PDataTFPDU . . . 904
gdcm::PDBElement
Class to represent a PDB Element . . . 908
gdcm::PDBHeader
Class for PDBHeader . . . 911
gdcm::PDFCodec
PDFCodec class . . . 913
gdcm::network::PDUFactory
PDUFactory basically, given an initial byte, construct the . . . 916
gdcm::PersonName
PersonName class . . . 919
gdcm::PGXCodec
Class to do PGX . . . 922
gdcm::PhotometricInterpretation
Class to represent an PhotometricInterpretation . . . 926
gdcm::PixelFormat
PixelFormat . . . 930
gdcm::Pixmap
Pixmap class . . . 939
gdcm::PixmapReader
PixmapReader . . . 946
gdcm::PixmapToPixmapFilter
PixmapToPixmapFilter class . . . 950
gdcm::PixmapWriter
PixmapWriter . . . 953
gdcm::PNMCodec
Class to do PNM . . . 958
gdcm::Preamble
DICOM Preamble (Part 10) . . . 963
gdcm::PresentationContext
PresentationContext . . . 967
gdcm::network::PresentationContextAC
PresentationContextAC . . . 970
gdcm::PresentationContextGenerator
PresentationContextGenerator . . . 973
gdcm::network::PresentationContextRQ
PresentationContextRQ . . . 976
gdcm::network::PresentationDataValue
PresentationDataValue . . . 979
gdcm::Printer
Printer class . . . 983

gdcm::PrivateDict
Private Dict 987
gdcm::PrivateTag
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner) 990
gdcm::ProgressEvent
ProgressEvent 995
gdcm::PVRGCodec
PVRGCodec 999
gdcm::PythonFilter
PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language 1003
gdcm::QueryBase
QueryBase 1005
gdcm::QueryFactory
QueryFactory.h 1008
gdcm::QueryImage
QueryImage 1009
gdcm::QueryPatient
QueryPatient 1012
gdcm::QuerySeries
QuerySeries 1014
gdcm::QueryStudy
QueryStudy.h 1017
gdcm::RAWCodec
RAWCodec class 1019
gdcm::Reader
Reader ala DOM (Document Object Model) 1024
gdcm::RealWorldValueMappingContent 1031
gdcm::Region
Class for manipulation region 1032
gdcm::Rescaler
Rescale class 1035
gdcm::RLECodec
Class to do RLE 1039
gdcm::network::RoleSelectionSub
RoleSelectionSub 1046
gdcm::Scanner
Scanner 1048
gdcm::Scanner2
Scanner2 1057
gdcm::Segment
This class defines a segment 1068
gdcm::SegmentedPaletteColorLookupTable
SegmentedPaletteColorLookupTable class 1077
gdcm::SegmentReader
This class defines a segment reader 1080
gdcm::SegmentWriter
This class defines a segment writer 1084
gdcm::SequenceOfFragments
Class to represent a Sequence Of Fragments 1090
gdcm::SequenceOfItems
Class to represent a Sequence Of Items 1097

gdcm::SerieHelper
SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned . . . 1106
gdcm::Series
Series . . . 1111
gdcm::network::ServiceClassApplicationInformation . . . 1112
gdcm::ServiceClassUser
ServiceClassUser . . . 1114
gdcm::SHA1
Class for SHA1 . . . 1122
gdcm::SimpleMemberCommand< T >
Command subclass that calls a pointer to a member function . . . 1124
gdcm::SimpleSubjectWatcher
SimpleSubjectWatcher . . . 1129
gdcm::MrProtocol::Slice . . . 1132
gdcm::MrProtocol::SliceArray . . . 1133
gdcm::SmartPointer< ObjectType >
Class for Smart Pointer . . . 1134
gdcm::network::SOPClassExtendedNegociationSub
SOPClassExtendedNegociationSub . . . 1138
gdcm::SOPClassUIDToIOD
Class convert a class SOP Class UID into IOD . . . 1139
gdcm::Sorter
Sorter . . . 1141
gdcm::Spacing
Class for Spacing . . . 1146
gdcm::Spectroscopy
Spectroscopy class . . . 1148
gdcm::SplitMosaicFilter
SplitMosaicFilter class . . . 1149
gdcm::StartEvent . . . 1152
gdcm::static_assert_test< x > . . . 1153
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 1154
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 1154
gdcm::StreamImageReader
StreamImageReader . . . 1155
gdcm::StreamImageWriter
StreamImageReader . . . 1159
gdcm::StrictScanner
StrictScanner . . . 1166
gdcm::StrictScanner2
StrictScanner2 . . . 1175
gdcm::String< TDelimiter, TMaxLength, TPadChar >
String . . . 1186
gdcm::StringFilter
StringFilter . . . 1190
gdcm::Study
Study . . . 1194
gdcm::Subject
Subject . . . 1195
gdcm::Surface
This class defines a SURFACE IE . . . 1198
gdcm::SurfaceHelper
SurfaceHelper . . . 1211

gdcm::SurfaceReader
This class defines a SURFACE IE reader . . . 1214
gdcm::SurfaceWriter
This class defines a SURFACE IE writer . . . 1219
gdcm::SwapCode
SwapCode representation . . . 1224
gdcm::SwapperDoOp . . . 1227
gdcm::SwapperNoOp . . . 1228
gdcm::System
Class to do system operation . . . 1228
gdcm::Table
Table . . . 1235
gdcm::TableEntry
TableEntry . . . 1238
gdcm::TableReader
Class for representing a TableReader . . . 1239
gdcm::network::TableRow . . . 1243
gdcm::Tag
Class to represent a DICOM Data Element (Attribute) Tag (Group, Element) . . . 1244
gdcm::TagPath
Class to handle a path of tag . . . 1255
gdcm::Testing
Class for testing . . . 1257
gdcm::Trace
Trace . . . 1263
gdcm::TransferSyntax
Class to manipulate Transfer Syntax . . . 1268
gdcm::network::TransferSyntaxSub
TransferSyntaxSub . . . 1274
gdcm::network::Transition . . . 1276
gdcm::Type
Type . . . 1278
gdcm::UI . . . 1281
gdcm::UIDGenerator
Class for generating unique UID . . . 1281
gdcm::UIDs
All known uids . . . 1284
gdcm::network::ULAction
ULAction . . . 1302
gdcm::network::ULActionAA1 . . . 1306
gdcm::network::ULActionAA2 . . . 1307
gdcm::network::ULActionAA3 . . . 1308
gdcm::network::ULActionAA4 . . . 1310
gdcm::network::ULActionAA5 . . . 1311
gdcm::network::ULActionAA6 . . . 1312
gdcm::network::ULActionAA7 . . . 1314
gdcm::network::ULActionAA8 . . . 1315
gdcm::network::ULActionAE1 . . . 1316
gdcm::network::ULActionAE2 . . . 1318
gdcm::network::ULActionAE3 . . . 1319
gdcm::network::ULActionAE4 . . . 1320
gdcm::network::ULActionAE5 . . . 1322
gdcm::network::ULActionAE6 . . . 1323
gdcm::network::ULActionAE7 . . . 1324

gdcm::network::ULActionAE8 . . . 1326
gdcm::network::ULActionAR1 . . . 1327
gdcm::network::ULActionAR10 . . . 1328
gdcm::network::ULActionAR2 . . . 1330
gdcm::network::ULActionAR3 . . . 1331
gdcm::network::ULActionAR4 . . . 1332
gdcm::network::ULActionAR5 . . . 1334
gdcm::network::ULActionAR6 . . . 1335
gdcm::network::ULActionAR7 . . . 1336
gdcm::network::ULActionAR8 . . . 1338
gdcm::network::ULActionAR9 . . . 1339
gdcm::network::ULActionDT1 . . . 1340
gdcm::network::ULActionDT2 . . . 1342
gdcm::network::ULBasicCallback
ULBasicCallback . . . 1343
gdcm::network::ULConnection
ULConnection . . . 1346
gdcm::network::ULConnectionCallback . . . 1351
gdcm::network::ULConnectionInfo
ULConnectionInfo . . . 1353
gdcm::network::ULConnectionManager
ULConnectionManager . . . 1355
gdcm::network::ULEvent
ULEvent . . . 1363
gdcm::network::ULTransitionTable
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates . . . 1364
gdcm::network::ULWritingCallback . . . 1366
gdcm::UNExplicitDataElement
Class to read/write a DataElement as UNExplicit Data Element . . . 1368
gdcm::UNExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element . . . 1372
gdcm::Unpacker12Bits
Pack/Unpack 12 bits pixel into 16bits . . . 1376
gdcm::Usage
Usage . . . 1377
gdcm::UserEvent . . . 1380
gdcm::network::UserInformation
UserInformation . . . 1381
gdcm::UUIDGenerator
Class for generating unique UUID . . . 1383
gdcm::Validate
Validate class . . . 1384
gdcm::Value
Class to represent the value of a Data Element . . . 1387
gdcm::ValueIO< TDE, TSwap, TType >
Class to dispatch template calls . . . 1390
gdcm::MrProtocol::Vector3 . . . 1391
gdcm::Version
Major/minor and build version . . . 1392
gdcm::VL
Value Length . . . 1394
gdcm::VM
Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n . . . 1398

gdcm::VMToLength< T > . . . 1403
gdcm::VR
VR class . . . 1403
gdcm::VR16ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 1411
gdcm::VRToEncoding< T > . . . 1414
gdcm::VRToType< T > . . . 1415
gdcm::VRVLSize< T > . . . 1415
gdcm::VRVLSize< 0 > . . . 1416
gdcm::VRVLSize< 1 > . . . 1417
vtkGDCMImageReader . . . 1418
vtkGDCMImageReader2 . . . 1433
vtkGDCMImageWriter . . . 1448
vtkGDCMMedicalImageProperties . . . 1456
vtkGDCMPolyDataReader . . . 1459
vtkGDCMPolyDataWriter . . . 1464
vtkGDCMTesting . . . 1469
vtkGDCMThreadedImageReader . . . 1472
vtkGDCMThreadedImageReader2 . . . 1478
vtkImageColorViewer . . . 1485
vtkImageMapToColors16 . . . 1497
vtkImageMapToWindowLevelColors2 . . . 1503
vtkImagePlanarComponentsToComponents . . . 1507
vtkImageRGBToYBR . . . 1510
vtkImageYBRToRGB . . . 1512
vtkLookupTable16 . . . 1514
vtkRTStructSetProperties . . . 1518
gdcm::Waveform
Waveform class . . . 1528
gdcm::WLMFindQuery
PatientRootQuery . . . 1529
gdcm::Writer
Writer ala DOM (Document Object Model) . . . 1533
gdcm::XMLDictReader
Class for representing a XMLDictReader . . . 1539
gdcm::XMLPrinter . . . 1542
gdcm::XMLPrivateDictReader
Class for representing a XMLPrivateDictReader . . . 1545

Chapter 9

File Index

9.1 File List

Here is a list of all files with brief descriptions:

gdcmASN1.h . . . 1549
gdcmBase64.h . . . 1551
gdcmBoxRegion.h . . . 1552
gdcmByteSwap.h . . . 1553
gdcmCAPICryptoFactory.h . . . 1555
gdcmCAPICryptographicMessageSyntax.h . . . 1556
gdcmCommand.h . . . 1559
gdcmCryptoFactory.h . . . 1562
gdcmCryptographicMessageSyntax.h . . . 1564
gdcmDataEvent.h . . . 1566
gdcmDeflateStream.h . . . 1568
gdcmDirectory.h . . . 1568
gdcmDummyValueGenerator.h . . . 1571
gdcmEvent.h . . . 1572
gdcmException.h . . . 1575
gdcmFilename.h . . . 1577
gdcmFileNameEvent.h . . . 1578
gdcmFilenameGenerator.h . . . 1580
gdcmLegacyMacro.h . . . 1581
gdcmMD5.h . . . 1584
gdcmObject.h . . . 1585
gdcmOpenSSLCryptoFactory.h . . . 1588
gdcmOpenSSLCryptographicMessageSyntax.h . . . 1589
gdcmOpenSSLP7CryptoFactory.h . . . 1591
gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1593
gdcmProgressEvent.h . . . 1595
gdcmRegion.h . . . 1596
gdcmSHA1.h . . . 1599
gdcmSmartPointer.h . . . 1600
gdcmStaticAssert.h . . . 1602
gdcmString.h . . . 1604

gdcmSubject.h . . . 1608
gdcmSwapCode.h . . . 1609
gdcmSwapper.h . . . 1611
gdcmSystem.h . . . 1614
gdcmTerminal.h . . . 1616
gdcmTestDriver.h . . . 1618
gdcmTesting.h . . . 1619
gdcmTrace.h . . . 1621
gdcmTypes.h . . . 1627
gdcmUnpacker12Bits.h . . . 1628
gdcmVersion.h . . . 1629
gdcmWin32.h . . . 1630
gdcmCSAHeaderDict.h . . . 1632
gdcmCSAHeaderDictEntry.h . . . 1635
gdcmDict.h . . . 1638
gdcmDictConverter.h . . . 1643
gdcmDictEntry.h . . . 1645
gdcmDicts.h . . . 1648
gdcmGlobal.h . . . 1650
gdcmGroupDict.h . . . 1652
gdcmSOPClassUIDToIOD.h . . . 1654
gdcmUIDs.h . . . 1655
gdcmAttribute.h . . . 1669
gdcmBasicOffsetTable.h . . . 1683
gdcmByteBuffer.h . . . 1686
gdcmByteSwapFilter.h . . . 1689
gdcmByteValue.h . . . 1690
gdcmCodeString.h . . . 1694
gdcmCP246ExplicitDataElement.h . . . 1696
gdcmCSAElement.h . . . 1697
gdcmCSAHeader.h . . . 1701
gdcmDataElement.h . . . 1703
gdcmDataSet.h . . . 1707
gdcmDataSetEvent.h . . . 1711
gdcmElement.h . . . 1713
gdcmExplicitDataElement.h . . . 1725
gdcmExplicitImplicitDataElement.h . . . 1727
gdcmFile.h . . . 1729
gdcmFileMetaInformation.h . . . 1730
gdcmFileSet.h . . . 1733
gdcmFragment.h . . . 1735
gdcmImplicitDataElement.h . . . 1740
gdcmItem.h . . . 1741
gdcmLO.h . . . 1747
gdcmMediaStorage.h . . . 1748
gdcmMrProtocol.h . . . 1752
gdcmParseException.h . . . 1754
gdcmParser.h . . . 1756
gdcmPDBElement.h . . . 1759
gdcmPDBHeader.h . . . 1761
gdcmPreamble.h . . . 1763
gdcmPrivateTag.h . . . 1765
gdcmReader.h . . . 1767
gdcmSequenceOfFragments.h . . . 1769

gdcmSequenceOfItems.h . . . 1774
gdcmTag.h . . . 1778
gdcmTagToVR.h . . . 1783
gdcmTransferSyntax.h . . . 1784
gdcmUNExplicitDataElement.h . . . 1786
gdcmUNExplicitImplicitDataElement.h . . . 1788
gdcmValue.h . . . 1789
gdcmValueIO.h . . . 1791
gdcmVL.h . . . 1792
gdcmVM.h . . . 1795
gdcmVR.h . . . 1798
gdcmVR16ExplicitDataElement.h . . . 1805
gdcmWriter.h . . . 1807
gdcmDefinedTerms.h . . . 1809
gdcmDefs.h . . . 1810
gdcmEnumeratedValues.h . . . 1813
gdcmIOD.h . . . 1814
gdcmIODEntry.h . . . 1817
gdcmIODs.h . . . 1819
gdcmMacro.h . . . 1822
gdcmMacroEntry.h . . . 1825
gdcmMacros.h . . . 1828
gdcmModule.h . . . 1830
gdcmModuleEntry.h . . . 1833
gdcmModules.h . . . 1836
gdcmNestedModuleEntries.h . . . 1838
gdcmPatient.h . . . 1840
gdcmSeries.h . . . 1842
gdcmStudy.h . . . 1843
gdcmTable.h . . . 1845
gdcmTableEntry.h . . . 1847
gdcmTableReader.h . . . 1849
gdcmType.h . . . 1852
gdcmUsage.h . . . 1854
gdcmXMLDictReader.h . . . 1857
gdcmXMLPrivateDictReader.h . . . 1859
gdcmAnonymizeEvent.h . . . 1860
gdcmAnonymizer.h . . . 1862
gdcmApplicationEntity.h . . . 1864
gdcmAudioCodec.h . . . 1866
gdcmBitmap.h . . . 1867
gdcmBitmapToBitmapFilter.h . . . 1871
gdcmCleaner.h . . . 1872
gdcmCodec.h . . . 1874
gdcmCoder.h . . . 1876
gdcmConstCharWrapper.h . . . 1877
gdcmCurve.h . . . 1878
gdcmDataSetHelper.h . . . 1881
gdcmDecoder.h . . . 1882
gdcmDeltaEncodingCodec.h . . . 1884
gdcmDICOMDIR.h . . . 1885
gdcmDICOMDIRGenerator.h . . . 1886
gdcmDictPrinter.h . . . 1888
gdcmDirectionCosines.h . . . 1889

gdcmDirectoryHelper.h . . . 1891
gdcmDPath.h . . . 1892
gdcmDumper.h . . . 1894
gdcmEmptyMaskGenerator.h . . . 1896
gdcmEncapsulatedDocument.h . . . 1897
gdcmEquipmentManufacturer.h . . . 1898
gdcmFiducials.h . . . 1900
gdcmFileAnonymizer.h . . . 1901
gdcmFileChangeTransferSyntax.h . . . 1902
gdcmFileDecompressLookupTable.h . . . 1904
gdcmFileDerivation.h . . . 1906
gdcmFileExplicitFilter.h . . . 1908
gdcmFileStreamer.h . . . 1909
gdcmIconImage.h . . . 1911
gdcmIconImageFilter.h . . . 1913
gdcmIconImageGenerator.h . . . 1915
gdcmImage.h . . . 1916
gdcmImageApplyLookupTable.h . . . 1919
gdcmImageChangePhotometricInterpretation.h . . . 1920
gdcmImageChangePlanarConfiguration.h . . . 1923
gdcmImageChangeTransferSyntax.h . . . 1924
gdcmImageCodec.h . . . 1926
gdcmImageConverter.h . . . 1929
gdcmImageFragmentSplitter.h . . . 1931
gdcmImageHelper.h . . . 1932
gdcmImageReader.h . . . 1934
gdcmImageRegionReader.h . . . 1936
gdcmImageToImageFilter.h . . . 1938
gdcmImageWriter.h . . . 1939
gdcmIPPSorter.h . . . 1941
gdcmJPEG12Codec.h . . . 1943
gdcmJPEG16Codec.h . . . 1944
gdcmJPEG2000Codec.h . . . 1946
gdcmJPEG8Codec.h . . . 1948
gdcmJPEGCodec.h . . . 1949
gdcmJPEGLSCodec.h . . . 1952
gdcmJSON.h . . . 1953
gdcmKAKADUCodec.h . . . 1955
gdcmLookupTable.h . . . 1956
gdcmMEC_MR3.h . . . 1959
gdcmMeshPrimitive.h . . . 1960
gdcmOrientation.h . . . 1963
gdcmOverlay.h . . . 1964
gdcmPDFCodec.h . . . 1967
gdcmPersonName.h . . . 1968
gdcmPGXCodec.h . . . 1970
gdcmPhotometricInterpretation.h . . . 1971
gdcmPixelFormat.h . . . 1973
gdcmPixmap.h . . . 1977
gdcmPixmapReader.h . . . 1980
gdcmPixmapToPixmapFilter.h . . . 1982
gdcmPixmapWriter.h . . . 1983
gdcmPNMCodec.h . . . 1985
gdcmPrinter.h . . . 1986

gdcmPVRGCodec.h . . . 1989
gdcmRAWCodec.h . . . 1991
gdcmRescaler.h . . . 1992
gdcmRLECodec.h . . . 1994
gdcmScanner.h . . . 1995
gdcmScanner2.h . . . 1998
gdcmSegment.h . . . 2001
gdcmSegmentedPaletteColorLookupTable.h . . . 2005
gdcmSegmentHelper.h . . . 2006
gdcmSegmentReader.h . . . 2008
gdcmSegmentWriter.h . . . 2010
gdcmSerieHelper.h . . . 2012
gdcmSimpleSubjectWatcher.h . . . 2015
gdcmSorter.h . . . 2017
gdcmSpacing.h . . . 2020
gdcmSpectroscopy.h . . . 2021
gdcmSplitMosaicFilter.h . . . 2022
gdcmStreamImageReader.h . . . 2025
gdcmStreamImageWriter.h . . . 2026
gdcmStrictScanner.h . . . 2028
gdcmStrictScanner2.h . . . 2031
gdcmStringFilter.h . . . 2034
gdcmSurface.h . . . 2036
gdcmSurfaceHelper.h . . . 2040
gdcmSurfaceReader.h . . . 2043
gdcmSurfaceWriter.h . . . 2045
gdcmTagPath.h . . . 2046
gdcmUIDGenerator.h . . . 2048
gdcmUUIDGenerator.h . . . 2050
gdcmValidate.h . . . 2051
gdcmWaveform.h . . . 2052
gdcmXMLPrinter.h . . . 2053
gdcmAAbortPDU.h . . . 2056
gdcmAAssociateACPDU.h . . . 2057
gdcmAAssociateRJPDU.h . . . 2060
gdcmAAssociateRQPDU.h . . . 2061
gdcmAbstractSyntax.h . . . 2064
gdcmApplicationContext.h . . . 2066
gdcmAReleaseRPPDU.h . . . 2067
gdcmAReleaseRQPDU.h . . . 2069
gdcmARTIMTimer.h . . . 2070
gdcmAsynchronousOperationsWindowSub.h . . . 2072
gdcmBaseCompositeMessage.h . . . 2073
gdcmBaseNormalizedMessage.h . . . 2075
gdcmBasePDU.h . . . 2076
gdcmBaseQuery.h . . . 2078
gdcmBaseRootQuery.h . . . 2080
gdcmCEchoMessages.h . . . 2083
gdcmCFindMessages.h . . . 2084
gdcmCMoveMessages.h . . . 2085
gdcmCommandDataSet.h . . . 2087
gdcmCompositeMessageFactory.h . . . 2089
gdcmCompositeNetworkFunctions.h . . . 2090
gdcmCStoreMessages.h . . . 2092

gdcmDIMSE.h . . . 2093
gdcmFindPatientRootQuery.h . . . 2095
gdcmFindStudyRootQuery.h . . . 2097
gdcmImplementationClassUIDSub.h . . . 2098
gdcmImplementationUIDSub.h . . . 2100
gdcmImplementationVersionNameSub.h . . . 2101
gdcmMaximumLengthSub.h . . . 2103
gdcmModalityPerformedProcedureStepCreateQuery.h . . . 2105
gdcmModalityPerformedProcedureStepSetQuery.h . . . 2106
gdcmMovePatientRootQuery.h . . . 2107
gdcmMoveStudyRootQuery.h . . . 2109
gdcmNActionMessages.h . . . 2110
gdcmNCreateMessages.h . . . 2111
gdcmNDeleteMessages.h . . . 2113
gdcmNetworkEvents.h . . . 2114
gdcmNetworkStateID.h . . . 2116
gdcmNEventReportMessages.h . . . 2118
gdcmNGetMessages.h . . . 2119
gdcmNormalizedMessageFactory.h . . . 2120
gdcmNormalizedNetworkFunctions.h . . . 2122
gdcmNSetMessages.h . . . 2124
gdcmPDataTFPDU.h . . . 2125
gdcmPDUFactory.h . . . 2127
gdcmPresentationContext.h . . . 2128
gdcmPresentationContextAC.h . . . 2130
gdcmPresentationContextGenerator.h . . . 2132
gdcmPresentationContextRQ.h . . . 2134
gdcmPresentationDataValue.h . . . 2136
gdcmQueryBase.h . . . 2138
gdcmQueryFactory.h . . . 2141
gdcmQueryImage.h . . . 2142
gdcmQueryPatient.h . . . 2144
gdcmQuerySeries.h . . . 2146
gdcmQueryStudy.h . . . 2147
gdcmRoleSelectionSub.h . . . 2149
gdcmServiceClassApplicationInformation.h . . . 2150
gdcmServiceClassUser.h . . . 2152
gdcmSOPClassExtendedNegociationSub.h . . . 2154
gdcmTransferSyntaxSub.h . . . 2155
gdcmULAction.h . . . 2157
gdcmULActionAA.h . . . 2159
gdcmULActionAE.h . . . 2161
gdcmULActionAR.h . . . 2163
gdcmULActionDT.h . . . 2166
gdcmULBasicCallback.h . . . 2167
gdcmULConnection.h . . . 2168
gdcmULConnectionCallback.h . . . 2171
gdcmULConnectionInfo.h . . . 2172
gdcmULConnectionManager.h . . . 2174
gdcmULEvent.h . . . 2177
gdcmULTransitionTable.h . . . 2179
gdcmULWritingCallback.h . . . 2182
gdcmUserInformation.h . . . 2183
gdcmWLMFindQuery.h . . . 2185

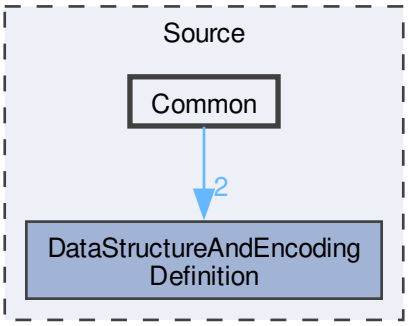
vtkGDCMImageReader.h 2186
vtkGDCMImageReader2.h 2192
vtkGDCMImageWriter.h 2197
vtkGDCMMedicalImageProperties.h 2200
vtkGDCMPolyDataReader.h 2206
vtkGDCMPolyDataWriter.h 2207
vtkGDCMTesting.h 2209
vtkGDCMThreadedImageReader.h 2211
vtkGDCMThreadedImageReader2.h 2213
vtkImageColorViewer.h 2215
vtkImageMapToColors16.h 2219
vtkImageMapToWindowLevelColors2.h 2221
vtkImagePlanarComponentsToComponents.h 2223
vtkImageRGBToYBR.h 2224
vtkImageYBRToRGB.h 2226
vtkLookupTable16.h 2227
vtkRTStructSetProperties.h 2229
gdcmPythonFilter.h 2231

Chapter 10

Directory Documentation

10.1 Common Directory Reference

Directory dependency graph for Common:



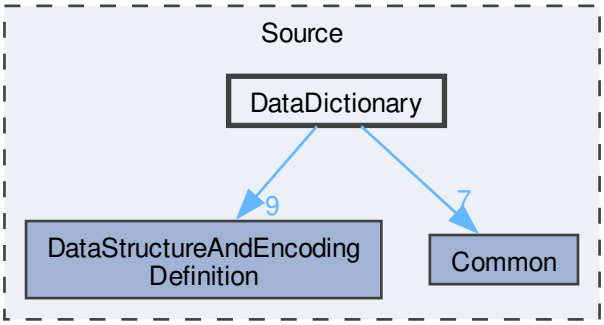
Files

- file gdcmASN1.h
- file gdcmBase64.h
- file gdcmBoxRegion.h
- file gdcmByteSwap.h
- file gdcmCAPICryptoFactory.h
- file gdcmCAPICryptographicMessageSyntax.h
- file gdcmCommand.h
- file gdcmCryptoFactory.h

- file gdcmCryptographicMessageSyntax.h
- file gdcmDataEvent.h
- file gdcmDeflateStream.h
- file gdcmDirectory.h
- file gdcmDummyValueGenerator.h
- file gdcmEvent.h
- file gdcmException.h
- file gdcmFilename.h
- file gdcmFileNameEvent.h
- file gdcmFilenameGenerator.h
- file gdcmLegacyMacro.h
- file gdcmMD5.h
- file gdcmObject.h
- file gdcmOpenSSLCryptoFactory.h
- file gdcmOpenSSLCryptographicMessageSyntax.h
- file gdcmOpenSSLP7CryptoFactory.h
- file gdcmOpenSSLP7CryptographicMessageSyntax.h
- file gdcmProgressEvent.h
- file gdcmRegion.h
- file gdcmSHA1.h
- file gdcmSmartPointer.h
- file gdcmStaticAssert.h
- file gdcmString.h
- file gdcmSubject.h
- file gdcmSwapCode.h
- file gdcmSwapper.h
- file gdcmSystem.h
- file gdcmTerminal.h
- file gdcmTestDriver.h
- file gdcmTesting.h
- file gdcmTrace.h
- file gdcmTypes.h
- file gdcmUnpacker12Bits.h
- file gdcmVersion.h
- file gdcmWin32.h

10.2 DataDictionary Directory Reference

Directory dependency graph for DataDictionary:

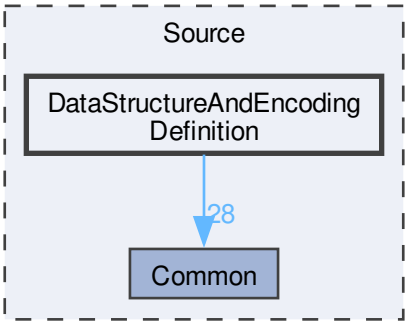


Files

- file gdcmCSAHeaderDict.h
- file gdcmCSAHeaderDictEntry.h
- file gdcmDict.h
- file gdcmDictConverter.h
- file gdcmDictEntry.h
- file gdcmDicts.h
- file gdcmGlobal.h
- file gdcmGroupDict.h
- file gdcmSOPClassUIDToIOD.h
- file gdcmUIDs.h

10.3 DataStructureAndEncodingDefinition Directory Reference

Directory dependency graph for DataStructureAndEncodingDefinition:



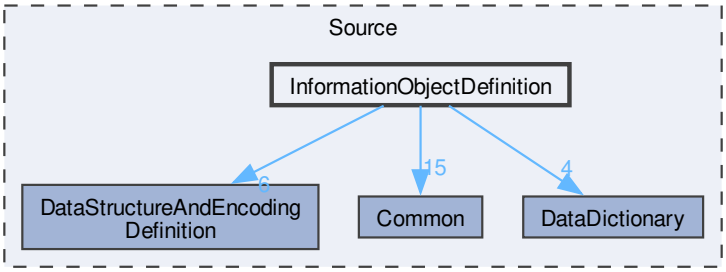
Files

- file gdcmAttribute.h
- file gdcmBasicOffsetTable.h
- file gdcmByteBuffer.h
- file gdcmByteSwapFilter.h
- file gdcmByteValue.h
- file gdcmCodeString.h
- file gdcmCP246ExplicitDataElement.h
- file gdcmCSAElement.h
- file gdcmCSAHeader.h
- file gdcmDataElement.h
- file gdcmDataSet.h
- file gdcmDataSetEvent.h
- file gdcmElement.h
- file gdcmExplicitDataElement.h
- file gdcmExplicitImplicitDataElement.h
- file gdcmFile.h
- file gdcmFileMetaInformation.h
- file gdcmFileSet.h
- file gdcmFragment.h
- file gdcmImplicitDataElement.h
- file gdcmItem.h
- file gdcmLO.h
- file gdcmMediaStorage.h
- file gdcmMrProtocol.h

- file gdcmParseException.h
- file gdcmParser.h
- file gdcmPDBElement.h
- file gdcmPDBHeader.h
- file gdcmPreamble.h
- file gdcmPrivateTag.h
- file gdcmReader.h
- file gdcmSequenceOfFragments.h
- file gdcmSequenceOfItems.h
- file gdcmTag.h
- file gdcmTagToVR.h
- file gdcmTransferSyntax.h
- file gdcmUNExplicitDataElement.h
- file gdcmUNExplicitImplicitDataElement.h
- file gdcmValue.h
- file gdcmValueIO.h
- file gdcmVL.h
- file gdcmVM.h
- file gdcmVR.h
- file gdcmVR16ExplicitDataElement.h
- file gdcmWriter.h

10.4 InformationObjectDefinition Directory Reference

Directory dependency graph for InformationObjectDefinition:



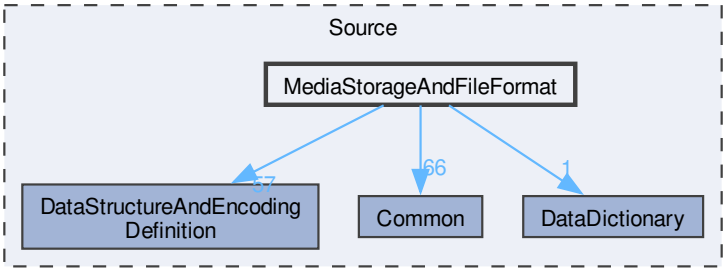
Files

- file gdcmDefinedTerms.h
- file gdcmDefs.h
- file gdcmEnumeratedValues.h
- file gdcmIOD.h

- file gdcmIODEntry.h
- file gdcmIODs.h
- file gdcmMacro.h
- file gdcmMacroEntry.h
- file gdcmMacros.h
- file gdcmModule.h
- file gdcmModuleEntry.h
- file gdcmModules.h
- file gdcmNestedModuleEntries.h
- file gdcmPatient.h
- file gdcmSeries.h
- file gdcmStudy.h
- file gdcmTable.h
- file gdcmTableEntry.h
- file gdcmTableReader.h
- file gdcmType.h
- file gdcmUsage.h
- file gdcmXMLDictReader.h
- file gdcmXMLPrivateDictReader.h

10.5 MediaStorageAndFileFormat Directory Reference

Directory dependency graph for MediaStorageAndFileFormat:



Files

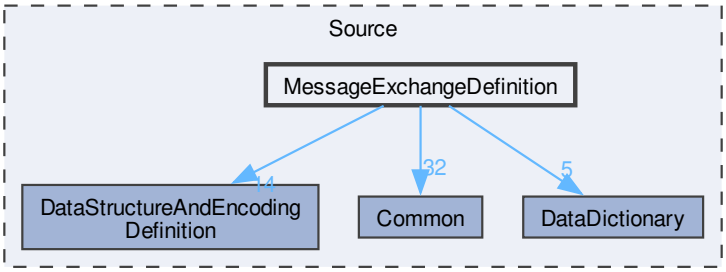
- file gdcmAnonymizeEvent.h
- file gdcmAnonymizer.h
- file gdcmApplicationEntity.h
- file gdcmAudioCodec.h
- file gdcmBitmap.h
- file gdcmBitmapToBitmapFilter.h

- file gdcmCleaner.h
- file gdcmCodec.h
- file gdcmCoder.h
- file gdcmConstCharWrapper.h
- file gdcmCurve.h
- file gdcmDataSetHelper.h
- file gdcmDecoder.h
- file gdcmDeltaEncodingCodec.h
- file gdcmDICOMDIR.h
- file gdcmDICOMDIRGenerator.h
- file gdcmDictPrinter.h
- file gdcmDirectionCosines.h
- file gdcmDirectoryHelper.h
- file gdcmDPath.h
- file gdcmDumper.h
- file gdcmEmptyMaskGenerator.h
- file gdcmEncapsulatedDocument.h
- file gdcmEquipmentManufacturer.h
- file gdcmFiducials.h
- file gdcmFileAnonymizer.h
- file gdcmFileChangeTransferSyntax.h
- file gdcmFileDecompressLookupTable.h
- file gdcmFileDerivation.h
- file gdcmFileExplicitFilter.h
- file gdcmFileStreamer.h
- file gdcmIconImage.h
- file gdcmIconImageFilter.h
- file gdcmIconImageGenerator.h
- file gdcmImage.h
- file gdcmImageApplyLookupTable.h
- file gdcmImageChangePhotometricInterpretation.h
- file gdcmImageChangePlanarConfiguration.h
- file gdcmImageChangeTransferSyntax.h
- file gdcmImageCodec.h
- file gdcmImageConverter.h
- file gdcmImageFragmentSplitter.h
- file gdcmImageHelper.h
- file gdcmImageReader.h
- file gdcmImageRegionReader.h
- file gdcmImageToImageFilter.h
- file gdcmImageWriter.h
- file gdcmIPPSorter.h
- file gdcmJPEG12Codec.h
- file gdcmJPEG16Codec.h
- file gdcmJPEG2000Codec.h
- file gdcmJPEG8Codec.h
- file gdcmJPEGCodec.h
- file gdcmJPEGLSCodec.h
- file gdcmJSON.h
- file gdcmKAKADUCodec.h
- file gdcmLookupTable.h

- file gdcmMEC_MR3.h
- file gdcmMeshPrimitive.h
- file gdcmOrientation.h
- file gdcmOverlay.h
- file gdcmPDFCodec.h
- file gdcmPersonName.h
- file gdcmPGXCodec.h
- file gdcmPhotometricInterpretation.h
- file gdcmPixelFormat.h
- file gdcmPixmap.h
- file gdcmPixmapReader.h
- file gdcmPixmapToPixmapFilter.h
- file gdcmPixmapWriter.h
- file gdcmPNMCodec.h
- file gdcmPrinter.h
- file gdcmPVRGCodec.h
- file gdcmRAWCodec.h
- file gdcmRescaler.h
- file gdcmRLECodec.h
- file gdcmScanner.h
- file gdcmScanner2.h
- file gdcmSegment.h
- file gdcmSegmentedPaletteColorLookupTable.h
- file gdcmSegmentHelper.h
- file gdcmSegmentReader.h
- file gdcmSegmentWriter.h
- file gdcmSerieHelper.h
- file gdcmSimpleSubjectWatcher.h
- file gdcmSorter.h
- file gdcmSpacing.h
- file gdcmSpectroscopy.h
- file gdcmSplitMosaicFilter.h
- file gdcmStreamImageReader.h
- file gdcmStreamImageWriter.h
- file gdcmStrictScanner.h
- file gdcmStrictScanner2.h
- file gdcmStringFilter.h
- file gdcmSurface.h
- file gdcmSurfaceHelper.h
- file gdcmSurfaceReader.h
- file gdcmSurfaceWriter.h
- file gdcmTagPath.h
- file gdcmUIDGenerator.h
- file gdcmUUIDGenerator.h
- file gdcmValidate.h
- file gdcmWaveform.h
- file gdcmXMLPrinter.h

10.6 MessageExchangeDefinition Directory Reference

Directory dependency graph for MessageExchangeDefinition:



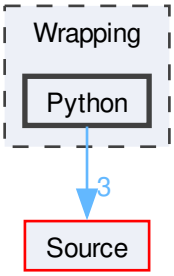
Files

- file gdcmAAbortPDU.h
- file gdcmAAssociateACPDU.h
- file gdcmAAssociateRJPDU.h
- file gdcmAAssociateRQPDU.h
- file gdcmAbstractSyntax.h
- file gdcmApplicationContext.h
- file gdcmAReleaseRPPDU.h
- file gdcmAReleaseRQPDU.h
- file gdcmARTIMTimer.h
- file gdcmAsynchronousOperationsWindowSub.h
- file gdcmBaseCompositeMessage.h
- file gdcmBaseNormalizedMessage.h
- file gdcmBasePDU.h
- file gdcmBaseQuery.h
- file gdcmBaseRootQuery.h
- file gdcmCEchoMessages.h
- file gdcmCFindMessages.h
- file gdcmCMoveMessages.h
- file gdcmCommandDataSet.h
- file gdcmCompositeMessageFactory.h
- file gdcmCompositeNetworkFunctions.h
- file gdcmCStoreMessages.h
- file gdcmDIMSE.h
- file gdcmFindPatientRootQuery.h
- file gdcmFindStudyRootQuery.h
- file gdcmImplementationClassUIDSub.h
- file gdcmImplementationUIDSub.h

- file gdcmImplementationVersionNameSub.h
- file gdcmMaximumLengthSub.h
- file gdcmModalityPerformedProcedureStepCreateQuery.h
- file gdcmModalityPerformedProcedureStepSetQuery.h
- file gdcmMovePatientRootQuery.h
- file gdcmMoveStudyRootQuery.h
- file gdcmNActionMessages.h
- file gdcmNCreateMessages.h
- file gdcmNDeleteMessages.h
- file gdcmNetworkEvents.h
- file gdcmNetworkStateID.h
- file gdcmNEventReportMessages.h
- file gdcmNGetMessages.h
- file gdcmNormalizedMessageFactory.h
- file gdcmNormalizedNetworkFunctions.h
- file gdcmNSetMessages.h
- file gdcmPDataTFPDU.h
- file gdcmPDUFactory.h
- file gdcmPresentationContext.h
- file gdcmPresentationContextAC.h
- file gdcmPresentationContextGenerator.h
- file gdcmPresentationContextRQ.h
- file gdcmPresentationDataValue.h
- file gdcmQueryBase.h
- file gdcmQueryFactory.h
- file gdcmQueryImage.h
- file gdcmQueryPatient.h
- file gdcmQuerySeries.h
- file gdcmQueryStudy.h
- file gdcmRoleSelectionSub.h
- file gdcmServiceClassApplicationInformation.h
- file gdcmServiceClassUser.h
- file gdcmSOPClassExtendedNegociationSub.h
- file gdcmTransferSyntaxSub.h
- file gdcmULAction.h
- file gdcmULActionAA.h
- file gdcmULActionAE.h
- file gdcmULActionAR.h
- file gdcmULActionDT.h
- file gdcmULBasicCallback.h
- file gdcmULConnection.h
- file gdcmULConnectionCallback.h
- file gdcmULConnectionInfo.h
- file gdcmULConnectionManager.h
- file gdcmULEvent.h
- file gdcmULTransitionTable.h
- file gdcmULWritingCallback.h
- file gdcmUserInformation.h
- file gdcmWLMFindQuery.h

10.7 Python Directory Reference

Directory dependency graph for Python:

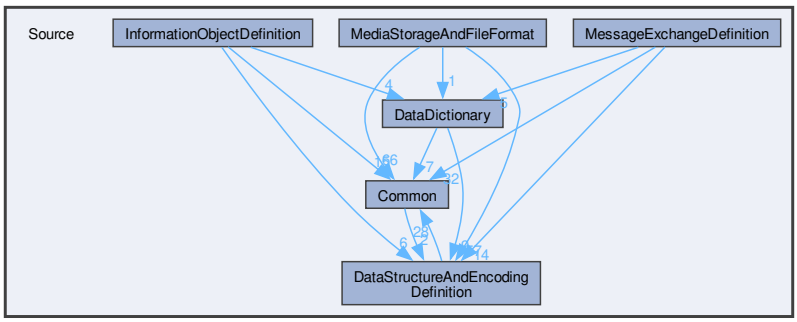


Files

- file gdcmPythonFilter.h

10.8 Source Directory Reference

Directory dependency graph for Source:



Directories

- directory Common
- directory DataDictionary
- directory DataStructureAndEncodingDefinition
- directory InformationObjectDefinition
- directory MediaStorageAndFileFormat
- directory MessageExchangeDefinition

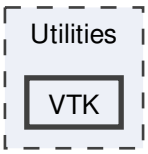
10.9 Utilities Directory Reference

Directories

- directory VTK

10.10 VTK Directory Reference

Directory dependency graph for VTK:

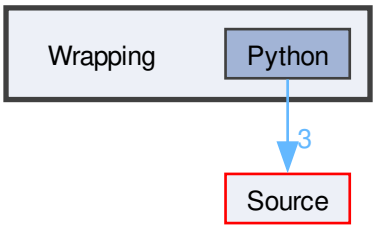


Files

- file vtkGDCMImageReader.h
- file vtkGDCMImageReader2.h
- file vtkGDCMImageWriter.h
- file vtkGDCMMedicalImageProperties.h
- file vtkGDCMPolyDataReader.h
- file vtkGDCMPolyDataWriter.h
- file vtkGDCMTesting.h
- file vtkGDCMThreadedImageReader.h
- file vtkGDCMThreadedImageReader2.h
- file vtkImageColorViewer.h
- file vtkImageMapToColors16.h
- file vtkImageMapToWindowLevelColors2.h
- file vtkImagePlanarComponentsToComponents.h
- file vtkImageRGBToYBR.h
- file vtkImageYBRToRGB.h
- file vtkLookupTable16.h
- file vtkRTStructSetProperties.h

10.11 Wrapping Directory Reference

Directory dependency graph for Wrapping:



Directories

- directory Python

Chapter 11

Namespace Documentation

11.1 gdcm Namespace Reference

Namespaces

- namespace network
- namespace SegmentHelper
- namespace terminal

  Class for Terminal.

Classes

- class AbortEvent
- class AnonymizeEvent

  AnonymizeEvent.
- class Anonymizer

  Anonymizer.
- class AnyEvent
- class ApplicationEntity

  ApplicationEntity.
- class ASN1

  Class for ASN1.
- class Attribute

  Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.
- class Attribute< Group, Element, TVR, VM::VM1 >
- class Attribute< Group, Element, TVR, VM::VM1_3 >
- class Attribute< Group, Element, TVR, VM::VM1_8 >
- class Attribute< Group, Element, TVR, VM::VM1_n >
- class Attribute< Group, Element, TVR, VM::VM2_2n >
- class Attribute< Group, Element, TVR, VM::VM2_n >
- class Attribute< Group, Element, TVR, VM::VM3_3n >
- class Attribute< Group, Element, TVR, VM::VM3_n >

- class AudioCodec

  AudioCodec.
- class Base64

  Class for Base64.
- class BaseQuery

  BaseQuery.
- class BaseRootQuery

  BaseRootQuery.
- class BasicOffsetTable

  Class to represent a BasicOffsetTable.
- class Bitmap

  Bitmap class.
- class BitmapToBitmapFilter

  BitmapToBitmapFilter class.
- class BoxRegion

  Class for manipulation box region.
- class ByteBuffer

  ByteBuffer.
- class ByteSwap

  ByteSwap.
- class ByteSwapFilter

  ByteSwapFilter.
- class ByteValue

  Class to represent binary value (array of bytes).
- class CAPICryptoFactory
- class CAPICryptographicMessageSyntax
- class Cleaner

  Cleaner.
- class Codec

  Codec class.
- class Coder

  Coder.
- class CodeString

  CodeString.
- class Command

  Command superclass for callback/observer methods.
- class CommandDataSet

  Class to represent a Command DataSet.
- class CompositeNetworkFunctions

  Composite Network Functions.
- class ConstCharWrapper

  Do not use me.
- class CP246ExplicitDataElement

  Class to read/write a DataElement as CP246Explicit Data Element.
- class CryptoFactory

  Class to do handle the crypto factory.
- class CryptographicMessageSyntax

- class CSAElement

  Class to represent a CSA Element.
- class CSAHeader

  Class for CSAHeader.
- class CSAHeaderDict

  Class to represent a map of CSAHeaderDictEntry.
- class CSAHeaderDictEntry

  Class to represent an Entry in the Dict.
- class CSAHeaderDictException
- class Curve

  Curve class to handle element 50xx,3000 Curve Data.
- class DataElement

  Class to represent a Data Element either Implicit or Explicit.
- class DataElementException
- class DataEvent

  DataEvent.
- class DataSet

  Class to represent a Data Set (which contains Data Elements).
- class DataSetEvent

  DataSetEvent.
- class DataSetHelper

  DataSetHelper (internal class, not intended for user level).
- class Decoder

  Decoder.
- class DefinedTerms

  Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.
- class Defs

  FIXME I do not like the name 'Defs'.
- class DeltaEncodingCodec

  DeltaEncodingCodec compression used by some private vendor.
- class DICOMDIR

  DICOMDIR class.
- class DICOMDIRGenerator

  DICOMDIRGenerator class.
- class Dict

  Class to represent a map of DictEntry.
- class DictConverter

  Class to convert a .dic file into something else:
- class DictEntry

  Class to represent an Entry in the Dict.
- class DictPrinter

  DictPrinter class.

- class Dicts

  Class to manipulate the sum of knowledge (all the dict user load).
- class DirectionCosines

  class to handle DirectionCosines
- class Directory

  Class for manipulation directories.
- class DirectoryHelper

  DirectoryHelper.
- class DPath

  class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.google.com/g/comp.protocols.↩dicom/c/IyIH0IOBMPA`
- class DummyValueGenerator

  Class for generating dummy value.
- class Dumper

  Codec class.
- class Element

  Element class.
- class Element< TVR, VM::VM1_2 >
- class Element< TVR, VM::VM1_n >
- class Element< TVR, VM::VM2_2n >
- class Element< TVR, VM::VM2_n >
- class Element< TVR, VM::VM3_3n >
- class Element< TVR, VM::VM3_4 >
- class Element< TVR, VM::VM3_n >
- class Element< VR::AS, VM::VM5 >
- class Element< VR::OB, VM::VM1 >
- class Element< VR::OW, VM::VM1 >
- class ElementDisableCombinations

  A class which is used to produce compile errors for an invalid combination of template parameters.
- class ElementDisableCombinations< VR::OB, VM::VM1_n >
- class ElementDisableCombinations< VR::OW, VM::VM1_n >
- class EmptyMaskGenerator

  EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory.
- class EncapsulatedDocument

  EncapsulatedDocument.
- class EncodingImplementation

  EncodingImplementation.
- class EncodingImplementation< VR::VRASCII >
- class EncodingImplementation< VR::VRBINARY >
- class EndEvent
- class EnumeratedValues

  Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:
- class EquipmentManufacturer
- class Event

  superclass for callback/observer methods

- class Exception

  Exception.
- class ExitEvent
- class ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.
- class ExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.
- class Fiducials

  Fiducials.
- class File

  a DICOM File
- class FileAnonymizer

  FileAnonymizer.
- class FileChangeTransferSyntax

  FileChangeTransferSyntax.
- class FileDecompressLookupTable

  FileDecompressLookupTable class.
- class FileDerivation

  FileDerivation class.
- class FileExplicitFilter

  FileExplicitFilter class.
- class FileMetaInformation

  Class to represent a File Meta Information.
- class Filename

  Class to manipulate file name's.
- class FileNameEvent

  FileNameEvent.
- class FilenameGenerator

  FilenameGenerator.
- class FileSet
- class FileStreamer

  FileStreamer.
- class FileWithName

  FileWithName.
- class FindPatientRootQuery

  PatientRootQuery.
- class FindStudyRootQuery

  FindStudyRootQuery.
- class Fragment

  Class to represent a Fragment.
- class Global

  Global.
- class GroupDict

  Class to represent the mapping from group number to its abbreviation and name.
- class IconImageFilter

  IconImageFilter.
- class IconImageGenerator

IconImageGenerator.

- struct ignore_char
- class Image

  Image.
- class ImageApplyLookupTable

  ImageApplyLookupTable class.
- class ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class.
- class ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class.
- class ImageChangeTransferSyntax

  ImageChangeTransferSyntax class.
- class ImageCodec

  ImageCodec.
- class ImageConverter

  Image Converter.
- class ImageFragmentSplitter

  ImageFragmentSplitter class.
- class ImageHelper

  ImageHelper (internal class, not intended for user level).
- class ImageReader

  ImageReader.
- class ImageRegionReader

  ImageRegionReader.
- class ImageToImageFilter

  ImageToImageFilter class.
- class ImageWriter

  ImageWriter.
- class ImplicitDataElement

  Class to represent an Implicit VR Data Element.
- class InitializeEvent
- class IOD

  Class for representing a IOD.
- class IODEntry

  Class for representing a IODEntry.
- class IODs

  Class for representing a IODs.
- class IPPSorter

  IPPSorter.
- class Item

  Class to represent an Item.
- class IterationEvent
- class JPEG12Codec

  Class to do JPEG 12bits (lossy & lossless).
- class JPEG16Codec

  Class to do JPEG 16bits (lossless).
- class JPEG2000Codec

Class to do JPEG 2000.

- class JPEG8Codec

  Class to do JPEG 8bits (lossy & lossless).
- class JPEGCodec

  JPEG codec.
- class JPEGLSCodec

  JPEG-LS.
- class JSON
- class KAKADUCodec

  KAKADUCodec.
- class LO

  LO.
- class LookupTable

  LookupTable class.
- class Macro

  Class for representing a Macro.
- class Macros

  Class for representing a Modules.
- class MD5

  Class for MD5.
- class MEC_MR3

  Class for MEC_MR3.
- class MediaStorage

  MediaStorage.
- class MemberCommand

  Command subclass that calls a pointer to a member function.
- class MeshPrimitive

  This class defines surface mesh primitives.
- class ModalityPerformedProcedureStepCreateQuery

  ModalityPerformedProcedureStepCreateQuery.
- class ModalityPerformedProcedureStepSetQuery

  ModalityPerformedProcedureStepSetQuery.
- class ModifiedEvent
- class Module

  Class for representing a Module.
- class ModuleEntry

  Class for representing a ModuleEntry.
- class Modules

  Class for representing a Modules.
- class MovePatientRootQuery

  MovePatientRootQuery.
- class MoveStudyRootQuery

  MoveStudyRootQuery.
- class MrProtocol

  Class for MrProtocol.
- class NestedModuleEntries

  Class for representing a NestedModuleEntries.

- class NoEvent
- class NormalizedNetworkFunctions

  Normalized Network Functions.
- class Object

  Object.
- class OpenSSLCryptoFactory
- class OpenSSLCryptographicMessageSyntax
- class OpenSSLP7CryptoFactory
- class OpenSSLP7CryptographicMessageSyntax
- class Orientation

  class to handle Orientation
- class Overlay

  Overlay class.
- class ParseException

  ParseException Standard exception handling object.
- class Parser

  Parser ala XML_Parser from expat (SAX).
- class Patient

  See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.
- class PDBElement

  Class to represent a PDB Element.
- class PDBHeader

  Class for PDBHeader.
- class PDFCodec

  PDFCodec class.
- class PersonName

  PersonName class.
- class PGXCodec

  Class to do PGX.
- class PhotometricInterpretation

  Class to represent an PhotometricInterpretation.
- class PixelFormat

  PixelFormat.
- class Pixmap

  Pixmap class.
- class PixmapReader

  PixmapReader.
- class PixmapToPixmapFilter

  PixmapToPixmapFilter class.
- class PixmapWriter

  PixmapWriter.
- class PNMCodec

  Class to do PNM.
- class Preamble

  DICOM Preamble (Part 10).
- class PresentationContext

  PresentationContext.

- class PresentationContextGenerator

  PresentationContextGenerator.
- class Printer

  Printer class.
- class PrivateDict

  Private Dict.
- class PrivateTag

  Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner).
- class ProgressEvent

  ProgressEvent.
- class PVRGCodec

  PVRGCodec.
- class PythonFilter

  PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.
- class QueryBase

  QueryBase.
- class QueryFactory

  QueryFactory.h.
- class QueryImage

  QueryImage.
- class QueryPatient

  QueryPatient.
- class QuerySeries

  QuerySeries.
- class QueryStudy

  QueryStudy.h.
- class RAWCodec

  RAWCodec class.
- class Reader

  Reader ala DOM (Document Object Model).
- struct RealWorldValueMappingContent
- class Region

  Class for manipulation region.
- class Rescaler

  Rescale class.
- class RLECodec

  Class to do RLE.
- class Scanner

  Scanner.
- class Scanner2

  Scanner2.
- class Segment

  This class defines a segment.
- class SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.
- class SegmentReader

This class defines a segment reader.

- class SegmentWriter

  This class defines a segment writer.
- class SequenceOfFragments

  Class to represent a Sequence Of Fragments.
- class SequenceOfItems

  Class to represent a Sequence Of Items.
- class SerieHelper

  SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.
- class Series

  Series.
- class ServiceClassUser

  ServiceClassUser.
- class SHA1

  Class for SHA1.
- class SimpleMemberCommand

  Command subclass that calls a pointer to a member function.
- class SimpleSubjectWatcher

  SimpleSubjectWatcher.
- class SmartPointer

  Class for Smart Pointer.
- class SOPClassUIDToIOD

  Class convert a class SOP Class UID into IOD.
- class Sorter

  Sorter.
- class Spacing

  Class for Spacing.
- class Spectroscopy

  Spectroscopy class.
- class SplitMosaicFilter

  SplitMosaicFilter class.
- class StartEvent
- struct static_assert_test
- struct STATIC_ASSERTION_FAILURE
- struct STATIC_ASSERTION_FAILURE< true >
- class StreamImageReader

  StreamImageReader.
- class StreamImageWriter

  StreamImageReader.
- class StrictScanner

  StrictScanner.
- class StrictScanner2

  StrictScanner2.
- class String

  String.
- class StringFilter

StringFilter.

- class Study

  Study.
- class Subject

  Subject.
- class Surface

  This class defines a SURFACE IE.
- class SurfaceHelper

  SurfaceHelper.
- class SurfaceReader

  This class defines a SURFACE IE reader.
- class SurfaceWriter

  This class defines a SURFACE IE writer.
- class SwapCode

  SwapCode representation.
- class SwapperDoOp
- class SwapperNoOp
- class System

  Class to do system operation.
- class Table

  Table.
- class TableEntry

  TableEntry.
- class TableReader

  Class for representing a TableReader.
- class Tag

  Class to represent a DICOM Data Element (Attribute) Tag (Group, Element).
- class TagPath

  class to handle a path of tag.
- class Testing

  class for testing
- class Trace

  Trace.
- class TransferSyntax

  Class to manipulate Transfer Syntax.
- class Type

  Type.
- struct UI
- class UIDGenerator

  Class for generating unique UID.
- class UIDs

  all known uids
- class UNExplicitDataElement

  Class to read/write a DataElement as UNExplicit Data Element.
- class UNExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.
- class Unpacker12Bits

Pack/Unpack 12 bits pixel into 16bits.

- class Usage

  Usage.
- class UserEvent
- class UUIDGenerator

  Class for generating unique UUID.
- class Validate

  Validate class.
- class Value

  Class to represent the value of a Data Element.
- class ValueIO

  Class to dispatch template calls.
- class Version

  major/minor and build version
- class VL

  Value Length.
- class VM

  Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.
- struct VMToLength
- class VR

  VR class.
- class VR16ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.
- struct VRToEncoding
- struct VRToType
- class VRVLSize
- class VRVLSize< 0 >
- class VRVLSize< 1 >
- class Waveform

  Waveform class.
- class WLMFindQuery

  PatientRootQuery.
- class Writer

  Writer ala DOM (Document Object Model).
- class XMLDictReader

  Class for representing a XMLDictReader.
- class XMLPrinter
- class XMLPrivateDictReader

  Class for representing a XMLPrivateDictReader.

Typedefs

- typedef String<'\\', 16 > AEComp
- typedef String<'\\', 64 > ASComp
- typedef bool(* BOOL_FUNCTION_PFILE_PFILE_POINTER) (File *, File *)
- typedef String<'\\', 16 > CSComp
- typedef String<'\\', 64 > DAComp

- typedef String<'\\', 64 > DTComp
- typedef std::vector< SmartPointer< FileWithName > > FileList
- typedef Bitmap IconImage
- typedef String<'\\', 64 > LOComp
- typedef String<'\\', 64 > LTComp
- typedef ModuleEntry MacroEntry
- typedef NestedModuleEntries NestedMacroEntries
- typedef String<'\\', 64 > PNComp
- typedef String<'\\', 64 > SHComp
- typedef String<'\\', 64 > STComp
- typedef String<'\\', 16 > TMComp
- typedef String<'\\', 4294967294 > UCComp
- typedef String<'\\', 64, 0 > UIComp
- typedef String<'\\', 4294967294 > URComp
- typedef String<'\\', 64 > UTComp

Enumerations

- enum CompOperators {
  GDCM_EQUAL = 0 ,
  GDCM_DIFFERENT ,
  GDCM_GREATER ,
  GDCM_GREATEROREQUAL ,
  GDCM_LESS ,
  GDCM_LESSOREQUAL }
- enum ECharSet {
  eLatin1 = 0 ,
  eLatin2 ,
  eLatin3 ,
  eLatin4 ,
  eCyrillic ,
  eArabic ,
  eGreek ,
  eHebrew ,
  eLatin5 ,
  eJapanese ,
  eThai ,
  eJapaneseKanjiMultibyte ,
  eJapaneseSupplementaryKanjiMultibyte ,
  eKoreanHangulHanjaMultibyte ,
  eUTF8 ,
  eGB18030 }
- enum ENQueryType {
  eCreateMMPS = 0 ,
  eSetMMPS }
- enum EQueryLevel {
  ePatient = 0 ,
  eStudy = 1 ,
  eSeries = 2 ,
  eImage = 3 }

- enum EQueryType {
  eFind = 0 ,
  eMove ,
  eWLMFind }
- enum ERootType {
  ePatientRootType ,
  eStudyRootType }
- enum LodModeType {
  LD_ALL = 0x00000000 ,
  LD_NOSEQ = 0x00000001 ,
  LD_NOSHADOW = 0x00000002 ,
  LD_NOSHADOWSEQ = 0x00000004 }

Functions

- static int add1 (char *buf, int n)
- ignore_char const backslash ('\\')
- template<typename T>
  static T Clamp (int v)
- static void clean (char *mant)
- static int doround (char *buf, unsigned int n)
- VR::VRType GetVRFromTag (Tag const &tag)
- bool operator!= (const CodeString &ref, const CodeString &cs)
- bool operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)
- std::ostream & operator<< (std::ostream &_os, const IOD &_val)
- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const IODs &_val)
- std::ostream & operator<< (std::ostream &_os, const Macro &_val)
- std::ostream & operator<< (std::ostream &_os, const Macros &_val)
- std::ostream & operator<< (std::ostream &_os, const MediaStorage &ms)
- std::ostream & operator<< (std::ostream &_os, const Module &_val)
- std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const Modules &_val)
- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)
- std::ostream & operator<< (std::ostream &_os, const Tag &_val)
- std::ostream & operator<< (std::ostream &_os, const TransferSyntax &ts)
- std::ostream & operator<< (std::ostream &_os, const Type &val)
- std::ostream & operator<< (std::ostream &_os, const UI &_val)
- std::ostream & operator<< (std::ostream &_os, const UIDs &uid)
- std::ostream & operator<< (std::ostream &_os, const Usage &val)
- std::ostream & operator<< (std::ostream &_os, const VM &_val)
- std::ostream & operator<< (std::ostream &_os, const VR &val)
- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- std::ostream & operator<< (std::ostream &os, const CommandDataSet &val)
- std::ostream & operator<< (std::ostream &os, const CSAElement &val)
- std::ostream & operator<< (std::ostream &os, const CSAHeader &d)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDict &val)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDictEntry &val)

- std::ostream & operator<< (std::ostream &os, const DataElement &val)
- std::ostream & operator<< (std::ostream &os, const DataSet &val)
- std::ostream & operator<< (std::ostream &os, const Dict &val)
- std::ostream & operator<< (std::ostream &os, const DictEntry &val)
- std::ostream & operator<< (std::ostream &os, const Dicts &d)
- std::ostream & operator<< (std::ostream &os, const Directory &d)
- std::ostream & operator<< (std::ostream &os, const DPath &val)
- std::ostream & operator<< (std::ostream &os, const Event &e)

  Generic inserter operator for Event and its subclasses.
- std::ostream & operator<< (std::ostream &os, const File &val)
- std::ostream & operator<< (std::ostream &os, const FileMetaInformation &val)
- std::ostream & operator<< (std::ostream &os, const FileSet &f)
- std::ostream & operator<< (std::ostream &os, const Fragment &val)
- std::ostream & operator<< (std::ostream &os, const Global &g)
- std::ostream & operator<< (std::ostream &os, const Item &val)
- std::ostream & operator<< (std::ostream &os, const MrProtocol &d)
- std::ostream & operator<< (std::ostream &os, const Object &obj)
- std::ostream & operator<< (std::ostream &os, const Orientation &o)
- std::ostream & operator<< (std::ostream &os, const PDBElement &val)
- std::ostream & operator<< (std::ostream &os, const PDBHeader &d)
- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &val)
- std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)
- std::ostream & operator<< (std::ostream &os, const Preamble &val)
- std::ostream & operator<< (std::ostream &os, const PrivateDict &val)
- std::ostream & operator<< (std::ostream &os, const PrivateTag &val)
- std::ostream & operator<< (std::ostream &os, const Region &r)
- std::ostream & operator<< (std::ostream &os, const Scanner &s)
- std::ostream & operator<< (std::ostream &os, const Scanner2 &s)
- std::ostream & operator<< (std::ostream &os, const Sorter &s)
- std::ostream & operator<< (std::ostream &os, const StrictScanner &s)
- std::ostream & operator<< (std::ostream &os, const StrictScanner2 &s)
- std::ostream & operator<< (std::ostream &os, const SwapCode &sc)
- std::ostream & operator<< (std::ostream &os, const Version &v)
- std::ostream & operator<< (std::ostream &os, const VL &val)
- bool operator== (const CodeString &ref, const CodeString &cs)
- std::istream & operator>> (std::istream &_is, Tag &_val)
- std::istream & operator>> (std::istream &in, ignore_char const &ic)
- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
  std::istream & operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)
- template<typename T>
  static int Round (T x)
- static int roundat (char *buf, size_t bufLen, unsigned int i, int iexp)
- template<typename Float>
  static void x16printf (char *buf, int size, Float f)

Variables

- static Global GlobalInstance

11.1.1 Detailed Description

This header defines the classes for the AA Actions, Association Abort Related Actions (Table 9-9 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AE Actions, Association Establishment Related Actions (Table 9-6 of ps 3.↩ 8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the AR Actions, Association Release Related Actions (Table 9-8 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

This header defines the classes for the DT Actions, Data Transfer Related Actions (Table 9-8 of ps 3.8-2009).

Since each class is essentially a placeholder for a function pointer, I'm breaking with having each class have its own file for the sake of brevity of the number of files.

11.1.2 Typedef Documentation

11.1.2.1 AEComp

```
typedef String<'\\',16> gdcm::AEComp
```

11.1.2.2 ASComp

```
typedef String<'\\',64> gdcm::ASComp
```

11.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER

```
typedef bool(* gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER) (File *, File *)
```

11.1.2.4 CSComp

```
typedef String<'\\',16> gdcm::CSComp
```

11.1.2.5 DAComp

```
typedef String<'\\',64> gdcm::DAComp
```

Examples

TemplateEmptyImage.cxx.

11.1.2.6 DTComp

```
typedef String<'\\',64> gdcm::DTComp
```

11.1.2.7 FileList

```
typedef std::vector< SmartPointer<FileWithName> > gdcm::FileList
```

11.1.2.8 IconImage

```
typedef Bitmap gdcm::IconImage
```

Examples

ExtractIconFromFile.cxx.

11.1.2.9 LOComp

```
typedef String<'\\',64> gdcm::LOComp
```

11.1.2.10 LTComp

```
typedef String<'\\',64> gdcm::LTComp
```

11.1.2.11 MacroEntry

```
typedef ModuleEntry gdcm::MacroEntry
```

11.1.2.12 NestedMacroEntries

```
typedef NestedModuleEntries gdcm::NestedMacroEntries
```

11.1.2.13 PNComp

```
typedef String<'\\',64> gdcm::PNComp
```

11.1.2.14 SHComp

```
typedef String<'\\',64> gdcm::SHComp
```

11.1.2.15 STComp

```
typedef String<'\\',64> gdcm::STComp
```

11.1.2.16 TMComp

```
typedef String<'\\',16> gdcm::TMComp
```

Examples

TemplateEmptyImage.cxx.

11.1.2.17 UCComp

```
typedef String<'\\',4294967294> gdcm::UCComp
```

11.1.2.18 UIComp

```
typedef String<'\\',64,0> gdcm::UIComp
```

11.1.2.19 URComp

```
typedef String<'\\',4294967294> gdcm::URComp
```

11.1.2.20 UTComp

```
typedef String<'\\',64> gdcm::UTComp
```

11.1.3 Enumeration Type Documentation

11.1.3.1 CompOperators

`enum gdcm::CompOperators`

Enumerator

GDCM_EQUAL	
GDCM_DIFFERENT	
GDCM_GREATER	
GDCM_GREATEROREQUAL	
GDCM_LESS	
GDCM_LESSOREQUAL	

11.1.3.2 ECharSet

`enum gdcm::ECharSet`

The character sets enumerated in PS 3.3 2009 Annex C, section C.12.1.1.2 The resulting character set is stored in 0008,0005 The conversion to the data element is performed by the QueryFactory itself

Enumerator

eLatin1	
eLatin2	
eLatin3	
eLatin4	
eCyrillic	
eArabic	
eGreek	
eHebrew	
eLatin5	
eJapanese	
eThai	
eJapaneseKanjiMultibyte	
eJapaneseSupplementaryKanjiMultibyte	
eKoreanHangulHanjaMultibyte	
eUTF8	
eGB18030	

11.1.3.3 ENQueryType

enum gdcm::ENQueryType

Enumerator

eCreateMMPS	
eSetMMPS	

11.1.3.4 EQueryLevel

enum gdcm::EQueryLevel

Enumerator

ePatient	
eStudy	
eSeries	
eImage	

11.1.3.5 EQueryType

enum gdcm::EQueryType

Enumerator

eFind	
eMove	
eWLMFind	

11.1.3.6 ERootType

enum gdcm::ERootType

Enumerator

ePatientRootType	
eStudyRootType	

11.1.3.7 LodModeType

```
enum gdcm::LodModeType
```

Enumerator

LD_ALL	
LD_NOSEQ	
LD_NOSHADOW	
LD_NOSHADOWSEQ	

11.1.4 Function Documentation

11.1.4.1 add1()

```
int gdcm::add1 (
            char * buf,
            int n)  [static]
```

References add1().

Referenced by add1(), and doround().

11.1.4.2 backslash()

```
ignore_char const gdcm::backslash (
            '\\' )
```

References backslash().

Referenced by backslash(), and gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength().

11.1.4.3 Clamp()

```
template<typename T>
T gdcm::Clamp (
            int v)  [inline], [static]
```

Referenced by gdcm::ImageChangePhotometricInterpretation::RGB2YBR(), and gdcm::ImageChangePhotometricInterpretation::YBR2R

11.1.4.4 clean()

```
void gdcm::clean (
            char * mant)  [inline], [static]
```

References clean().

Referenced by clean(), and x16printf().

11.1.4.5 doround()

```
int gdcm::doround (
            char * buf,
            unsigned int n)  [static]
```

References add1(), and doround().

Referenced by doround(), and roundat().

11.1.4.6 GetVRFromTag()

```
VR::VRType gdcm::GetVRFromTag (
            Tag const & tag)
```

11.1.4.7 operator"!=() [1/2]

```
bool gdcm::operator!= (
            const CodeString & ref,
            const CodeString & cs)  [inline]
```

Referenced by operator!=().

11.1.4.8 operator"!=() [2/2]

```
bool gdcm::operator!= (
            const DataElement & lhs,
            const DataElement & rhs)  [inline]
```

References operator!=().

11.1.4.9 operator<<() [1/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const GroupDict & _val)  [inline]
```

11.1.4.10 operator<<() [2/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const IOD & _val)  [inline]
```

11.1.4.11 operator<<() [3/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const IODEntry & _val)  [inline]
```

11.1.4.12 operator<<() [4/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const IODs & _val)  [inline]
```

11.1.4.13 operator<<() [5/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Macro & _val)  [inline]
```

11.1.4.14 operator<<() [6/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Macros & _val)  [inline]
```

11.1.4.15 operator<<() [7/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const MediaStorage & ms)  [inline]
```

11.1.4.16 operator<<() [8/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Module & _val)  [inline]
```

11.1.4.17 operator<<() [9/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const ModuleEntry & _val)  [inline]
```

11.1.4.18 operator<<() [10/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Modules & _val)  [inline]
```

11.1.4.19 operator<<() [11/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const NestedModuleEntries & _val)  [inline]
```

11.1.4.20 operator<<() [12/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Tag & _val)  [inline]
```

11.1.4.21 operator<<() [13/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const TransferSyntax & ts)  [inline]
```

11.1.4.22 operator<<() [14/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Type & val)  [inline]
```

11.1.4.23 operator<<() [15/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const UI & _val)  [inline]
```

11.1.4.24 operator<<() [16/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const UIDs & uid)  [inline]
```

References gdcm::UIDs::GetName(), and gdcm::UIDs::GetString().

11.1.4.25 operator<<() [17/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const Usage & val)  [inline]
```

11.1.4.26 operator<<() [18/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const VM & _val)  [inline]
```

11.1.4.27 operator<<() [19/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & _os,
            const VR & val)  [inline]
```

11.1.4.28 operator<<() [20/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const BasicOffsetTable & val)  [inline]
```

11.1.4.29 operator<<() [21/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CodeString & str)  [inline]
```

11.1.4.30 operator<<() [22/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CommandDataSet & val)  [inline]
```

11.1.4.31 operator<<() [23/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CSAElement & val)  [inline]
```

11.1.4.32 operator<<() [24/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CSAHeader & d)  [inline]
```

11.1.4.33 operator<<() [25/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CSAHeaderDict & val)  [inline]
```

11.1.4.34 operator<<() [26/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const CSAHeaderDictEntry & val)  [inline]
```

11.1.4.35 operator<<() [27/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const DataElement & val)  [inline]
```

11.1.4.36 operator<<() [28/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const DataSet & val)  [inline]
```

11.1.4.37 operator<<() [29/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Dict & val)  [inline]
```

11.1.4.38 operator<<() [30/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const DictEntry & val)  [inline]
```

11.1.4.39 operator<<() [31/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Dicts & d)  [inline]
```

11.1.4.40 operator<<() [32/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Directory & d)  [inline]
```

11.1.4.41 operator<<() [33/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const DPath & val)  [inline]
```

11.1.4.42 operator<<() [34/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Event & e)  [inline]
```

Generic inserter operator for Event and its subclasses.

References gdcm::Event::Print().

11.1.4.43 operator<<() [35/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const File & val)  [inline]
```

11.1.4.44 operator<<() [36/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const FileMetaInformation & val)  [inline]
```

11.1.4.45 operator<<() [37/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const FileSet & f)  [inline]
```

11.1.4.46 operator<<() [38/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Fragment & val)  [inline]
```

11.1.4.47 operator<<() [39/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Global & g)  [inline]
```

11.1.4.48 operator<<() [40/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Item & val)  [inline]
```

11.1.4.49 operator<<() [41/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const MrProtocol & d)  [inline]
```

11.1.4.50 operator<<() [42/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Object & obj)  [inline]
```

11.1.4.51 operator<<() [43/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Orientation & o)  [inline]
```

11.1.4.52 operator<<() [44/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PDBElement & val)  [inline]
```

11.1.4.53 operator<<() [45/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PDBHeader & d)  [inline]
```

11.1.4.54 operator<<() [46/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PhotometricInterpretation & val)  [inline]
```

11.1.4.55 operator<<() [47/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PixelFormat & pf)  [inline]
```

11.1.4.56 operator<<() [48/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Preamble & val)  [inline]
```

11.1.4.57 operator<<() [49/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PrivateDict & val)  [inline]
```

11.1.4.58 operator<<() [50/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const PrivateTag & val)  [inline]
```

11.1.4.59 operator<<() [51/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Region & r)  [inline]
```

References gdcm::Region::Print().

11.1.4.60 operator<<() [52/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Scanner & s)  [inline]
```

11.1.4.61 operator<<() [53/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Scanner2 & s)  [inline]
```

11.1.4.62 operator<<() [54/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Sorter & s)  [inline]
```

11.1.4.63 operator<<() [55/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const StrictScanner & s)  [inline]
```

11.1.4.64 operator<<() [56/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const StrictScanner2 & s)  [inline]
```

11.1.4.65 operator<<() [57/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const SwapCode & sc)  [inline]
```

11.1.4.66 operator<<() [58/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const Version & v)  [inline]
```

11.1.4.67 operator<<() [59/59]

```
std::ostream & gdcm::operator<< (
            std::ostream & os,
            const VL & val)  [inline]
```

11.1.4.68 operator==()

```
bool gdcm::operator== (
            const CodeString & ref,
            const CodeString & cs)  [inline]
```

11.1.4.69 operator>>() [1/3]

```
std::istream & gdcm::operator>> (
            std::istream & _is,
            Tag & _val)  [inline]
```

11.1.4.70 operator>>() [2/3]

```
std::istream & gdcm::operator>> (
            std::istream & in,
            ignore_char const & ic)  [inline]
```

References gdcm::ignore_char::m_char.

11.1.4.71 operator>>() [3/3]

```
template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
std::istream & gdcm::operator>> (
            std::istream & is,
            String< TDelimiter, TMaxLength, TPadChar > & ms)  [inline]
```

11.1.4.72 Round()

```
template<typename T>
int gdcm::Round (
            T x)  [inline], [static]
```

Referenced by gdcm::ImageChangePhotometricInterpretation::RGB2YBR(), and gdcm::ImageChangePhotometricInterpretation::YBR2RG

11.1.4.73 roundat()

```
int gdcm::roundat (
            char * buf,
            size_t bufLen,
            unsigned int i,
            int iexp)  [static]
```

References doround(), and roundat().

Referenced by roundat(), and x16printf().

11.1.4.74 x16printf()

```
template<typename Float>
void gdcm::x16printf (
            char * buf,
            int size,
            Float f)  [static]
```

References clean(), roundat(), and x16printf().

Referenced by gdcm::EncodingImplementation< VR::VRASCII >::Write(), and x16printf().

11.1.5 Variable Documentation

11.1.5.1 GlobalInstance

```
Global gdcm::GlobalInstance  [static]
```

11.2 gdcm::network Namespace Reference

Classes

- class AAbortPDU

  AAbortPDU.
- class AAssociateACPDU

  AAssociateACPDU.
- class AAssociateRJPDU

  AAssociateRJPDU.
- class AAssociateRQPDU

  AAssociateRQPDU.
- class AbstractSyntax

  AbstractSyntax.
- class ApplicationContext

ApplicationContext.

- class AReleaseRPPDU

  AReleaseRPPDU.
- class AReleaseRQPDU

  AReleaseRQPDU.
- class ARTIMTimer

  ARTIMTimer.
- class AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub.
- class BaseCompositeMessage

  BaseCompositeMessage.
- class BaseNormalizedMessage

  BaseNormalizedMessage.
- class BasePDU

  BasePDU.
- class CEchoRQ

  CEchoRQ.
- class CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.
- class CFind
- class CFindCancelRQ

  CFindCancelRQ this file defines the messages for the cfind action.
- class CFindRQ

  CFindRQ.
- class CFindRSP

  CFindRSP this file defines the messages for the cfind action.
- class CMoveCancelRq
- class CMoveRQ

  CMoveRQ.
- class CMoveRSP

  CMoveRSP this file defines the messages for the cmove action.
- class CompositeMessageFactory

  CompositeMessageFactory.
- class CStoreRQ

  CStoreRQ.
- class CStoreRSP

  CStoreRSP this file defines the messages for the cecho action.
- class DIMSE

  DIMSE.
- class ImplementationClassUIDSub

  ImplementationClassUIDSub.
- class ImplementationUIDSub

  ImplementationUIDSub.
- class ImplementationVersionNameSub

  ImplementationVersionNameSub.
- class MaximumLengthSub

  MaximumLengthSub.

- class NActionRQ

  NActionRQ.
- class NActionRSP

  NActionRSP this file defines the messages for the NAction action.
- class NCreateRQ

  NCreateRQ.
- class NCreateRSP

  NCreateRSP this file defines the messages for the ncreate action.
- class NDeleteRQ

  NDeleteRQ.
- class NDeleteRSP

  NDeleteRSP this file defines the messages for the ndelete action.
- class NEventReportRQ

  NEventReportRQ.
- class NEventReportRSP

  NEventReportRSP this file defines the messages for the neventreport action.
- class NGetRQ

  NGetRQ.
- class NGetRSP

  NGetRSP this file defines the messages for the nget action.
- class NormalizedMessageFactory
- class NSetRQ

  NSetRQ.
- class NSetRSP

  NSetRSP this file defines the messages for the nset action.
- class PDataTFPDU

  PDataTFPDU.
- class PDUFactory

  PDUFactory basically, given an initial byte, construct the.
- class PresentationContextAC

  PresentationContextAC.
- class PresentationContextRQ

  PresentationContextRQ.
- class PresentationDataValue

  PresentationDataValue.
- class RoleSelectionSub

  RoleSelectionSub.
- class ServiceClassApplicationInformation
- class SOPClassExtendedNegociationSub

  SOPClassExtendedNegociationSub.
- class TableRow
- class TransferSyntaxSub

  TransferSyntaxSub.
- struct Transition
- class ULAction

  ULAction.
- class ULActionAA1

- class ULActionAA2
- class ULActionAA3
- class ULActionAA4
- class ULActionAA5
- class ULActionAA6
- class ULActionAA7
- class ULActionAA8
- class ULActionAE1
- class ULActionAE2
- class ULActionAE3
- class ULActionAE4
- class ULActionAE5
- class ULActionAE6
- class ULActionAE7
- class ULActionAE8
- class ULActionAR1
- class ULActionAR10
- class ULActionAR2
- class ULActionAR3
- class ULActionAR4
- class ULActionAR5
- class ULActionAR6
- class ULActionAR7
- class ULActionAR8
- class ULActionAR9
- class ULActionDT1
- class ULActionDT2
- class ULBasicCallback

  ULBasicCallback.
- class ULConnection

  ULConnection.
- class ULConnectionCallback
- class ULConnectionInfo

  ULConnectionInfo.
- class ULConnectionManager

  ULConnectionManager.
- class ULEvent

  ULEvent.
- class ULTransitionTable

  ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.
- class ULWritingCallback
- class UserInformation

  UserInformation.

Enumerations

- enum EEventID {
  eAASSOCIATERequestLocalUser = 0 ,
  eTransportConnConfirmLocal ,
  eASSOCIATE_ACPDUreceived ,
  eASSOCIATE_RJPDUreceived ,
  eTransportConnIndicLocal ,
  eAASSOCIATE_RQPDUreceived ,
  eAASSOCIATEresponseAccept ,
  eAASSOCIATEresponseReject ,
  ePDATArequest ,
  ePDATATFPDU ,
  eARELEASERequest ,
  eARELEASE_RQPDUReceivedOpen ,
  eARELEASE_RPPDUReceived ,
  eARELEASEResponse ,
  eAABORTRequest ,
  eAABORTPDUReceivedOpen ,
  eTransportConnectionClosed ,
  eARTIMTimerExpired ,
  eUnrecognizedPDUReceived ,
  eEventDoesNotExist }
- enum EStateID {
  eStaDoesNotExist = 0 ,
  eSta1Idle = 1 ,
  eSta2Open = 2 ,
  eSta3WaitLocalAssoc = 4 ,
  eSta4LocalAssocDone = 8 ,
  eSta5WaitRemoteAssoc = 16 ,
  eSta6TransferReady = 32 ,
  eSta7WaitRelease = 64 ,
  eSta8WaitLocalRelease = 128 ,
  eSta9ReleaseCollisionRqLocal = 256 ,
  eSta10ReleaseCollisionAc = 512 ,
  eSta11ReleaseCollisionRq = 1024 ,
  eSta12ReleaseCollisionAcLocal = 2048 ,
  eSta13AwaitingClose = 4096 }

Functions

- int GetStateIndex (EStateID inState)

Variables

- const int cMaxEventID = eEventDoesNotExist
- const int cMaxStateID = 13

11.2.1 Enumeration Type Documentation

11.2.1.1 EEventID

enum gdcm::network::EEventID

Enumerator

eAASSOCIATERequestLocalUser	
eTransportConnConfirmLocal	
eASSOCIATE_ACPDUreceived	
eASSOCIATE_RJPDUreceived	
eTransportConnIndicLocal	
eAASSOCIATE_RQPDUreceived	
eAASSOCIATEresponseAccept	
eAASSOCIATEresponseReject	
ePDATArequest	
ePDATATFPDU	
eARELEASERequest	
eARELEASE_RQPDUReceivedOpen	
eARELEASE_RPPDUReceived	
eARELEASEResponse	
eAABORTRequest	
eAABORTPDUReceivedOpen	
eTransportConnectionClosed	
eARTIMTimerExpired	
eUnrecognizedPDUReceived	
eEventDoesNotExist	

11.2.1.2 EStateID

enum gdcm::network::EStateID

Each network connection will be in a particular state at any given time. Those states have IDs as described in the standard ps3.8-2009, roughly 1-13. This enumeration lists those states. The actual ULState class will contain more information about transitions to other states.

name and date: 16 sept 2010 mmr

Enumerator

eStaDoesNotExist	

eSta1Idle	
eSta2Open	
eSta3WaitLocalAssoc	
eSta4LocalAssocDone	
eSta5WaitRemoteAssoc	
eSta6TransferReady	
eSta7WaitRelease	
eSta8WaitLocalRelease	
eSta9ReleaseCollisionRqLocal	
eSta10ReleaseCollisionAc	
eSta11ReleaseCollisionRq	
eSta12ReleaseCollisionAcLocal	
eSta13AwaitingClose	

11.2.2 Function Documentation

11.2.2.1 GetStateIndex()

```
int gdcm::network::GetStateIndex (
            EStateID inState)  [inline]
```

References eSta10ReleaseCollisionAc, eSta11ReleaseCollisionRq, eSta12ReleaseCollisionAcLocal, eSta13AwaitingClose, eSta1Idle, eSta2Open, eSta3WaitLocalAssoc, eSta4LocalAssocDone, eSta5WaitRemoteAssoc, eSta6TransferReady, eSta7WaitRelease, eSta8WaitLocalRelease, eSta9ReleaseCollisionRqLocal, and eStaDoesNotExist.

11.2.3 Variable Documentation

11.2.3.1 cMaxEventID

```
const int gdcm::network::cMaxEventID = eEventDoesNotExist
```

11.2.3.2 cMaxStateID

```
const int gdcm::network::cMaxStateID = 13
```

Referenced by gdcm::network::TableRow::TableRow(), and gdcm::network::TableRow::~TableRow().

11.3 gdcm::SegmentHelper Namespace Reference

Classes

- struct BasicCodedEntry

  This structure defines a basic coded entry with all of its attributes.

11.4 gdcm::terminal Namespace Reference

Class for Terminal.

Enumerations

- enum Attribute {
  reset = 0 ,
  bright = 1 ,
  dim = 2 ,
  underline = 3 ,
  blink = 5 ,
  reverse = 7 ,
  hidden = 8 }
- enum Color {
  black = 0 ,
  red ,
  green ,
  yellow ,
  blue ,
  magenta ,
  cyan ,
  white }
- enum Mode {
  CONSOLE = 0 ,
  VT100 }

Functions

- GDCM_EXPORT std::string setattribute (Attribute att)
- GDCM_EXPORT std::string setbgcolor (Color c)
- GDCM_EXPORT std::string setfgcolor (Color c)
- GDCM_EXPORT void setmode (Mode m)

11.4.1 Detailed Description

Class for Terminal.

Allow one to print in color in a shell

- support VT100 compatible shell
- win32 console

11.4.2 Enumeration Type Documentation

11.4.2.1 Attribute

enum gdcm::terminal::Attribute

Enumerator

reset	
bright	
dim	
underline	
blink	
reverse	
hidden	

11.4.2.2 Color

enum gdcm::terminal::Color

Enumerator

black	
red	
green	
yellow	
blue	
magenta	
cyan	
white	

11.4.2.3 Mode

enum gdcm::terminal::Mode

Enumerator

CONSOLE	
VT100	

11.4.3 Function Documentation

11.4.3.1 setattribute()

```
GDCM_EXPORT std::string gdcm::terminal::setattribute (
            Attribute att)
```

References GDCM_EXPORT.

11.4.3.2 setbgcolor()

```
GDCM_EXPORT std::string gdcm::terminal::setbgcolor (
            Color c)
```

References GDCM_EXPORT.

11.4.3.3 setfgcolor()

```
GDCM_EXPORT std::string gdcm::terminal::setfgcolor (
            Color c)
```

References GDCM_EXPORT.

11.4.3.4 setmode()

```
GDCM_EXPORT void gdcm::terminal::setmode (
            Mode m)
```

References GDCM_EXPORT.

Chapter 12

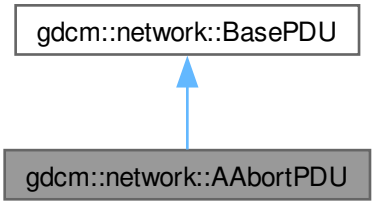
Class Documentation

12.1 gdcm::network::AAbortPDU Class Reference

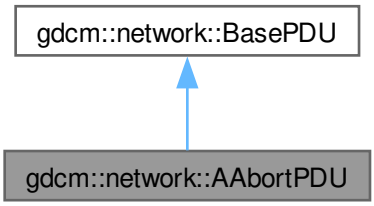
AAbortPDU.

```
#include <gdcmAAbortPDU.h>
```

Inheritance diagram for gdcm::network::AAbortPDU:



Collaboration diagram for gdcm::network::AAbortPDU:



Public Member Functions

- AAbortPDU ()
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- void SetReason (const uint8_t r)
- void SetSource (const uint8_t s)
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

Public Member Functions inherited from gdcm::network::BasePDU

- virtual ~BasePDU ()=default

12.1.1 Detailed Description

AAbortPDU.

Table 9-26 A-ABORT PDU FIELDS

12.1.2 Constructor & Destructor Documentation

12.1.2.1 AAbortPDU()

```
gdcm::network::AAbortPDU::AAbortPDU ()
```

12.1.3 Member Function Documentation

12.1.3.1 IsLastFragment()

```
bool gdcm::network::AAbortPDU::IsLastFragment () const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.1.3.2 Print()

```
void gdcm::network::AAbortPDU::Print (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.1.3.3 Read()

```
std::istream & gdcm::network::AAbortPDU::Read (
            std::istream & is)  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.1.3.4 SetReason()

```
void gdcm::network::AAbortPDU::SetReason (
            const uint8_t r)
```

12.1.3.5 SetSource()

```
void gdcm::network::AAbortPDU::SetSource (
            const uint8_t s)
```

12.1.3.6 Size()

```
size_t gdcm::network::AAbortPDU::Size () const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.1.3.7 Write()

```
const std::ostream & gdcm::network::AAbortPDU::Write (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

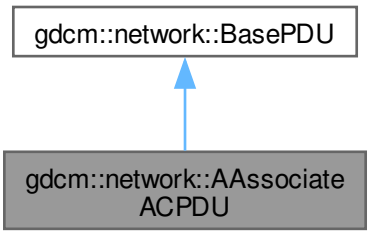
- gdcmAAbortPDU.h

12.2 gdcm::network::AAssociateACPDU Class Reference

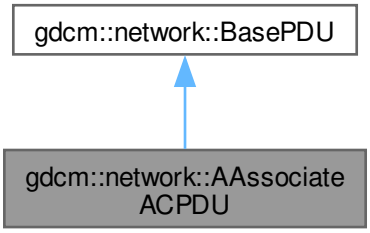
AAssociateACPDU.

```
#include <gdcmAAssociateACPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateACPDU:



Collaboration diagram for gdcm::network::AAssociateACPDU:



Public Types

- typedef std::vector< PresentationContextAC >::size_type SizeType

Public Member Functions

- AAssociateACPDU ()
- void AddPresentationContextAC (PresentationContextAC const &pcac)
- SizeType GetNumberOfPresentationContextAC () const
- const PresentationContextAC & GetPresentationContextAC (SizeType i)
- const UserInformation & GetUserInformation () const
- void InitFromRQ (AAssociateRQPDU const &rqpdu)
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- SizeType Size () const override
- const std::ostream & Write (std::ostream &os) const override

Public Member Functions inherited from gdcm::network::BasePDU

- virtual ~BasePDU ()=default

Protected Member Functions

- void SetCalledAETitle (const char calledaetitle[16])
- void SetCallingAETitle (const char callingaetitle[16])

Friends

- class AAssociateRQPDU

12.2.1 Detailed Description

AAssociateACPDU.

Table 9-17 ASSOCIATE-AC PDU fields

12.2.2 Member Typedef Documentation

12.2.2.1 SizeType

```
typedef std::vector<PresentationContextAC>::size_type gdcm::network::AAssociateACPDU::SizeType
```

12.2.3 Constructor & Destructor Documentation

12.2.3.1 AAssociateACPDU()

```
gdcm::network::AAssociateACPDU::AAssociateACPDU ()
```

12.2.4 Member Function Documentation

12.2.4.1 AddPresentationContextAC()

```
void gdcm::network::AAssociateACPDU::AddPresentationContextAC (
            PresentationContextAC const & pcac)
```

12.2.4.2 GetNumberOfPresentationContextAC()

```
SizeType gdcm::network::AAssociateACPDU::GetNumberOfPresentationContextAC () const  [inline]
```

12.2.4.3 GetPresentationContextAC()

```
const PresentationContextAC & gdcm::network::AAssociateACPDU::GetPresentationContextAC (
            SizeType i)  [inline]
```

12.2.4.4 GetUserInformation()

```
const UserInformation & gdcm::network::AAssociateACPDU::GetUserInformation () const  [inline]
```

12.2.4.5 InitFromRQ()

```
void gdcm::network::AAssociateACPDU::InitFromRQ (
            AAssociateRQPDU const & rqpdu)
```

References AAssociateRQPDU.

12.2.4.6 IsLastFragment()

```
bool gdcm::network::AAssociateACPDU::IsLastFragment () const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.2.4.7 Print()

```
void gdcm::network::AAssociateACPDU::Print (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.2.4.8 Read()

```
std::istream & gdcm::network::AAssociateACPDU::Read (
            std::istream & is)  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.2.4.9 SetCalledAETitle()

```
void gdcm::network::AAssociateACPDU::SetCalledAETitle (
            const char calledaetitle[16])  [protected]
```

12.2.4.10 SetCallingAETitle()

```
void gdcm::network::AAssociateACPDU::SetCallingAETitle (
            const char callingaetitle[16])  [protected]
```

References AAssociateRQPDU.

12.2.4.11 Size()

```
SizeType gdcm::network::AAssociateACPDU::Size () const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.2.4.12 Write()

```
const std::ostream & gdcm::network::AAssociateACPDU::Write (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.2.5 Friends And Related Symbol Documentation

12.2.5.1 AAssociateRQPDU

```
friend class AAssociateRQPDU  [friend]
```

References AAssociateRQPDU.

Referenced by AAssociateRQPDU, InitFromRQ(), and SetCallingAETitle().

The documentation for this class was generated from the following file:

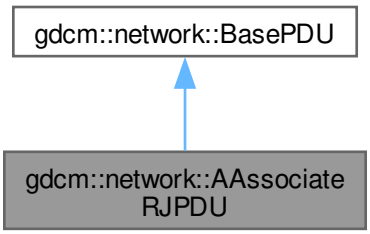
- gdcmAAssociateACPDU.h

12.3 gdcm::network::AAssociateRJPDU Class Reference

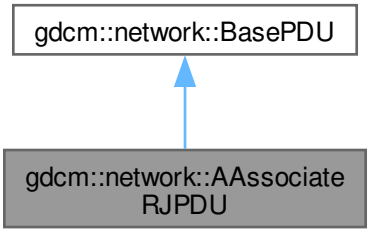
AAssociateRJPDU.

```
#include <gdcmAAssociateRJPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateRJPDU:



Collaboration diagram for gdcm::network::AAssociateRJPDU:



Public Member Functions

- AAssociateRJPDU ()
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

Public Member Functions inherited from gdcm::network::BasePDU

- virtual ~BasePDU ()=default

12.3.1 Detailed Description

AAssociateRJPDU.

Table 9-21 ASSOCIATE-RJ PDU FIELDS

12.3.2 Constructor & Destructor Documentation

12.3.2.1 AAssociateRJPDU()

```
gdcm::network::AAssociateRJPDU::AAssociateRJPDU ()
```

12.3.3 Member Function Documentation

12.3.3.1 IsLastFragment()

```
bool gdcm::network::AAssociateRJPDU::IsLastFragment () const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.3.3.2 Print()

```
void gdcm::network::AAssociateRJPDU::Print (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.3.3.3 Read()

```
std::istream & gdcm::network::AAssociateRJPDU::Read (
            std::istream & is)  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.3.3.4 Size()

```
size_t gdcm::network::AAssociateRJPDU::Size () const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.3.3.5 Write()

```
const std::ostream & gdcm::network::AAssociateRJPDU::Write (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

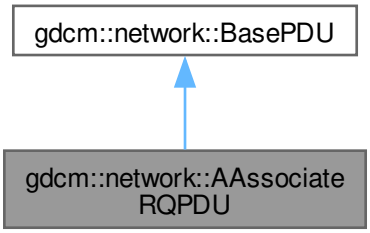
- gdcmAAssociateRJPDU.h

12.4 gdcm::network::AAssociateRQPDU Class Reference

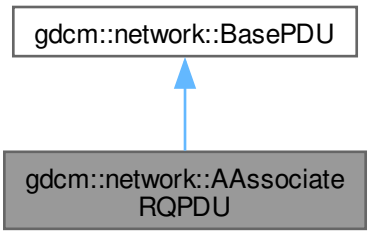
AAssociateRQPDU.

```
#include <gdcmAAssociateRQPDU.h>
```

Inheritance diagram for gdcm::network::AAssociateRQPDU:



Collaboration diagram for gdcm::network::AAssociateRQPDU:



Public Types

- typedef std::vector< PresentationContextRQ > PresentationContextArrayType
- typedef std::vector< PresentationContextRQ >::size_type SizeType

Public Member Functions

- AAssociateRQPDU ()
- AAssociateRQPDU (const AAssociateRQPDU &pdu)
- void AddPresentationContext (PresentationContextRQ const &pc)
- std::string GetCalledAETitle () const
- std::string GetCallingAETitle () const
- SizeType GetNumberOfPresentationContext () const
- PresentationContextRQ const & GetPresentationContext (SizeType i) const
- const PresentationContextRQ * GetPresentationContextByAbstractSyntax (AbstractSyntax const &absyn) const
- const PresentationContextRQ * GetPresentationContextByID (uint8_t i) const
- PresentationContextArrayType const & GetPresentationContexts ()
- const UserInformation & GetUserInformation () const
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- void SetCalledAETitle (const char calledaetitle[16])

  Set the Called AE Title.
- void SetCallingAETitle (const char callingaetitle[16])

  Set the Calling AE Title.
- void SetUserInformation (UserInformation const &ui)
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

Public Member Functions inherited from gdcm::network::BasePDU

- virtual ~BasePDU ()=default

Static Public Member Functions

- static bool IsAETitleValid (const char title[16])

  Check whether or not the.

Protected Member Functions

- std::string GetReserved43_74 () const

Friends

- class AAssociateACPDU

12.4.1 Detailed Description

AAssociateRQPDU.

Table 9-11 ASSOCIATE-RQ PDU fields

12.4.2 Member Typedef Documentation

12.4.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContextRQ> gdcm::network::AAssociateRQPDU::PresentationContextArrayType
```

12.4.2.2 SizeType

```
typedef std::vector<PresentationContextRQ>::size_type gdcm::network::AAssociateRQPDU::SizeType
```

12.4.3 Constructor & Destructor Documentation

12.4.3.1 AAssociateRQPDU() [1/2]

```
gdcm::network::AAssociateRQPDU::AAssociateRQPDU ()
```

Referenced by AAssociateRQPDU().

12.4.3.2 AAssociateRQPDU() [2/2]

```
gdcm::network::AAssociateRQPDU::AAssociateRQPDU (
            const AAssociateRQPDU & pdu)  [inline]
```

References AAssociateRQPDU().

12.4.4 Member Function Documentation

12.4.4.1 AddPresentationContext()

```
void gdcm::network::AAssociateRQPDU::AddPresentationContext (
            PresentationContextRQ const & pc)
```

12.4.4.2 GetCalledAETitle()

```
std::string gdcm::network::AAssociateRQPDU::GetCalledAETitle () const  [inline]
```

12.4.4.3 GetCallingAETitle()

```
std::string gdcm::network::AAssociateRQPDU::GetCallingAETitle () const  [inline]
```

12.4.4.4 GetNumberOfPresentationContext()

```
SizeType gdcm::network::AAssociateRQPDU::GetNumberOfPresentationContext () const  [inline]
```

12.4.4.5 GetPresentationContext()

```
PresentationContextRQ const & gdcm::network::AAssociateRQPDU::GetPresentationContext (
            SizeType i) const  [inline]
```

12.4.4.6 GetPresentationContextByAbstractSyntax()

```
const PresentationContextRQ * gdcm::network::AAssociateRQPDU::GetPresentationContextByAbstract↩
Syntax (
            AbstractSyntax const & absyn) const
```

12.4.4.7 GetPresentationContextByID()

```
const PresentationContextRQ * gdcm::network::AAssociateRQPDU::GetPresentationContextByID (
            uint8_t i) const
```

12.4.4.8 GetPresentationContexts()

```
PresentationContextArrayType const & gdcm::network::AAssociateRQPDU::GetPresentationContexts ()
[inline]
```

12.4.4.9 GetReserved43_74()

```
std::string gdcm::network::AAssociateRQPDU::GetReserved43_74 () const  [protected]
```

12.4.4.10 GetUserInformation()

```
const UserInformation & gdcm::network::AAssociateRQPDU::GetUserInformation () const  [inline]
```

12.4.4.11 IsAETitleValid()

```
bool gdcm::network::AAssociateRQPDU::IsAETitleValid (
            const char title[16])  [static]
```

Check whether or not the.

Parameters

title	is a valid AE title

12.4.4.12 IsLastFragment()

```
bool gdcm::network::AAssociateRQPDU::IsLastFragment () const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.4.4.13 Print()

```
void gdcm::network::AAssociateRQPDU::Print (
            std::ostream & os) const  [override], [virtual]
```

This function will initialize an AAssociateACPDU from the fields in the AAssociateRQPDU structure

Implements gdcm::network::BasePDU.

12.4.4.14 Read()

```
std::istream & gdcm::network::AAssociateRQPDU::Read (
            std::istream & is)  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.4.4.15 SetCalledAETitle()

```
void gdcm::network::AAssociateRQPDU::SetCalledAETitle (
            const char calledaetitle[16])
```

Set the Called AE Title.

12.4.4.16 SetCallingAETitle()

```
void gdcm::network::AAssociateRQPDU::SetCallingAETitle (
            const char callingaetitle[16])
```

Set the Calling AE Title.

12.4.4.17 SetUserInformation()

```
void gdcm::network::AAssociateRQPDU::SetUserInformation (
            UserInformation const & ui)
```

12.4.4.18 Size()

```
size_t gdcm::network::AAssociateRQPDU::Size () const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.4.4.19 Write()

```
const std::ostream & gdcm::network::AAssociateRQPDU::Write (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.4.5 Friends And Related Symbol Documentation

12.4.5.1 AAssociateACPDU

```
friend class AAssociateACPDU  [friend]
```

References AAssociateACPDU.

Referenced by AAssociateACPDU.

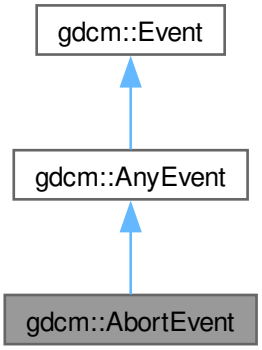
The documentation for this class was generated from the following file:

- gdcmAAssociateRQPDU.h

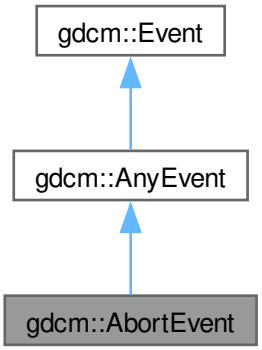
12.5 gdcm::AbortEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::AbortEvent:



Collaboration diagram for gdcm::AbortEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()

- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

- gdcmEvent.h

12.6 gdcm::network::AbstractSyntax Class Reference

AbstractSyntax.

```
#include <gdcmAbstractSyntax.h>
```

Public Member Functions

- AbstractSyntax ()
- DataElement GetAsDataElement () const
- const char * GetName () const
- bool operator== (const AbstractSyntax &as) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char *name)
- void SetNameFromUID (UIDs::TSName tsname)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.6.1 Detailed Description

AbstractSyntax.

Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS

12.6.2 Constructor & Destructor Documentation

12.6.2.1 AbstractSyntax()

```
gdcm::network::AbstractSyntax::AbstractSyntax ()
```

Referenced by operator==().

12.6.3 Member Function Documentation

12.6.3.1 GetAsDataElement()

```
DataElement gdcm::network::AbstractSyntax::GetAsDataElement () const
```

12.6.3.2 GetName()

```
const char * gdcm::network::AbstractSyntax::GetName () const  [inline]
```

12.6.3.3 operator==()

```
bool gdcm::network::AbstractSyntax::operator== (
            const AbstractSyntax & as) const  [inline]
```

References AbstractSyntax().

12.6.3.4 Print()

```
void gdcm::network::AbstractSyntax::Print (
            std::ostream & os) const
```

12.6.3.5 Read()

```
std::istream & gdcm::network::AbstractSyntax::Read (
            std::istream & is)
```

12.6.3.6 SetName()

```
void gdcm::network::AbstractSyntax::SetName (
            const char * name)  [inline]
```

12.6.3.7 SetNameFromUID()

```
void gdcm::network::AbstractSyntax::SetNameFromUID (
            UIDs::TSName tsname)
```

12.6.3.8 Size()

```
size_t gdcm::network::AbstractSyntax::Size () const
```

12.6.3.9 Write()

```
const std::ostream & gdcm::network::AbstractSyntax::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

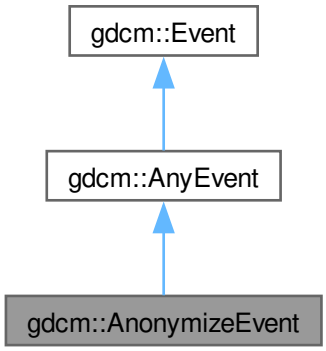
- gdcmAbstractSyntax.h

12.7 gdcm::AnonymizeEvent Class Reference

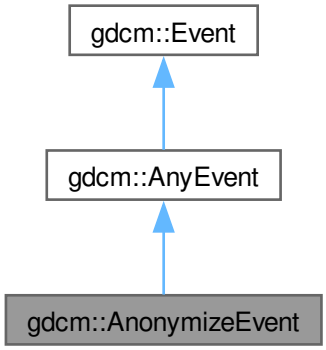
AnonymizeEvent.

```
#include <gdcmAnonymizeEvent.h>
```

Inheritance diagram for gdcm::AnonymizeEvent:



Collaboration diagram for gdcm::AnonymizeEvent:



Public Types

- typedef AnonymizeEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- AnonymizeEvent (const Self &s)
- AnonymizeEvent (Tag const &tag=0)
- ~AnonymizeEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- const char * GetEventName () const override
- Tag const & GetTag () const
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete
- void SetTag (const Tag &t)

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

12.7.1 Detailed Description

AnonymizeEvent.

Special type of event triggered during the Anonymization process

See also

Anonymizer

Examples

BasicAnonymizer.cs, Cleaner.cs, and ClinicalTrialIdentificationWorkflow.cs.

12.7.2 Member Typedef Documentation

12.7.2.1 Self

```
typedef AnonymizeEvent gdcm::AnonymizeEvent::Self
```

12.7.2.2 Superclass

```
typedef AnyEvent gdcm::AnonymizeEvent::Superclass
```

12.7.3 Constructor & Destructor Documentation

12.7.3.1 AnonymizeEvent() [1/2]

```
gdcm::AnonymizeEvent::AnonymizeEvent (
            Tag const & tag = 0)  [inline]
```

12.7.3.2 ~AnonymizeEvent()

```
gdcm::AnonymizeEvent::~AnonymizeEvent ()  [override], [default]
```

12.7.3.3 AnonymizeEvent() [2/2]

```
gdcm::AnonymizeEvent::AnonymizeEvent (
            const Self & s)  [inline]
```

12.7.4 Member Function Documentation

12.7.4.1 CheckEvent()

```
bool gdcm::AnonymizeEvent::CheckEvent (
            const ::gdcm::Event * e) const  [inline], [override]
```

12.7.4.2 GetEventName()

```
const char * gdcm::AnonymizeEvent::GetEventName () const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

12.7.4.3 GetTag()

```
Tag const & gdcm::AnonymizeEvent::GetTag () const  [inline]
```

Examples

BasicAnonymizer.cs, Cleaner.cs, and ClinicalTrialIdentificationWorkflow.cs.

12.7.4.4 MakeObject()

```
::gdcm::Event * gdcm::AnonymizeEvent::MakeObject () const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

12.7.4.5 operator=()

```
void gdcm::AnonymizeEvent::operator= (
            const Self & )  [delete]
```

12.7.4.6 SetTag()

```
void gdcm::AnonymizeEvent::SetTag (
            const Tag & t)  [inline]
```

The documentation for this class was generated from the following file:

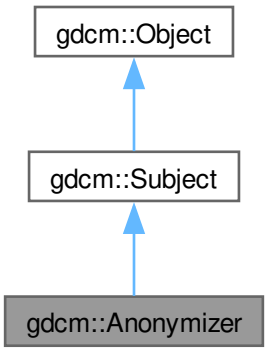
- gdcmAnonymizeEvent.h

12.8 gdcm::Anonymizer Class Reference

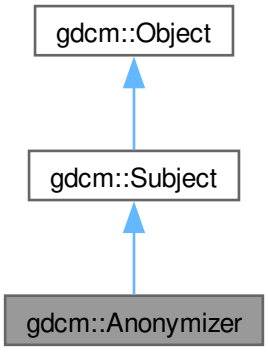
Anonymizer.

```
#include <gdcmAnonymizer.h>
```

Inheritance diagram for gdcm::Anonymizer:



Collaboration diagram for gdcm::Anonymizer:



Public Member Functions

- Anonymizer ()
- ~Anonymizer () override
- bool BasicApplicationLevelConfidentialityProfile (bool deidentify=true)
- bool Clear (PrivateTag const &pt)
- bool Clear (Tag const &t)

  Identical to 'Empty' except no action is done when tag is not present.
- bool Empty (PrivateTag const &pt)
- bool Empty (Tag const &t)

  Make Tag t empty (if not found tag will be created).
- const CryptographicMessageSyntax * GetCryptographicMessageSyntax () const
- File & GetFile ()
- bool Remove (PrivateTag const &pt)
- bool Remove (Tag const &t)

  remove a tag (even a SQ can be removed)
- bool RemoveGroupLength ()

  Main function that loop over all elements and remove group length.
- bool RemovePrivateTags ()

  Main function that loop over all elements and remove private tags.
- bool RemoveRetired ()

  Main function that loop over all elements and remove retired element.
- bool Replace (PrivateTag const &t, const char *value)
- bool Replace (PrivateTag const &t, const char *value, VL const &vl)
- bool Replace (Tag const &t, const char *value)
- bool Replace (Tag const &t, const char *value, VL const &vl)
- void SetCryptographicMessageSyntax (CryptographicMessageSyntax *cms)

  Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.
- void SetFile (const File &f)

  Set/Get File.

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static void ClearInternalUIDs ()
- static std::vector< Tag > GetBasicApplicationLevelConfidentialityProfileAttributes ()

  Return the list of Tag that will be considered when anonymizing a DICOM file.
- static SmartPointer< Anonymizer > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- bool BALCPProtect (DataSet &ds, Tag const &tag, const IOD &iod)
- bool CanEmptyTag (Tag const &tag, const IOD &iod) const
- void RecurseDataSet (DataSet &ds)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.8.1 Detailed Description

Anonymizer.

This class is a multi purpose anonymizer. It can work in 2 mode:

- Full (irreversible) anonymizer (aka dumb mode)
- reversible de-identifier/re-identifier (aka smart mode). This implements the Basic Application Level Confidentiality Profile, DICOM PS 3.15-2009

1. dumb mode This is a dumb anonymizer implementation. All it allows user is simple operation such as:

Tag based functions:

- complete removal of DICOM attribute (Remove)

- make a tag empty, ie make it's length 0 (Empty)
- replace with another string-based value (Replace)

DataSet based functions:

- Remove all group length attribute from a DICOM dataset (Group Length element are deprecated, DICOM 2008)
- Remove all private attributes
- Remove all retired attributes

All function calls actually execute the user specified request. Previous implementation were calling a general Anonymize function but traversing a std::set is O(n) operation, while a simple user specified request is O(log(n)) operation. So 'm' user interaction is O(m∗log(n)) which is $<$ O(n) complexity.

1. smart mode this mode implements the Basic Application Level Confidentiality Profile (DICOM PS 3.15-2008) In this case, it is extremely important to use the same Anonymizer class when anonymizing a FileSet. Once the Anonymizer is destroyed its memory of known (already processed) UIDs will be lost. which will make the anonymizer behaves incorrectly for attributes such as Series UID Study UID where user want some consistency. When attribute is Type 1 / Type 1C, a dummy generator will take in the existing value and produce a dummy value (a sha1 representation). sha1 algorithm is considered to be cryptographically strong (compared to md5sum) so that we meet the following two conditions:
   - Produce the same dummy value for the same input value
   - do not provide an easy way to retrieve the original value from the sha1 generated value

This class implement the Subject/Observer pattern trigger the following event:

- AnonymizeEvent
- IterationEvent
- StartEvent
- EndEvent

See also

CryptographicMessageSyntax

Examples

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, ManipulateFile.cs, and MpegVideoInfo.cs.

12.8.2 Constructor & Destructor Documentation

12.8.2.1 Anonymizer()

```
gdcm::Anonymizer::Anonymizer () [inline]
```

Referenced by New().

12.8.2.2 ~Anonymizer()

```
gdcm::Anonymizer::~Anonymizer () [override]
```

12.8.3 Member Function Documentation

12.8.3.1 BALCPProtect()

```
bool gdcm::Anonymizer::BALCPProtect (
            DataSet & ds,
            Tag const & tag,
            const IOD & iod) [protected]
```

12.8.3.2 BasicApplicationLevelConfidentialityProfile()

```
bool gdcm::Anonymizer::BasicApplicationLevelConfidentialityProfile (
            bool deidentify = true)
```

PS 3.15 / E.1.1 De-Identifier An Application may claim conformance to the Basic Application Level Confidentiality Profile as a deidentifier if it protects all Attributes that might be used by unauthorized entities to identify the patient. NOT THREAD SAFE

Examples

BasicAnonymizer.cs.

12.8.3.3 CanEmptyTag()

```
bool gdcm::Anonymizer::CanEmptyTag (
            Tag const & tag,
            const IOD & iod) const [protected]
```

12.8.3.4 Clear() [1/2]

```
bool gdcm::Anonymizer::Clear (
            PrivateTag const & pt)
```

12.8.3.5 Clear() [2/2]

```
bool gdcm::Anonymizer::Clear (
            Tag const & t)
```

Identical to 'Empty' except no action is done when tag is not present.

12.8.3.6 ClearInternalUIDs()

```
void gdcm::Anonymizer::ClearInternalUIDs ()  [static]
```

Clear the internal mapping of real UIDs to generated UIDs

Warning

the mapping is definitely lost

12.8.3.7 Empty() [1/2]

```
bool gdcm::Anonymizer::Empty (
            PrivateTag const & pt)
```

Make PrivateTag pt empty (if not found tag will be created) Pay special attention that this code must be done before any call to Empty/Remove of the associated Private Creator, but before any call to Replace.

12.8.3.8 Empty() [2/2]

```
bool gdcm::Anonymizer::Empty (
            Tag const & t)
```

Make Tag t empty (if not found tag will be created).

Examples

CreateJPIPDataSet.cxx.

12.8.3.9 GetBasicApplicationLevelConfidentialityProfileAttributes()

```
std::vector< Tag > gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes ()
[static]
```

Return the list of Tag that will be considered when anonymizing a DICOM file.

Examples

GenFakeIdentifyFile.cxx, and TraverseModules.cxx.

12.8.3.10 GetCryptographicMessageSyntax()

```
const CryptographicMessageSyntax * gdcm::Anonymizer::GetCryptographicMessageSyntax () const
```

12.8.3.11 GetFile()

```
File & gdcm::Anonymizer::GetFile () [inline]
```

Examples

BasicAnonymizer.cs, and ManipulateFile.cs.

12.8.3.12 New()

```
SmartPointer< Anonymizer > gdcm::Anonymizer::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

References Anonymizer().

12.8.3.13 RecurseDataSet()

```
void gdcm::Anonymizer::RecurseDataSet (
            DataSet & ds) [protected]
```

12.8.3.14 Remove() [1/2]

```
bool gdcm::Anonymizer::Remove (
            PrivateTag const & pt)
```

remove a private tag (even a SQ can be removed) Pay special attention that this code must be done before any call to Empty/Remove of the associated Private Creator, but before any call to Replace. When the private reservation becomes empty, no check is done to automatically remove the private creator

12.8.3.15 Remove() [2/2]

```
bool gdcm::Anonymizer::Remove (
            Tag const & t)
```

remove a tag (even a SQ can be removed)

12.8.3.16 RemoveGroupLength()

```
bool gdcm::Anonymizer::RemoveGroupLength ()
```

Main function that loop over all elements and remove group length.

Examples

ClinicalTrialAnnotate.cxx, and ManipulateFile.cs.

12.8.3.17 RemovePrivateTags()

```
bool gdcm::Anonymizer::RemovePrivateTags ()
```

Main function that loop over all elements and remove private tags.

Examples

ClinicalTrialAnnotate.cxx, and ManipulateFile.cs.

12.8.3.18 RemoveRetired()

```
bool gdcm::Anonymizer::RemoveRetired ()
```

Main function that loop over all elements and remove retired element.

12.8.3.19 Replace() [1/4]

```
bool gdcm::Anonymizer::Replace (
            PrivateTag const & t,
            const char * value)
```

12.8.3.20 Replace() [2/4]

```
bool gdcm::Anonymizer::Replace (
            PrivateTag const & t,
            const char * value,
            VL const & vl)
```

12.8.3.21 Replace() [3/4]

```
bool gdcm::Anonymizer::Replace (
            Tag const & t,
            const char * value)
```

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCII

Examples

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, ManipulateFile.cs, and MpegVideoInfo.cs.

12.8.3.22 Replace() [4/4]

```
bool gdcm::Anonymizer::Replace (
            Tag const & t,
            const char * value,
            VL const & vl)
```

when the value contains \0, it is a good idea to specify the length. This function is required when dealing with VRBINARY tag

12.8.3.23 SetCryptographicMessageSyntax()

```
void gdcm::Anonymizer::SetCryptographicMessageSyntax (
            CryptographicMessageSyntax * cms)
```

Set/Get CMS key that will be used to encrypt the dataset within BasicApplicationLevelConfidentialityProfile.

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

12.8.3.24 SetFile()

```
void gdcm::Anonymizer::SetFile (
            const File & f)  [inline]
```

Set/Get File.

Examples

BasicAnonymizer.cs, ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, ManipulateFile.cs, and MpegVideoInfo.cs.

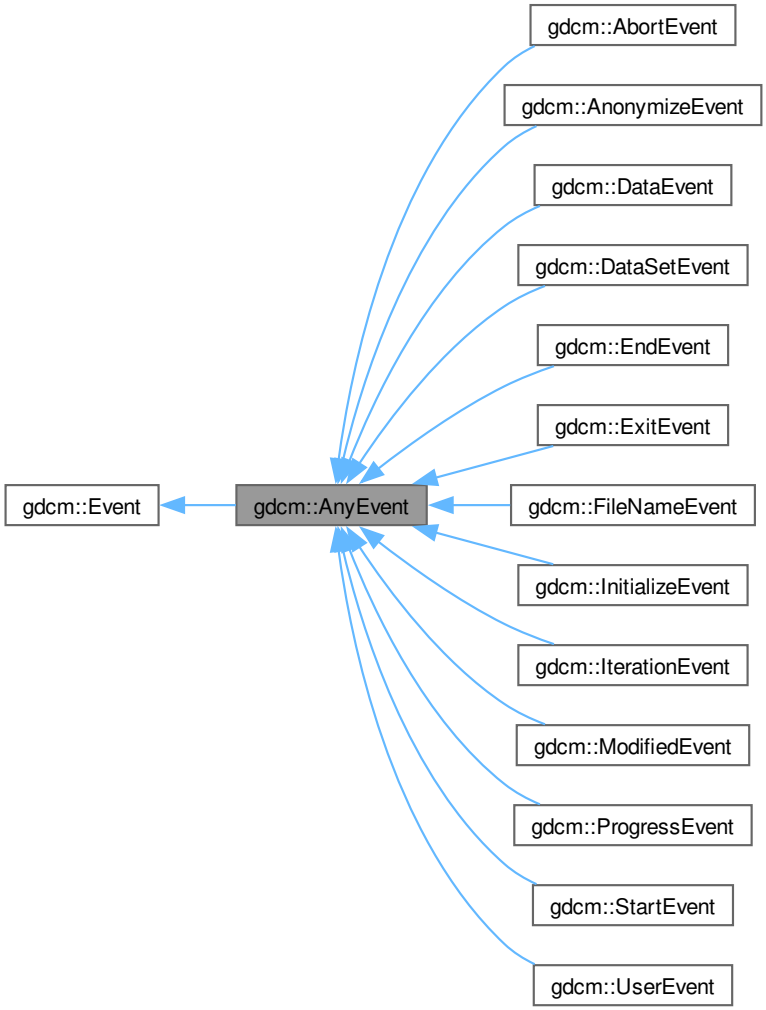
The documentation for this class was generated from the following file:

- gdcmAnonymizer.h

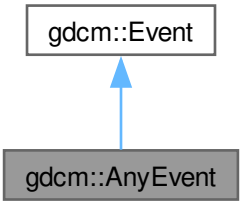
12.9 gdcm::AnyEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::AnyEvent:



Collaboration diagram for gdcm::AnyEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

- gdcmEvent.h

12.10 gdcm::network::ApplicationContext Class Reference

ApplicationContext.

```
#include <gdcmApplicationContext.h>
```

Public Member Functions

- ApplicationContext ()
- const char * GetName () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char *name)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.10.1 Detailed Description

ApplicationContext.

Table 9-12 APPLICATION CONTEXT ITEM FIELDS

Todo Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

12.10.2 Constructor & Destructor Documentation

12.10.2.1 ApplicationContext()

```
gdcm::network::ApplicationContext::ApplicationContext ()
```

12.10.3 Member Function Documentation

12.10.3.1 GetName()

```
const char * gdcm::network::ApplicationContext::GetName () const  [inline]
```

12.10.3.2 Print()

```
void gdcm::network::ApplicationContext::Print (
            std::ostream & os) const
```

12.10.3.3 Read()

```
std::istream & gdcm::network::ApplicationContext::Read (
            std::istream & is)
```

12.10.3.4 SetName()

```
void gdcm::network::ApplicationContext::SetName (
            const char * name)  [inline]
```

12.10.3.5 Size()

```
size_t gdcm::network::ApplicationContext::Size () const
```

12.10.3.6 Write()

```
const std::ostream & gdcm::network::ApplicationContext::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

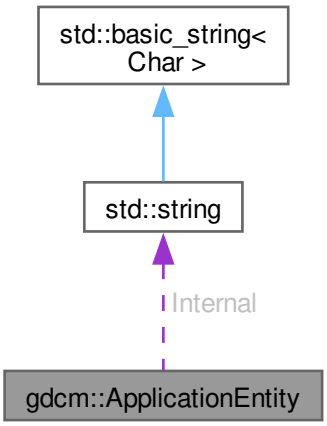
- gdcmApplicationContext.h

12.11 gdcm::ApplicationEntity Class Reference

ApplicationEntity.

```
#include <gdcmApplicationEntity.h>
```

Collaboration diagram for gdcm::ApplicationEntity:



Public Member Functions

- bool IsValid () const
- void Print (std::ostream &os) const
- void SetBlob (const std::vector< char > &v)
- void Squeeze ()

Public Attributes

- std::string Internal

Static Public Attributes

- static const unsigned int MaxLength = 16
- static const unsigned int MaxNumberOfComponents = 1
- static const char Padding = ' '
- static const char Separator = ' '

12.11.1 Detailed Description

ApplicationEntity.

- AE Application Entity
- A string of characters that identifies an Application Entity with leading and trailing spaces (20H) being non-significant. A value consisting solely of spaces shall not be used.
- Default Character Repertoire excluding character code 5CH (the BACKSLASH \ in ISO-IR 6), and control characters LF, FF, CR and ESC.
- 16 bytes maximum

12.11.2 Member Function Documentation

12.11.2.1 IsValid()

```
bool gdcm::ApplicationEntity::IsValid () const  [inline]
```

12.11.2.2 Print()

```
void gdcm::ApplicationEntity::Print (
            std::ostream & os) const  [inline]
```

12.11.2.3 SetBlob()

```
void gdcm::ApplicationEntity::SetBlob (
            const std::vector< char > & v)  [inline]
```

12.11.2.4 Squeeze()

```
void gdcm::ApplicationEntity::Squeeze ()  [inline]
```

12.11.3 Member Data Documentation

12.11.3.1 Internal

```
std::string gdcm::ApplicationEntity::Internal
```

12.11.3.2 MaxLength

```
const unsigned int gdcm::ApplicationEntity::MaxLength = 16  [static]
```

12.11.3.3 MaxNumberOfComponents

```
const unsigned int gdcm::ApplicationEntity::MaxNumberOfComponents = 1  [static]
```

12.11.3.4 Padding

```
const char gdcm::ApplicationEntity::Padding = ' '  [static]
```

12.11.3.5 Separator

```
const char gdcm::ApplicationEntity::Separator = ' '  [static]
```

The documentation for this class was generated from the following file:

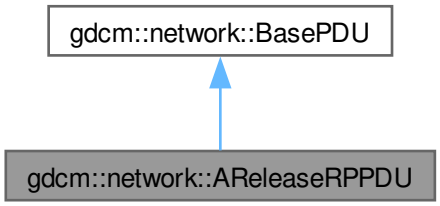
- gdcmApplicationEntity.h

12.12 gdcm::network::AReleaseRPPDU Class Reference

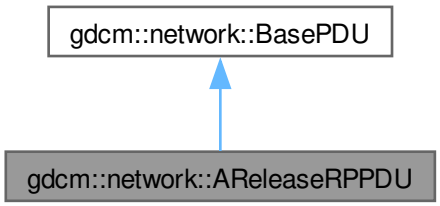
AReleaseRPPDU.

```
#include <gdcmAReleaseRPPDU.h>
```

Inheritance diagram for gdcm::network::AReleaseRPPDU:



Collaboration diagram for gdcm::network::AReleaseRPPDU:



Public Member Functions

- AReleaseRPPDU ()
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

Public Member Functions inherited from gdcm::network::BasePDU

- virtual ~BasePDU ()=default

12.12.1 Detailed Description

AReleaseRPPDU.

Table 9-25 A-RELEASE-RP PDU fields

12.12.2 Constructor & Destructor Documentation

12.12.2.1 AReleaseRPPDU()

```
gdcm::network::AReleaseRPPDU::AReleaseRPPDU ()
```

12.12.3 Member Function Documentation

12.12.3.1 IsLastFragment()

```
bool gdcm::network::AReleaseRPPDU::IsLastFragment () const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.12.3.2 Print()

```
void gdcm::network::AReleaseRPPDU::Print (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.12.3.3 Read()

```
std::istream & gdcm::network::AReleaseRPPDU::Read (
            std::istream & is)  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.12.3.4 Size()

```
size_t gdcm::network::AReleaseRPPDU::Size () const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.12.3.5 Write()

```
const std::ostream & gdcm::network::AReleaseRPPDU::Write (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

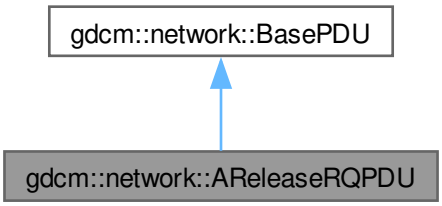
- gdcmAReleaseRPPDU.h

12.13 gdcm::network::AReleaseRQPDU Class Reference

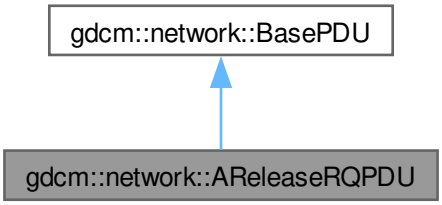
AReleaseRQPDU.

```
#include <gdcmAReleaseRQPDU.h>
```

Inheritance diagram for gdcm::network::AReleaseRQPDU:



Collaboration diagram for gdcm::network::AReleaseRQPDU:



Public Member Functions

- AReleaseRQPDU ()
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

Public Member Functions inherited from gdcm::network::BasePDU

- virtual ~BasePDU ()=default

12.13.1 Detailed Description

AReleaseRQPDU.

Table 9-24 A-RELEASE-RQ PDU FIELDS

12.13.2 Constructor & Destructor Documentation

12.13.2.1 AReleaseRQPDU()

```
gdcm::network::AReleaseRQPDU::AReleaseRQPDU ()
```

12.13.3 Member Function Documentation

12.13.3.1 IsLastFragment()

```
bool gdcm::network::AReleaseRQPDU::IsLastFragment () const  [inline], [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.13.3.2 Print()

```
void gdcm::network::AReleaseRQPDU::Print (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.13.3.3 Read()

```
std::istream & gdcm::network::AReleaseRQPDU::Read (
            std::istream & is)  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.13.3.4 Size()

```
size_t gdcm::network::AReleaseRQPDU::Size () const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.13.3.5 Write()

```
const std::ostream & gdcm::network::AReleaseRQPDU::Write (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

- gdcmAReleaseRQPDU.h

12.14 gdcm::network::ARTIMTimer Class Reference

ARTIMTimer.

```
#include <gdcmARTIMTimer.h>
```

Public Member Functions

- ARTIMTimer ()
- double GetElapsedTime () const
- bool GetHasExpired () const
- double GetTimeout () const
- void SetTimeout (double inTimeout)
- void Start ()
- void Stop ()

12.14.1 Detailed Description

ARTIMTimer.

This file contains the code for the ARTIM timer.

Basically, the ARTIM timer will just get the wall time when it's started, and then can be queried for the current time, and then can be stopped (ie, the start time reset).

Because we're trying to do this without threading, we should be able to 'start' the ARTIM timer by this mechanism, and then when waiting for a particular response, tight loop that with sleep calls and determinations of when the ARTIM timer has reached its peak. As such, this isn't a strict 'timer' in the traditional sense of the word, but more of a time keeper.

There can be only one ARTIM timer per connection.

12.14.2 Constructor & Destructor Documentation

12.14.2.1 ARTIMTimer()

```
gdcm::network::ARTIMTimer::ARTIMTimer ()
```

12.14.3 Member Function Documentation

12.14.3.1 GetElapsedTime()

```
double gdcm::network::ARTIMTimer::GetElapsedTime () const
```

12.14.3.2 GetHasExpired()

```
bool gdcm::network::ARTIMTimer::GetHasExpired () const
```

12.14.3.3 GetTimeout()

```
double gdcm::network::ARTIMTimer::GetTimeout () const
```

12.14.3.4 SetTimeout()

```
void gdcm::network::ARTIMTimer::SetTimeout (
            double inTimeout)
```

12.14.3.5 Start()

```
void gdcm::network::ARTIMTimer::Start ()
```

12.14.3.6 Stop()

```
void gdcm::network::ARTIMTimer::Stop ()
```

The documentation for this class was generated from the following file:

- gdcmARTIMTimer.h

12.15 gdcm::ASN1 Class Reference

Class for ASN1.

```
#include <gdcmASN1.h>
```

Public Member Functions

- ASN1 ()
- ASN1 (const ASN1 &)=delete
- ~ASN1 ()
- void operator= (const ASN1 &)=delete

Static Public Member Functions

- static bool ParseDump (const char *array, size_t length)
- static bool ParseDumpFile (const char *filename)

Protected Member Functions

- int TestPBKDF2 ()

12.15.1 Detailed Description

Class for ASN1.

12.15.2 Constructor & Destructor Documentation

12.15.2.1 ASN1() [1/2]

```
gdcm::ASN1::ASN1 ()
```

Referenced by ASN1(), and operator=().

12.15.2.2 ~ASN1()

```
gdcm::ASN1::~ASN1 ()
```

12.15.2.3 ASN1() [2/2]

```
gdcm::ASN1::ASN1 (
            const ASN1 & )  [delete]
```

References ASN1().

12.15.3 Member Function Documentation

12.15.3.1 operator=()

```
void gdcm::ASN1::operator= (
            const ASN1 & )  [delete]
```

References ASN1().

12.15.3.2 ParseDump()

```
bool gdcm::ASN1::ParseDump (
            const char * array,
            size_t length)  [static]
```

12.15.3.3 ParseDumpFile()

```
bool gdcm::ASN1::ParseDumpFile (
            const char * filename)  [static]
```

12.15.3.4 TestPBKDF2()

```
int gdcm::ASN1::TestPBKDF2 ()  [protected]
```

The documentation for this class was generated from the following file:

- gdcmASN1.h

12.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference

AsynchronousOperationsWindowSub.

```
#include <gdcmAsynchronousOperationsWindowSub.h>
```

Public Member Functions

- AsynchronousOperationsWindowSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.16.1 Detailed Description

AsynchronousOperationsWindowSub.

PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

12.16.2 Constructor & Destructor Documentation

12.16.2.1 AsynchronousOperationsWindowSub()

```
gdcm::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ()
```

12.16.3 Member Function Documentation

12.16.3.1 Print()

```
void gdcm::network::AsynchronousOperationsWindowSub::Print (
            std::ostream & os) const
```

12.16.3.2 Read()

```
std::istream & gdcm::network::AsynchronousOperationsWindowSub::Read (
            std::istream & is)
```

12.16.3.3 Size()

```
size_t gdcm::network::AsynchronousOperationsWindowSub::Size () const
```

12.16.3.4 Write()

```
const std::ostream & gdcm::network::AsynchronousOperationsWindowSub::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

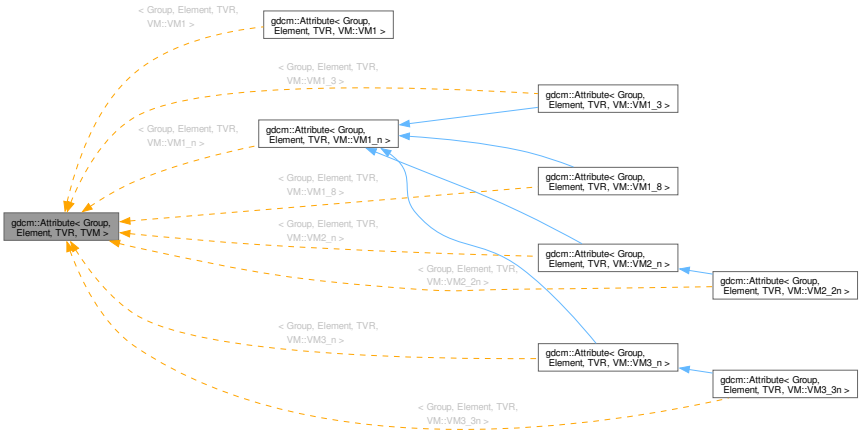
- gdcmAsynchronousOperationsWindowSub.h

12.17 gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference

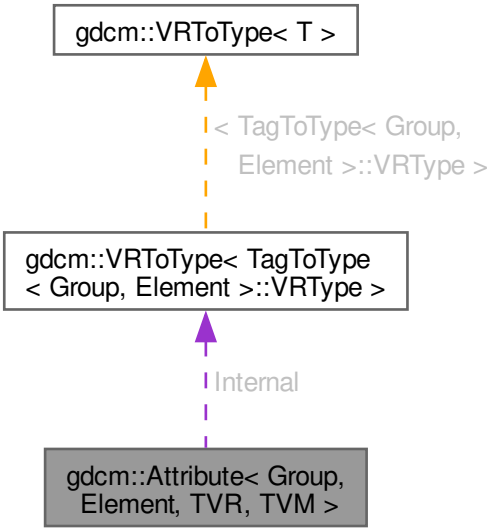
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

`#include <gdcmAttribute.h>`

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, TVM >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, TVM >:



Public Types

- enum { VMType = VMToLength<TVM>::Length }
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TVM==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- ArrayType const & operator[] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

12.17.1 Detailed Description

template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType>
class gdcm::Attribute< Group, Element, TVR, TVM >

Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.

Typical example that compile is: Attribute<0x0008,0x9007> a = {"ORIGINAL","PRIMARY","T1","NONE"};

Examples that will NOT compile are:

Attribute<0x0018,0x1182, VR::IS, VM::VM1> fd1 = {}; // not enough parameters Attribute<0x0018,0x1182, VR::IS, VM::VM2> fd2 = {0,1,2}; // too many initializers Attribute<0x0018,0x1182, VR::IS, VM::VM3> fd3 = {0,1,2}; // VM3 is not valid Attribute<0x0018,0x1182, VR::UL, VM::VM2> fd3 = {0,1}; // UL is not valid VR

Examples

CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_In FixOrientation.cxx, GenFakeIdentifyFile.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, SortImage.cxx, StreamImageReaderTest.cxx, VolumeSorter.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.17.2 Member Typedef Documentation

12.17.2.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

Examples

ReadAndPrintAttributes.cxx.

12.17.3 Member Enumeration Documentation

12.17.3.1 anonymous enum

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
anonymous enum
```

Enumerator

VMType	

12.17.4 Member Function Documentation

12.17.4.1 GDCM_STATIC_ASSERT() [1/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TVM==VM::VM1))||!((VR::VRType) TVR
&VR::VR_VM1)) )
```

References gdcm::VM::VM1, and gdcm::VR::VR_VM1.

12.17.4.2 GDCM_STATIC_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((VM::VMType) TVM &(VM::VMType)(TagToType< Group, Element >::VMType)) )
```

12.17.4.3 GDCM_STATIC_ASSERT() [3/3]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.17.4.4 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const [inline]
```

Examples

CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixOrientation.cxx, GenFakeIdentifyFile.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, and StreamImageReaderTest.cxx.

References GetNumberOfValues(), GetTag(), GetVR(), gdcm::DataElement::GetVR(), Internal, gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VRASCII.

12.17.4.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM () [inline], [static]
```

12.17.4.6 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR () [inline], [static]
```

12.17.4.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const [inline]
```

Examples

LargeVRDSExplicit.cxx.

Referenced by GetAsDataElement(), GetValue(), GetValue(), operator!=(), operator<(), gdcm::Attribute< Group, Element, TVR, VM::VM1 operator==(), Print(), SetByteValue(), SetByteValueNoSwap(), SetValue(), and SetValues().

12.17.4.8 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag () [inline], [static]
```

Examples

PatchFile.cxx, ReadAndPrintAttributes.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

Referenced by GetAsDataElement(), and Print().

12.17.4.9 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0) [inline]
```

Examples

DeriveSeries.cxx, FixOrientation.cxx, GetSequenceUltrasound.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References GetNumberOfValues(), and Internal.

Referenced by operator[ ](), and operator[ ]().

12.17.4.10 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType const & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0) const [inline]
```

References GetNumberOfValues(), and Internal.

12.17.4.11 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues () const [inline]
```

Examples

FixOrientation.cxx, LargeVRDSExplicit.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

References Internal.

Referenced by operator!=(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=(), operator<(), gdcm::Attribute< Group, Ele
operator==(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator==().

12.17.4.12 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetVM () [inline], [static]
```

12.17.4.13 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline], [static]
```

Examples

LargeVRDSExplicit.cxx.

Referenced by GetAsDataElement(), and SetFromDataElement().

12.17.4.14 operator"!=()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, TVM > & att) const [inline]
```

References GetNumberOfValues(), GetValues(), and Internal.

12.17.4.15 operator<()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, TVM > & att) const [inline]
```

References GetNumberOfValues(), GetValues(), and Internal.

12.17.4.16 operator==()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, TVM > & att) const [inline]
```

References GetNumberOfValues(), GetValues(), and Internal.

12.17.4.17 operator[]() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx)  [inline]
```

References GetValue().

12.17.4.18 operator[]() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType const & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx) const  [inline]
```

References GetValue().

12.17.4.19 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
            std::ostream & os) const  [inline]
```

References GetNumberOfValues(), GetTag(), and Internal.

12.17.4.20 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
            DataSet const & ds)  [inline]
```

Examples

LargeVRDSExplicit.cxx, SortImage.cxx, and VolumeSorter.cxx.

References gdcm::DataSet::GetDataElement(), and SetFromDataElement().

12.17.4.21 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), GetNumberOfValues(), gdcm::ByteValue::GetPointer(), and Internal.

Referenced by SetFromDataElement().

12.17.4.22 SetByteValueNoSwap()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), GetNumberOfValues(), gdcm::ByteValue::GetPointer(), and Internal.

Referenced by SetFromDataElement().

12.17.4.23 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
            DataElement const & de)  [inline]
```

Examples

GetSequenceUltrasound.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR2.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetTag(), GetVR(), gdcm::DataElement::GetVR(), gdcm::VR::INVALID, gdcm::DataElement::IsEmpty(), SetByteValue(), SetByteValueNoSwap(), and gdcm::VR::UN.

Referenced by Set(), and SetFromDataSet().

12.17.4.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

Examples

DeriveSeries.cxx, FixOrientation.cxx, ReadAndPrintAttributes.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), and SetFromDataElement().

12.17.4.25 SetValue()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
            ArrayType v,
            unsigned int idx = 0)  [inline]
```

Examples

CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, FixOrientation.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, and PatchFile.cxx.

References GetNumberOfValues(), and Internal.

12.17.4.26 SetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType)  [inline]
```

Examples

FixOrientation.cxx, and LargeVRDSExplicit.cxx.

References GetNumberOfValues(), Internal, and VMType.

12.17.5 Member Data Documentation

12.17.5.1 Internal

```
template<uint16_t Group, uint16_t Element, long long TVR = TagToType<Group, Element>::VRType, int
TVM = TagToType<Group, Element>::VMType>
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

Referenced by GetAsDataElement(), GetValue(), GetValue(), GetValues(), operator!=(), operator<(), operator==(), Print(), SetByteValue(), SetByteValueNoSwap(), SetValue(), and SetValues().

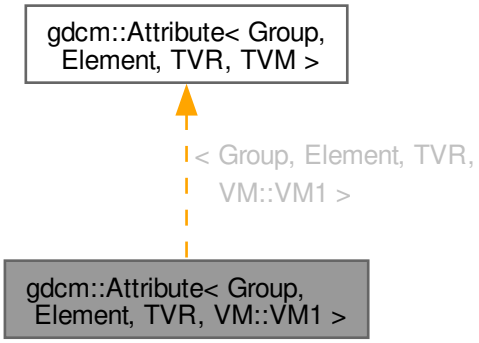
The documentation for this class was generated from the following file:

- gdcmAttribute.h

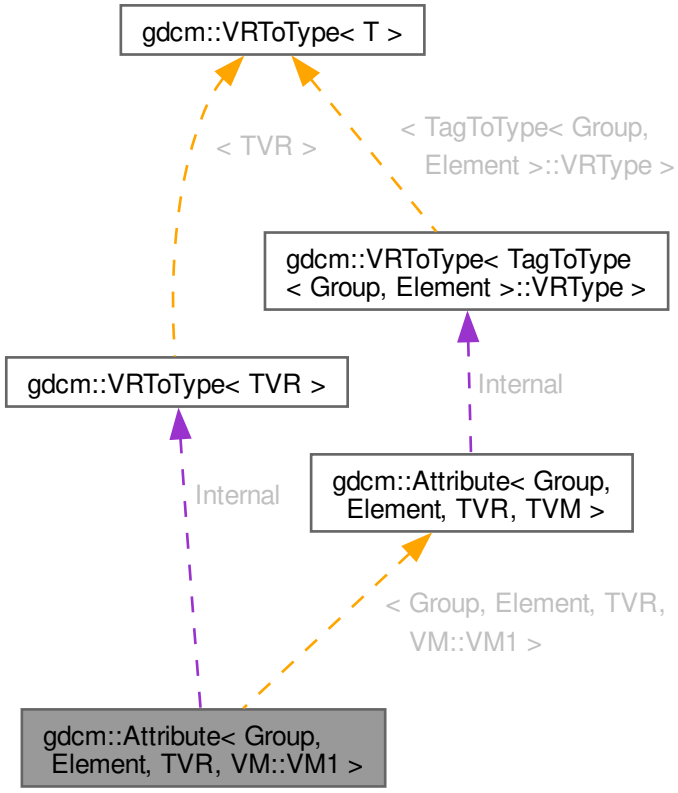
12.18 gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1 >:



Public Types

- enum
- enum { VMType = VMToLength<VM::VM1>::Length }
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length==1)
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue ()

- ArrayType const & GetValue () const
- const ArrayType ∗ GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v)
- void SetValues (const ArrayType ∗array, unsigned int numel=VMType)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal

Protected Member Functions

- void SetByteValue (const ByteValue ∗bv)
- void SetByteValueNoSwap (const ByteValue ∗bv)

12.18.1 Member Typedef Documentation

12.18.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType
```

12.18.2 Member Enumeration Documentation

12.18.2.1 anonymous enum

```
anonymous enum
```

12.18.2.2 anonymous enum

```
template<uint16_t Group, uint16_t Element, long long TVR>
anonymous enum
```

Enumerator

VMType	

12.18.3 Member Function Documentation

12.18.3.1 GDCM_STATIC_ASSERT() [1/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
            ((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRType)
TVR &VR::VR_VM1)) )
```

References gdcm::VM::VM1, and gdcm::VR::VR_VM1.

12.18.3.2 GDCM_STATIC_ASSERT() [2/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
            ((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)) )
```

References gdcm::VM::VM1.

12.18.3.3 GDCM_STATIC_ASSERT() [3/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.18.3.4 GDCM_STATIC_ASSERT() [4/4]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (
            VMToLength< VM::VM1 >::Length  = =1)
```

12.18.3.5 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement () const  [inline]
```

References GetNumberOfValues(), GetVR(), gdcm::DataElement::GetVR(), Internal, gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VRASCII.

12.18.3.6 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVM () [inline], [static]
```

12.18.3.7 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR () [inline], [static]
```

12.18.3.8 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues () const [inline]
```

Referenced by GetAsDataElement(), operator!=(), operator<(), operator==(), SetByteValue(), and SetByteValueNoSwap().

12.18.3.9 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR>
Tag gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag () [inline], [static]
```

Referenced by Print().

12.18.3.10 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue () [inline]
```

References Internal.

12.18.3.11 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue () const [inline]
```

References Internal.

12.18.3.12 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
const ArrayType * gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues () const [inline]
```

References Internal.

12.18.3.13 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM () [inline], [static]
```

References gdcm::VM::VM1.

12.18.3.14 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR () [inline], [static]
```

Referenced by GetAsDataElement(), and SetFromDataElement().

12.18.3.15 operator"!=()

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]
```

References GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and Internal.

12.18.3.16 operator<()

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator< (
            const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]
```

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), GetNumberOfValues(), gdcm::Attribute< Group, Ele
and Internal.

12.18.3.17 operator==()

```
template<uint16_t Group, uint16_t Element, long long TVR>
bool gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator== (
            const Attribute< Group, Element, TVR, VM::VM1 > & att) const [inline]
```

References GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and Internal.

12.18.3.18 operator[]()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx) [inline]
```

12.18.3.19 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print (
            std::ostream & os) const  [inline]
```

References GetTag(), and Internal.

12.18.3.20 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set (
            DataSet const & ds)  [inline]
```

References gdcm::DataSet::GetDataElement(), and SetFromDataElement().

12.18.3.21 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), GetNumberOfValues(), gdcm::ByteValue::GetPointer(), and Internal.

Referenced by SetFromDataElement().

12.18.3.22 SetByteValueNoSwap()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), GetNumberOfValues(), gdcm::ByteValue::GetPointer(), and Internal.

Referenced by SetFromDataElement().

12.18.3.23 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement (
            DataElement const & de)  [inline]
```

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetTag(), GetVR(), gdcm::DataElement::GetVR(), gdcm::VR::INVALID, gdcm::DataElement::IsEmpty(), SetByteValue(), SetByteValueNoSwap(), and gdcm::VR::UN.

Referenced by Set(), and SetFromDataSet().

12.18.3.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), and SetFromDataElement().

12.18.3.25 SetValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (
            ArrayType v)  [inline]
```

References Internal.

12.18.3.26 SetValues()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType)  [inline]
```

12.18.4 Member Data Documentation

12.18.4.1 Internal

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal
```

Referenced by GetAsDataElement(), GetValue(), GetValue(), GetValues(), operator!=(), operator<(), operator==(), Print(), SetByteValue(), SetByteValueNoSwap(), and SetValue().

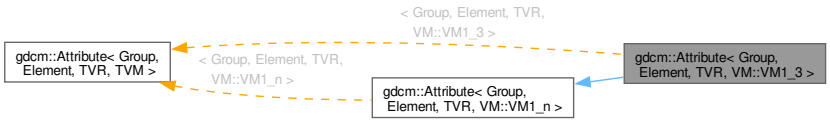
The documentation for this class was generated from the following file:

- gdcmAttribute.h

12.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

`#include <gdcmAttribute.h>`

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >:

Public Types

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- const ArrayType * GetValues () const
- VM GetVM () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- ArrayType const & operator[ ] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (ArrayType v)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

12.19.1 Member Typedef Documentation

12.19.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.19.2 Member Enumeration Documentation

12.19.2.1 anonymous enum

```
anonymous enum
```

12.19.3 Member Function Documentation

12.19.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.19.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const  [inline]
```

12.19.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ()  [inline], [static]
```

12.19.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR () [inline], [static]
```

12.19.3.5 GetNumberOfValues()

```
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const [inline]
```

12.19.3.6 GetTag()

```
Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag () [inline], [static]
```

12.19.3.7 GetValue()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0) [inline]
```

12.19.3.8 GetValues()

```
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues () const [inline]
```

12.19.3.9 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM () const [inline]
```

References gdcm::VM::VM1_3.

12.19.3.10 GetVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline], [static]
```

12.19.3.11 operator"!=()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM1_3 > & att) const [inline]
```

12.19.3.12 operator<()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, VM::VM1_3 > & att) const [inline]
```

12.19.3.13 operator==()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, VM::VM1_3 > & att) const  [inline]
```

12.19.3.14 operator[]()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx)  [inline]
```

12.19.3.15 Print()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
            std::ostream & os) const  [inline]
```

12.19.3.16 Set()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
            DataSet const & ds)  [inline]
```

12.19.3.17 SetByteValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

12.19.3.18 SetByteValueNoSwap()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

12.19.3.19 SetFromDataElement()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
            DataElement const & de)  [inline]
```

12.19.3.20 SetFromDataSet()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

12.19.3.21 SetValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
            ArrayType v,
            unsigned int idx = 0)  [inline]
```

12.19.3.22 SetValues()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType)  [inline]
```

12.19.4 Member Data Documentation

12.19.4.1 Internal

```
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

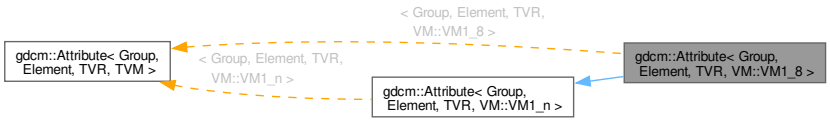
The documentation for this class was generated from the following file:

- gdcmAttribute.h

12.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >:

Public Types

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- const ArrayType * GetValues () const
- VM GetVM () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- ArrayType const & operator[ ] (unsigned int idx) const

- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (ArrayType v)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

12.20.1 Member Typedef Documentation

12.20.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.20.2 Member Enumeration Documentation

12.20.2.1 anonymous enum

```
anonymous enum
```

12.20.3 Member Function Documentation

12.20.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.20.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const  [inline]
```

12.20.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ()  [inline], [static]
```

12.20.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR ()  [inline], [static]
```

12.20.3.5 GetNumberOfValues()

```
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const  [inline]
```

12.20.3.6 GetTag()

```
Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag ()  [inline], [static]
```

12.20.3.7 GetValue()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0)  [inline]
```

12.20.3.8 GetValues()

```
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues () const  [inline]
```

12.20.3.9 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM () const  [inline]
```

References gdcm::VM::VM1_8.

12.20.3.10 GetVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR ()  [inline], [static]
```

12.20.3.11 operator"!=()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM1_8 > & att) const  [inline]
```

12.20.3.12 operator<()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, VM::VM1_8 > & att) const  [inline]
```

12.20.3.13 operator==()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, VM::VM1_8 > & att) const  [inline]
```

12.20.3.14 operator[]()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx)  [inline]
```

12.20.3.15 Print()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
            std::ostream & os) const  [inline]
```

12.20.3.16 Set()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
            DataSet const & ds)  [inline]
```

12.20.3.17 SetByteValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

12.20.3.18 SetByteValueNoSwap()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

12.20.3.19 SetFromDataElement()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
            DataElement const & de)  [inline]
```

12.20.3.20 SetFromDataSet()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

12.20.3.21 SetValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
            ArrayType v,
            unsigned int idx = 0)  [inline]
```

12.20.3.22 SetValues()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType)  [inline]
```

12.20.4 Member Data Documentation

12.20.4.1 Internal

```
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

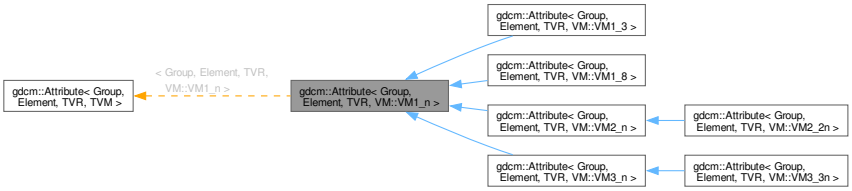
The documentation for this class was generated from the following file:

- gdcmAttribute.h

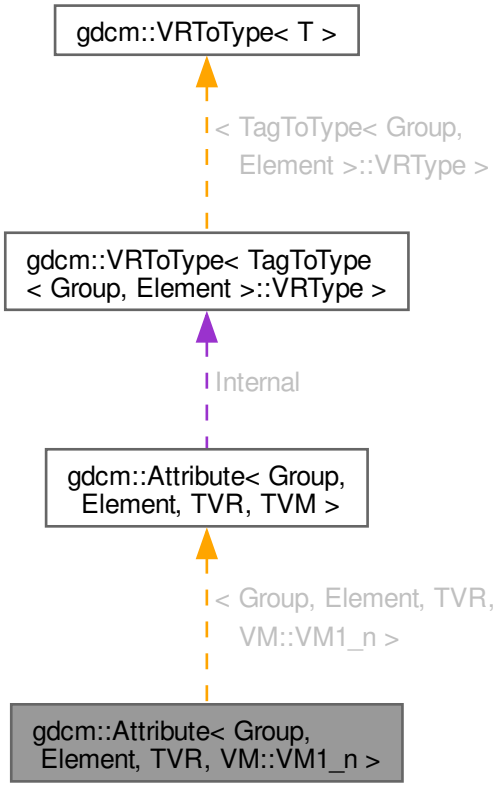
12.21 gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM1_n >:



Public Types

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)

- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- ArrayType const & operator[] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (ArrayType v)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

12.21.1 Member Typedef Documentation

12.21.1.1 ArrayType

```
template<uint16_t Group, uint16_t Element, long long TVR>
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::ArrayType
```

12.21.2 Member Enumeration Documentation

12.21.2.1 anonymous enum

```
anonymous enum
```

12.21.3 Constructor & Destructor Documentation

12.21.3.1 Attribute()

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute () [inline], [explicit]
```

12.21.3.2 ~Attribute()

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::~Attribute () [inline]
```

12.21.4 Member Function Documentation

12.21.4.1 GDCM_STATIC_ASSERT() [1/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
            ((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((V
TVR &VR::VR_VM1)) )
```

References gdcm::VM::VM1, and gdcm::VR::VR_VM1.

12.21.4.2 GDCM_STATIC_ASSERT() [2/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.21.4.3 GDCM_STATIC_ASSERT() [3/3]

```
template<uint16_t Group, uint16_t Element, long long TVR>
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (
            (VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)) )
```

References gdcm::VM::VM1_n.

12.21.4.4 GetAsDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement () const [inline]
```

References GetNumberOfValues(), GetTag(), GetVR(), gdcm::DataElement::GetVR(), gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VRASCII.

12.21.4.5 GetDictVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM ()  [inline], [static]
```

References GetVM().

12.21.4.6 GetDictVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR ()  [inline], [static]
```

12.21.4.7 GetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues () const  [inline]
```

Referenced by GetAsDataElement(), GetValue(), GetValue(), Print(), SetValue(), and SetValues().

12.21.4.8 GetTag()

```
template<uint16_t Group, uint16_t Element, long long TVR>
Tag gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag ()  [inline], [static]
```

Referenced by GetAsDataElement(), Print(), Set(), SetFromDataElement(), and SetFromDataSet().

12.21.4.9 GetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0)  [inline]
```

References GetNumberOfValues().

Referenced by operator[ ](), and operator[ ]().

12.21.4.10 GetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0) const  [inline]
```

References GetNumberOfValues().

12.21.4.11 GetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
const ArrayType * gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues () const  [inline]
```

12.21.4.12 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM ()  [inline], [static]
```

References gdcm::VM::VM1_n.

Referenced by GetDictVM(), and Print().

12.21.4.13 GetVR()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR ()  [inline], [static]
```

Referenced by GetAsDataElement(), Print(), and SetFromDataElement().

12.21.4.14 operator"!=()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM1_n > & att) const  [inline]
```

12.21.4.15 operator<()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, VM::VM1_n > & att) const  [inline]
```

12.21.4.16 operator==()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, VM::VM1_n > & att) const  [inline]
```

12.21.4.17 operator[]() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (
            unsigned int idx)  [inline]
```

References GetValue().

12.21.4.18 operator[]() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
ArrayType const & gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (
            unsigned int idx) const  [inline]
```

References GetValue().

12.21.4.19 Print()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print (
            std::ostream & os) const  [inline]
```

References GetNumberOfValues(), GetTag(), GetVM(), and GetVR().

12.21.4.20 Set()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Set (
            DataSet const & ds)  [inline]
```

References gdcm::DataSet::GetDataElement(), GetTag(), and SetFromDataElement().

12.21.4.21 SetByteValue()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), and SetValues().

Referenced by SetFromDataElement().

12.21.4.22 SetByteValueNoSwap()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

12.21.4.23 SetFromDataElement()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement (
            DataElement const & de)  [inline]
```

References gdcm::DataElement::GetByteValue(), GetTag(), gdcm::DataElement::GetTag(), GetVR(), gdcm::DataElement::GetVR(), gdcm::DataElement::IsEmpty(), and SetByteValue().

Referenced by Set(), and SetFromDataSet().

12.21.4.24 SetFromDataSet()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), GetTag(), and SetFromDataElement().

12.21.4.25 SetNumberOfValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues (
            unsigned int numel)  [inline]
```

References SetValues().

12.21.4.26 SetValue() [1/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
            ArrayType v)  [inline]
```

References SetValue().

Referenced by SetValue().

12.21.4.27 SetValue() [2/2]

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (
            unsigned int idx,
            ArrayType v)  [inline]
```

References GetNumberOfValues().

12.21.4.28 SetValues()

```
template<uint16_t Group, uint16_t Element, long long TVR>
void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (
            const ArrayType * array,
            unsigned int numel,
            bool own = false)  [inline]
```

References GetNumberOfValues().

Referenced by SetByteValue(), and SetNumberOfValues().

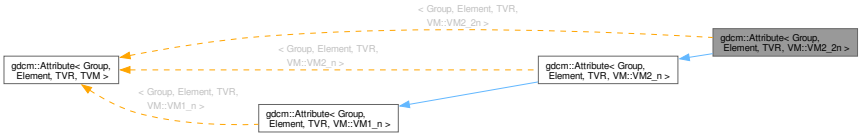
The documentation for this class was generated from the following file:

- gdcmAttribute.h

12.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >:

Public Types

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM2_n >

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM2_n >

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- const ArrayType * GetValues () const
- VM GetVM () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const

- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- ArrayType const & operator[] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (ArrayType v)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM2_n >

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Public Attributes inherited from gdcm::Attribute< Group, Element, TVR, VM::VM2_n >

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM2_n >

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

12.22.1 Member Typedef Documentation

12.22.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.22.2 Member Enumeration Documentation

12.22.2.1 anonymous enum

```
anonymous enum
```

12.22.3 Member Function Documentation

12.22.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.22.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const  [inline]
```

12.22.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ()  [inline], [static]
```

12.22.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR ()  [inline], [static]
```

12.22.3.5 GetNumberOfValues()

```
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const  [inline]
```

12.22.3.6 GetTag()

```
Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag ()  [inline], [static]
```

12.22.3.7 GetValue()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0)  [inline]
```

12.22.3.8 GetValues()

```
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues () const  [inline]
```

12.22.3.9 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM ()  [inline], [static]
```

References gdcm::VM::VM2_2n.

12.22.3.10 GetVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR ()  [inline], [static]
```

12.22.3.11 operator"!=()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM2_2n > & att) const  [inline]
```

12.22.3.12 operator<()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, VM::VM2_2n > & att) const  [inline]
```

12.22.3.13 operator==()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, VM::VM2_2n > & att) const  [inline]
```

12.22.3.14 operator[]()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx)  [inline]
```

12.22.3.15 Print()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
            std::ostream & os) const  [inline]
```

12.22.3.16 Set()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
            DataSet const & ds)  [inline]
```

12.22.3.17 SetByteValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

12.22.3.18 SetByteValueNoSwap()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

12.22.3.19 SetFromDataElement()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
            DataElement const & de)  [inline]
```

12.22.3.20 SetFromDataSet()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

12.22.3.21 SetValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
            ArrayType v,
            unsigned int idx = 0)  [inline]
```

12.22.3.22 SetValues()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType)  [inline]
```

12.22.4 Member Data Documentation

12.22.4.1 Internal

```
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

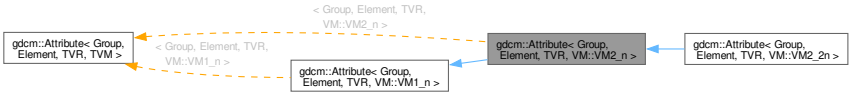
The documentation for this class was generated from the following file:

- gdcmAttribute.h

12.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_n >:

Public Types

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- const ArrayType * GetValues () const
- VM GetVM () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- ArrayType const & operator[ ] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (ArrayType v)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

12.23.1 Member Typedef Documentation

12.23.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.23.2 Member Enumeration Documentation

12.23.2.1 anonymous enum

```
anonymous enum
```

12.23.3 Member Function Documentation

12.23.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.23.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const  [inline]
```

12.23.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ()  [inline], [static]
```

12.23.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR () [inline], [static]
```

12.23.3.5 GetNumberOfValues()

```
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const [inline]
```

12.23.3.6 GetTag()

```
Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag () [inline], [static]
```

12.23.3.7 GetValue()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0) [inline]
```

12.23.3.8 GetValues()

```
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues () const [inline]
```

12.23.3.9 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM () const [inline]
```

References gdcm::VM::VM2_n.

12.23.3.10 GetVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline], [static]
```

12.23.3.11 operator"!=()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM2_n > & att) const [inline]
```

12.23.3.12 operator<()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, VM::VM2_n > & att) const [inline]
```

12.23.3.13 operator==()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, VM::VM2_n > & att) const  [inline]
```

12.23.3.14 operator[]()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx)  [inline]
```

12.23.3.15 Print()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
            std::ostream & os) const  [inline]
```

12.23.3.16 Set()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
            DataSet const & ds)  [inline]
```

12.23.3.17 SetByteValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

12.23.3.18 SetByteValueNoSwap()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

12.23.3.19 SetFromDataElement()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
            DataElement const & de)  [inline]
```

12.23.3.20 SetFromDataSet()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

12.23.3.21 SetValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
            ArrayType v,
            unsigned int idx = 0)  [inline]
```

12.23.3.22 SetValues()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType)  [inline]
```

12.23.4 Member Data Documentation

12.23.4.1 Internal

```
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

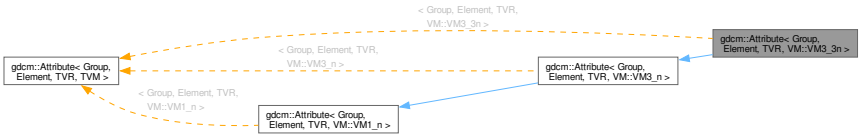
The documentation for this class was generated from the following file:

- gdcmAttribute.h

12.24 gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >:

Public Types

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM3_n >

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM3_n >

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- ArrayType const & operator[ ] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (ArrayType v)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM3_n >

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Public Attributes inherited from gdcm::Attribute< Group, Element, TVR, VM::VM3_n >

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM3_n >

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

12.24.1 Member Typedef Documentation

12.24.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.24.2 Member Enumeration Documentation

12.24.2.1 anonymous enum

```
anonymous enum
```

12.24.3 Member Function Documentation

12.24.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.24.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const  [inline]
```

12.24.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ()  [inline], [static]
```

12.24.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR ()  [inline], [static]
```

12.24.3.5 GetNumberOfValues()

```
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const  [inline]
```

12.24.3.6 GetTag()

```
Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag ()  [inline], [static]
```

12.24.3.7 GetValue()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0)  [inline]
```

12.24.3.8 GetValues()

```
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues () const  [inline]
```

12.24.3.9 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM ()  [inline], [static]
```

References gdcm::VM::VM3_3n.

12.24.3.10 GetVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR ()  [inline], [static]
```

12.24.3.11 operator"!=()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM3_3n > & att) const  [inline]
```

12.24.3.12 operator<()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, VM::VM3_3n > & att) const  [inline]
```

12.24.3.13 operator==()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, VM::VM3_3n > & att) const  [inline]
```

12.24.3.14 operator[]()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx)  [inline]
```

12.24.3.15 Print()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
            std::ostream & os) const  [inline]
```

12.24.3.16 Set()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
            DataSet const & ds)  [inline]
```

12.24.3.17 SetByteValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

12.24.3.18 SetByteValueNoSwap()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

12.24.3.19 SetFromDataElement()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
            DataElement const & de)  [inline]
```

12.24.3.20 SetFromDataSet()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

12.24.3.21 SetValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
            ArrayType v,
            unsigned int idx = 0)  [inline]
```

12.24.3.22 SetValues()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType)  [inline]
```

12.24.4 Member Data Documentation

12.24.4.1 Internal

```
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

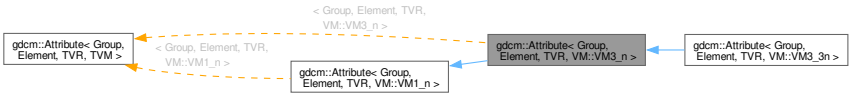
The documentation for this class was generated from the following file:

- gdcmAttribute.h

12.25 gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_n >:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM3_n >:

Public Types

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Types inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- enum
- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetValue (ArrayType v, unsigned int idx=0)
- void SetValues (const ArrayType *array, unsigned int numel=VMType)

Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- Attribute ()
- ~Attribute ()
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1) &&((VM::VMType) TagToType< Group, Element >::VMType==VM::VM1))||!((VR::VRType) TVR &VR::VR_VM1)))
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))
- DataElement GetAsDataElement () const
- unsigned int GetNumberOfValues () const
- ArrayType & GetValue (unsigned int idx=0)
- ArrayType const & GetValue (unsigned int idx=0) const
- const ArrayType * GetValues () const
- bool operator!= (const Attribute &att) const
- bool operator< (const Attribute &att) const
- bool operator== (const Attribute &att) const
- ArrayType & operator[ ] (unsigned int idx)
- ArrayType const & operator[ ] (unsigned int idx) const
- void Print (std::ostream &os) const
- void Set (DataSet const &ds)
- void SetFromDataElement (DataElement const &de)
- void SetFromDataSet (DataSet const &ds)
- void SetNumberOfValues (unsigned int numel)
- void SetValue (ArrayType v)
- void SetValue (unsigned int idx, ArrayType v)
- void SetValues (const ArrayType *array, unsigned int numel, bool own=false)

Static Public Member Functions

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- static VM GetDictVM ()
- static VR GetDictVR ()
- static Tag GetTag ()
- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- ArrayType Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

Protected Member Functions inherited from gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

- void SetByteValue (const ByteValue *bv)
- void SetByteValueNoSwap (const ByteValue *bv)

12.25.1 Member Typedef Documentation

12.25.1.1 ArrayType

```
typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, TVM >::ArrayType
```

12.25.2 Member Enumeration Documentation

12.25.2.1 anonymous enum

```
anonymous enum
```

12.25.3 Member Function Documentation

12.25.3.1 GDCM_STATIC_ASSERT()

```
gdcm::Attribute< Group, Element, TVR, TVM >::GDCM_STATIC_ASSERT (
            ((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)) )
```

12.25.3.2 GetAsDataElement()

```
DataElement gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement () const  [inline]
```

12.25.3.3 GetDictVM()

```
VM gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVM ()  [inline], [static]
```

12.25.3.4 GetDictVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetDictVR ()  [inline], [static]
```

12.25.3.5 GetNumberOfValues()

```
unsigned int gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues () const  [inline]
```

12.25.3.6 GetTag()

```
Tag gdcm::Attribute< Group, Element, TVR, TVM >::GetTag ()  [inline], [static]
```

12.25.3.7 GetValue()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::GetValue (
            unsigned int idx = 0)  [inline]
```

12.25.3.8 GetValues()

```
const ArrayType * gdcm::Attribute< Group, Element, TVR, TVM >::GetValues () const  [inline]
```

12.25.3.9 GetVM()

```
template<uint16_t Group, uint16_t Element, long long TVR>
VM gdcm::Attribute< Group, Element, TVR, VM::VM3_n >::GetVM () [inline], [static]
```

References gdcm::VM::VM3_n.

12.25.3.10 GetVR()

```
VR gdcm::Attribute< Group, Element, TVR, TVM >::GetVR () [inline], [static]
```

12.25.3.11 operator"!=()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator!= (
            const Attribute< Group, Element, TVR, VM::VM3_n > & att) const [inline]
```

12.25.3.12 operator<()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator< (
            const Attribute< Group, Element, TVR, VM::VM3_n > & att) const [inline]
```

12.25.3.13 operator==()

```
bool gdcm::Attribute< Group, Element, TVR, TVM >::operator== (
            const Attribute< Group, Element, TVR, VM::VM3_n > & att) const [inline]
```

12.25.3.14 operator[]()

```
ArrayType & gdcm::Attribute< Group, Element, TVR, TVM >::operator[] (
            unsigned int idx) [inline]
```

12.25.3.15 Print()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Print (
            std::ostream & os) const [inline]
```

12.25.3.16 Set()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::Set (
            DataSet const & ds) [inline]
```

12.25.3.17 SetByteValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue (
            const ByteValue * bv)  [inline], [protected]
```

12.25.3.18 SetByteValueNoSwap()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap (
            const ByteValue * bv)  [inline], [protected]
```

12.25.3.19 SetFromDataElement()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement (
            DataElement const & de)  [inline]
```

12.25.3.20 SetFromDataSet()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet (
            DataSet const & ds)  [inline]
```

12.25.3.21 SetValue()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValue (
            ArrayType v,
            unsigned int idx = 0)  [inline]
```

12.25.3.22 SetValues()

```
void gdcm::Attribute< Group, Element, TVR, TVM >::SetValues (
            const ArrayType * array,
            unsigned int numel = VMType)  [inline]
```

12.25.4 Member Data Documentation

12.25.4.1 Internal

```
ArrayType gdcm::Attribute< Group, Element, TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

The documentation for this class was generated from the following file:

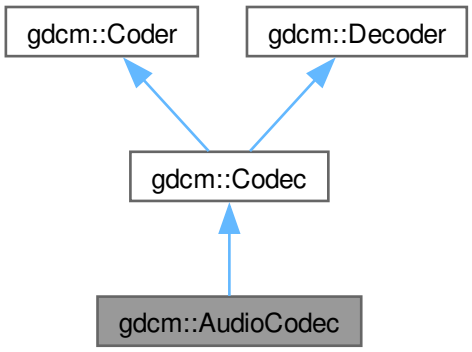
- gdcmAttribute.h

12.26 gdcm::AudioCodec Class Reference

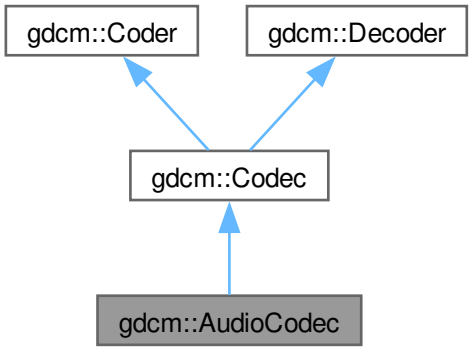
AudioCodec.

```
#include <gdcmAudioCodec.h>
```

Inheritance diagram for gdcm::AudioCodec:



Collaboration diagram for gdcm::AudioCodec:



Public Member Functions

- AudioCodec ()
- ~AudioCodec () override
- bool CanCode (TransferSyntax const &) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &) const override

  Return whether this decoder support this transfer syntax (can decode it).
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Additional Inherited Members

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Protected Member Functions inherited from gdcm::Decoder

- virtual bool DecodeByStreams (std::istream &, std::ostream &)

12.26.1 Detailed Description

AudioCodec.

12.26.2 Constructor & Destructor Documentation

12.26.2.1 AudioCodec()

```
gdcm::AudioCodec::AudioCodec ()
```

12.26.2.2 ~AudioCodec()

```
gdcm::AudioCodec::~AudioCodec () [override]
```

12.26.3 Member Function Documentation

12.26.3.1 CanCode()

```
bool gdcm::AudioCodec::CanCode (
            TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Implements gdcm::Coder.

12.26.3.2 CanDecode()

```
bool gdcm::AudioCodec::CanDecode (
            TransferSyntax const & ) const [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Implements gdcm::Decoder.

12.26.3.3 Decode()

```
bool gdcm::AudioCodec::Decode (
            DataElement const & ,
            DataElement & ) [override], [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

The documentation for this class was generated from the following file:

- gdcmAudioCodec.h

12.27 gdcm::Base64 Class Reference

Class for Base64.

```
#include <gdcmBase64.h>
```

Public Member Functions

- Base64 (const Base64 &)=delete
- void operator= (const Base64 &)=delete

Static Public Member Functions

- static size_t Decode (char ∗dst, size_t dlen, const char ∗src, size_t slen)

  Decode a base64-formatted buffer.
- static size_t Encode (char ∗dst, size_t dlen, const char ∗src, size_t slen)

  Encode a buffer into base64 format.
- static size_t GetDecodeLength (const char ∗src, size_t len)
- static size_t GetEncodeLength (const char ∗src, size_t srclen)

12.27.1 Detailed Description

Class for Base64.

12.27.2 Constructor & Destructor Documentation

12.27.2.1 Base64()

```
gdcm::Base64::Base64 (
            const Base64 & )  [delete]
```

References Base64().

Referenced by Base64(), and operator=().

12.27.3 Member Function Documentation

12.27.3.1 Decode()

```
size_t gdcm::Base64::Decode (
            char * dst,
            size_t dlen,
            const char * src,
            size_t slen)  [static]
```

Decode a base64-formatted buffer.

Parameters

dst	destination buffer
dlen	size of the buffer
src	source buffer
slen	amount of data to be decoded

Returns

0 if not successful, size of decoded otherwise

Examples

DumpExamCard.cxx, and DumpSiemensBase64.cxx.

12.27.3.2 Encode()

```
size_t gdcm::Base64::Encode (
            char * dst,
            size_t dlen,
            const char * src,
            size_t slen)  [static]
```

Encode a buffer into base64 format.

Parameters

dst	destination buffer
dlen	size of the buffer
src	source buffer
slen	amount of data to be encoded

Returns

0 if not successful, size of encoded otherwise

12.27.3.3 GetDecodeLength()

```
size_t gdcm::Base64::GetDecodeLength (
            const char * src,
            size_t len)  [static]
```

Call this function to obtain the required buffer size

Examples

DumpExamCard.cxx, and DumpSiemensBase64.cxx.

12.27.3.4 GetEncodeLength()

```
size_t gdcm::Base64::GetEncodeLength (
            const char * src,
            size_t srclen)  [static]
```

Call this function to obtain the required buffer size

12.27.3.5 operator=()

```
void gdcm::Base64::operator= (
            const Base64 & )  [delete]
```

References Base64().

The documentation for this class was generated from the following file:

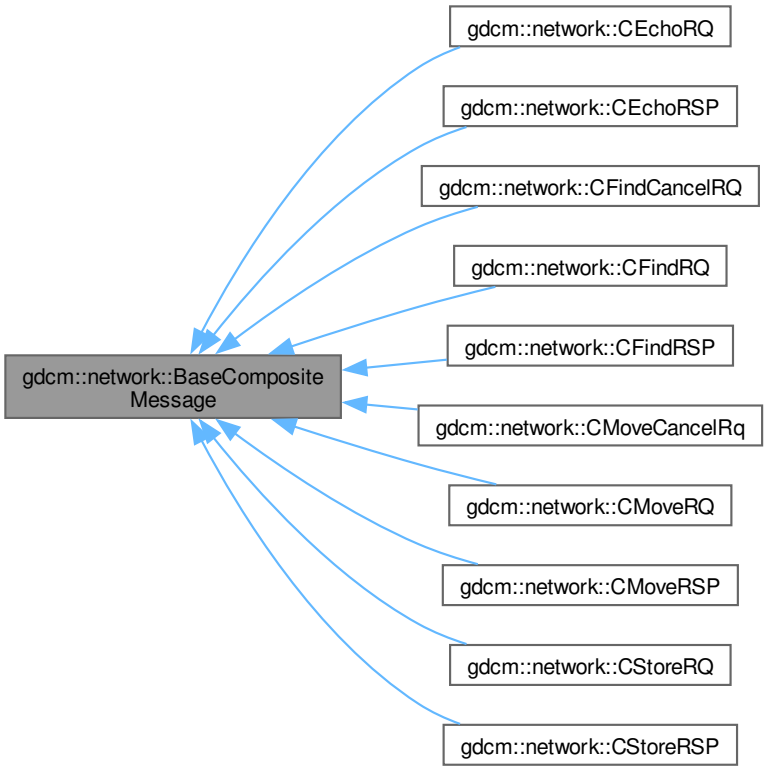
- gdcmBase64.h

12.28 gdcm::network::BaseCompositeMessage Class Reference

BaseCompositeMessage.

```
#include <gdcmBaseCompositeMessage.h>
```

Inheritance diagram for gdcm::network::BaseCompositeMessage:



Public Member Functions

- virtual ~BaseCompositeMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

12.28.1 Detailed Description

BaseCompositeMessage.

The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.

So, for the five composites:

- C-ECHO

- C-FIND
- C-MOVE
- C-GET
- C-STORE there are a series of messages. However, all of these messages are obtained as part of a PDataPDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, gdcmComposite↩ PDUFactory.

This is an abstract class. It cannot be instantiated on its own.

12.28.2 Constructor & Destructor Documentation

12.28.2.1 ~BaseCompositeMessage()

```
virtual gdcm::network::BaseCompositeMessage::~BaseCompositeMessage ()  [virtual], [default]
```

12.28.3 Member Function Documentation

12.28.3.1 ConstructPDV()

```
virtual std::vector< PresentationDataValue > gdcm::network::BaseCompositeMessage::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery)  [pure virtual]
```

Implemented in gdcm::network::CEchoRQ, gdcm::network::CFindRQ, and gdcm::network::CMoveRQ.

The documentation for this class was generated from the following file:

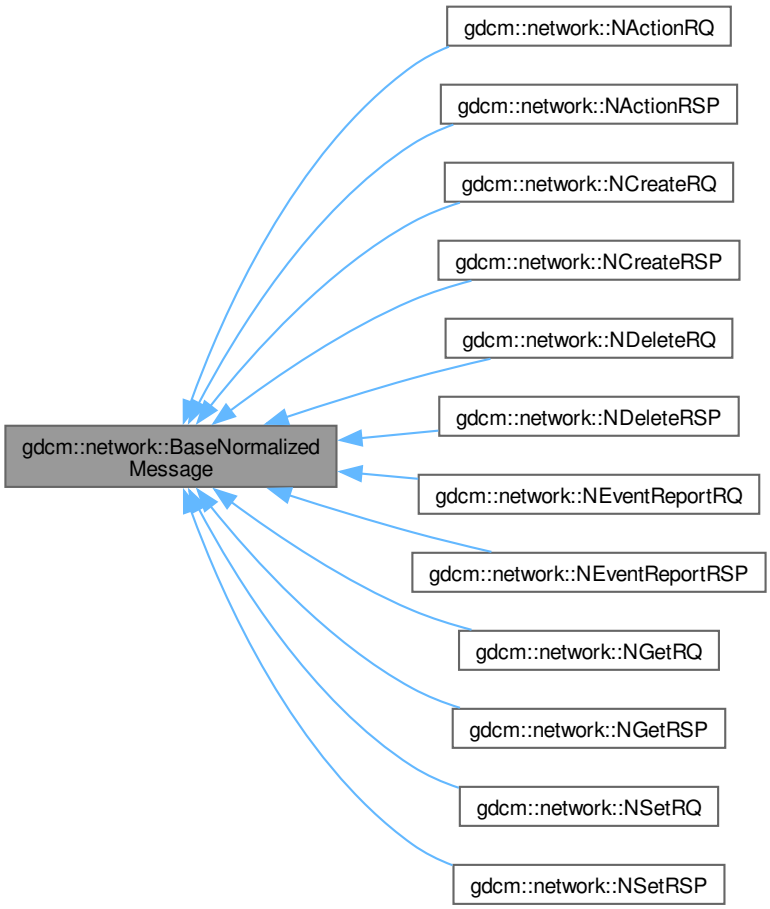
- gdcmBaseCompositeMessage.h

12.29 gdcm::network::BaseNormalizedMessage Class Reference

BaseNormalizedMessage.

```
#include <gdcmBaseNormalizedMessage.h>
```

Inheritance diagram for gdcm::network::BaseNormalizedMessage:



Public Member Functions

- virtual ~BaseNormalizedMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

12.29.1 Detailed Description

BaseNormalizedMessage.

The Normalized events described in section 3.7-2011 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2011 of the standard, and then fill in appropriate values in their datasets.

So, for the five normalized:

- N-ACTION
- N-CREATE
- N-DELETE
- N-EVENT
- N-GET
- N-SET there are a series of messages. However, all of these messages are obtained as part of a PData↩ PDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, gdcmNormalizedMessageFactory.h.

This is an abstract class. It cannot be instantiated on its own.

12.29.2 Constructor & Destructor Documentation

12.29.2.1 ∼BaseNormalizedMessage()

```
virtual gdcm::network::BaseNormalizedMessage::∼BaseNormalizedMessage () [virtual], [default]
```

12.29.3 Member Function Documentation

12.29.3.1 ConstructPDV()

```
virtual std::vector< PresentationDataValue > gdcm::network::BaseNormalizedMessage::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery) [pure virtual]
```

Implemented in gdcm::network::NActionRQ, gdcm::network::NCreateRQ, gdcm::network::NDeleteRQ, gdcm::network::NEventReportRQ, gdcm::network::NGetRQ, and gdcm::network::NSetRQ.

The documentation for this class was generated from the following file:

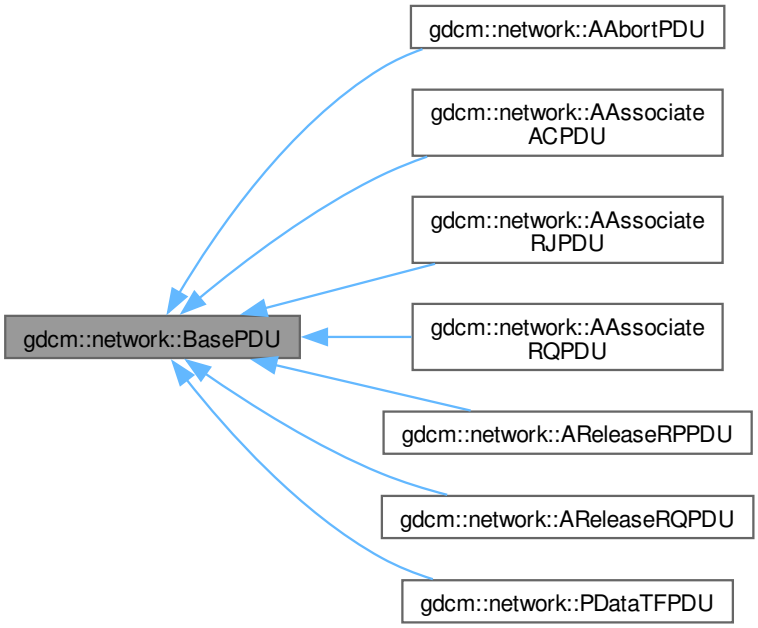
- gdcmBaseNormalizedMessage.h

12.30 gdcm::network::BasePDU Class Reference

BasePDU.

```
#include <gdcmBasePDU.h>
```

Inheritance diagram for gdcm::network::BasePDU:



Public Member Functions

- virtual ~BasePDU ()=default
- virtual bool IsLastFragment () const =0
- virtual void Print (std::ostream &os) const =0
- virtual std::istream & Read (std::istream &is)=0
- virtual size_t Size () const =0
- virtual const std::ostream & Write (std::ostream &os) const =0

12.30.1 Detailed Description

BasePDU.

base class for PDUs

all PDUs start with the first ten bytes as specified: 01 PDU type 02 reserved 3-6 PDU Length (unsigned) 7-10 variable

on some, 7-10 are split (7-8 as protocol version in Associate-RQ, for instance, while associate-rj splits those four bytes differently).

Also common to all the PDUs is their ability to read and write to a stream.

So, let's just get them all bunched together into one (abstract) class, shall we?

Why? 1) so that the ULEvent can have the PDU stored in it, since the event takes PDUs and not other class structures (other class structures get converted into PDUs) 2) to make reading PDUs in the event loop cleaner

12.30.2 Constructor & Destructor Documentation

12.30.2.1 ~BasePDU()

```
virtual gdcm::network::BasePDU::~BasePDU () [virtual], [default]
```

12.30.3 Member Function Documentation

12.30.3.1 IsLastFragment()

```
virtual bool gdcm::network::BasePDU::IsLastFragment () const [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

12.30.3.2 Print()

```
virtual void gdcm::network::BasePDU::Print (
            std::ostream & os) const [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

12.30.3.3 Read()

```
virtual std::istream & gdcm::network::BasePDU::Read (
            std::istream & is) [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

12.30.3.4 Size()

```
virtual size_t gdcm::network::BasePDU::Size () const  [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

12.30.3.5 Write()

```
virtual const std::ostream & gdcm::network::BasePDU::Write (
            std::ostream & os) const  [pure virtual]
```

Implemented in gdcm::network::AAbortPDU, gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AAssociateRQPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::PDataTFPDU.

The documentation for this class was generated from the following file:

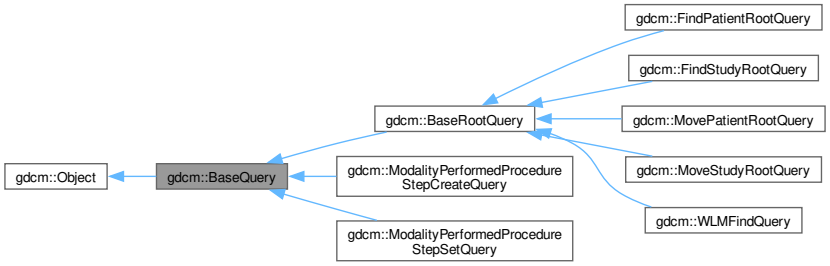
- gdcmBasePDU.h

12.31 gdcm::BaseQuery Class Reference

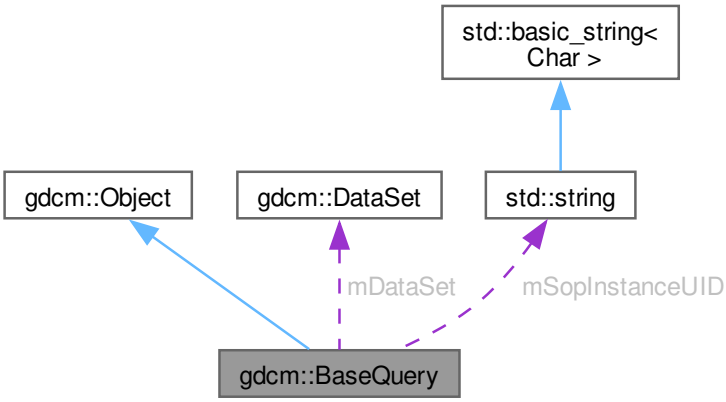
BaseQuery.

```
#include <gdcmBaseQuery.h>
```

Inheritance diagram for gdcm::BaseQuery:



Collaboration diagram for gdcm::BaseQuery:



Public Member Functions

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- virtual UIDs::TSName GetAbstractSyntaxUID () const =0
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- virtual bool ValidateQuery (bool inStrict=true) const =0
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Protected Member Functions

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes

- DataSet mDataSet
- std::string mSopInstanceUID

Friends

- class QueryFactory

12.31.1 Detailed Description

BaseQuery.

contains: a baseclass which will produce a dataset for all dimse messages

12.31.2 Constructor & Destructor Documentation

12.31.2.1 BaseQuery()

```
gdcm::BaseQuery::BaseQuery ()  [protected]
```

12.31.2.2 ∼BaseQuery()

```
gdcm::BaseQuery::∼BaseQuery ()  [override]
```

12.31.3 Member Function Documentation

12.31.3.1 AddQueryDataSet()

```
void gdcm::BaseQuery::AddQueryDataSet (
            const DataSet & ds)
```

12.31.3.2 GetAbstractSyntaxUID()

```
virtual UIDs::TSName gdcm::BaseQuery::GetAbstractSyntaxUID () const  [pure virtual]
```

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::ModalityPerformedProcedureStepCreateQuery, gdcm::ModalityPerformedProcedureStepSetQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

12.31.3.3 GetQueryDataSet() [1/2]

```
DataSet & gdcm::BaseQuery::GetQueryDataSet ()
```

12.31.3.4 GetQueryDataSet() [2/2]

```
DataSet const & gdcm::BaseQuery::GetQueryDataSet () const
```

Set/Get the internal representation of the query as a DataSet.

12.31.3.5 GetSOPInstanceUID()

```
std::string gdcm::BaseQuery::GetSOPInstanceUID () const  [inline]
```

References mSopInstanceUID.

12.31.3.6 Print()

```
void gdcm::BaseQuery::Print (
            std::ostream & os) const  [override], [virtual]
```

Reimplemented from gdcm::Object.

12.31.3.7 SetSearchParameter() [1/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const std::string & inKeyword,
            const std::string & inValue)
```

12.31.3.8 SetSearchParameter() [2/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const Tag & inTag,
            const DictEntry & inDictEntry,
            const std::string & inValue)  [protected]
```

12.31.3.9 SetSearchParameter() [3/3]

```
void gdcm::BaseQuery::SetSearchParameter (
            const Tag & inTag,
            const std::string & inValue)
```

12.31.3.10 SetSOPInstanceUID()

```
void gdcm::BaseQuery::SetSOPInstanceUID (
            const std::string & iSopInstanceUID)  [inline]
```

References mSopInstanceUID.

12.31.3.11 ValidateQuery()

```
virtual bool gdcm::BaseQuery::ValidateQuery (
            bool inStrict = true) const  [pure virtual]
```

Implemented in gdcm::BaseRootQuery, gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::ModalityPerformedProcedureS gdcm::ModalityPerformedProcedureStepSetQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

12.31.3.12 ValidDataSet()

```
bool gdcm::BaseQuery::ValidDataSet (
            const DataSet & dataSetToValid,
            const DataSet & dataSetReference) const  [protected]
```

12.31.3.13 WriteHelpFile()

```
const std::ostream & gdcm::BaseQuery::WriteHelpFile (
            std::ostream & os)
```

12.31.3.14 WriteQuery()

```
bool gdcm::BaseQuery::WriteQuery (
            const std::string & inFileName)
```

12.31.4 Friends And Related Symbol Documentation

12.31.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References QueryFactory.

Referenced by QueryFactory.

12.31.5 Member Data Documentation

12.31.5.1 mDataSet

```
DataSet gdcm::BaseQuery::mDataSet  [protected]
```

12.31.5.2 mSopInstanceUID

```
std::string gdcm::BaseQuery::mSopInstanceUID  [protected]
```

Referenced by GetSOPInstanceUID(), and SetSOPInstanceUID().

The documentation for this class was generated from the following file:

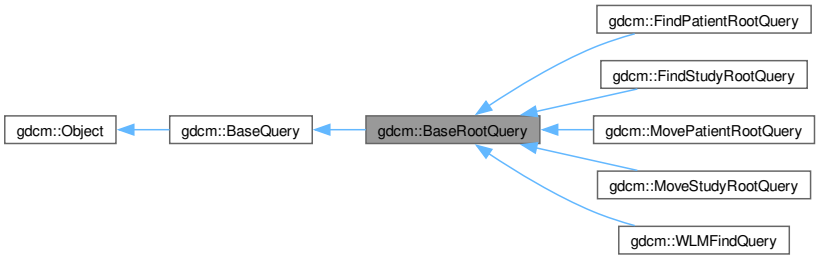
- gdcmBaseQuery.h

12.32 gdcm::BaseRootQuery Class Reference

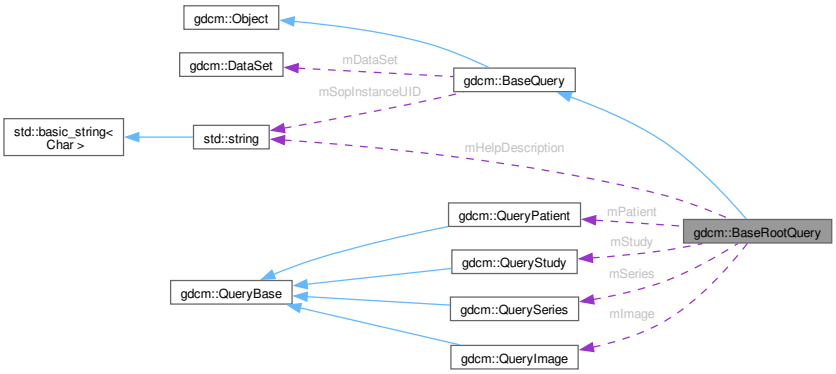
BaseRootQuery.

```
#include <gdcmBaseRootQuery.h>
```

Inheritance diagram for gdcm::BaseRootQuery:



Collaboration diagram for gdcm::BaseRootQuery:



Public Member Functions

- ~BaseRootQuery () override=default
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)
- virtual std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)=0
- virtual void InitializeDataSet (const EQueryLevel &inQueryLevel)=0
- bool ValidateQuery (bool inStrict=true) const override=0

Public Member Functions inherited from gdcm::BaseQuery

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- virtual UIDs::TSName GetAbstractSyntaxUID () const =0
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Static Public Member Functions

- static QueryBase * Construct (ERootType inRootType, EQueryLevel qlevel)
- static int GetQueryLevelFromString (const char *str)
- static const char * GetQueryLevelString (EQueryLevel ql)

Protected Member Functions

- BaseRootQuery ()

Protected Member Functions inherited from gdcm::BaseQuery

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes

- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Protected Attributes inherited from gdcm::BaseQuery

- DataSet mDataSet
- std::string mSopInstanceUID

Friends

- class QueryFactory

12.32.1 Detailed Description

BaseRootQuery.

contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root

This class contains the functionality used in patient c-find and c-move queries. PatientRootQuery and StudyRootQuery derive from this class.

Namely: 1) list all tags associated with a particular query type 2) produce a query dataset via tag association

Eventually, it can be used to validate a particular dataset type.

The dataset held by this object (or, really, one of its derivates) should be passed to a c-find or c-move query.

12.32.2 Constructor & Destructor Documentation

12.32.2.1 BaseRootQuery()

```
gdcm::BaseRootQuery::BaseRootQuery () [protected]
```

12.32.2.2 ~BaseRootQuery()

```
gdcm::BaseRootQuery::~BaseRootQuery () [override], [default]
```

12.32.3 Member Function Documentation

12.32.3.1 Construct()

```
QueryBase * gdcm::BaseRootQuery::Construct (
            ERootType inRootType,
            EQueryLevel qlevel) [static]
```

12.32.3.2 GetQueryLevelFromQueryRoot()

```
EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (
            ERootType roottype)
```

12.32.3.3 GetQueryLevelFromString()

```
int gdcm::BaseRootQuery::GetQueryLevelFromString (
            const char * str) [static]
```

12.32.3.4 GetQueryLevelString()

```
const char * gdcm::BaseRootQuery::GetQueryLevelString (
            EQueryLevel ql) [static]
```

12.32.3.5 GetTagListByLevel()

```
virtual std::vector< Tag > gdcm::BaseRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel) [pure virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

12.32.3.6 InitializeDataSet()

```
virtual void gdcm::BaseRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel)  [pure virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

12.32.3.7 ValidateQuery()

```
bool gdcm::BaseRootQuery::ValidateQuery (
            bool inStrict = true) const  [override], [pure virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseQuery.

Implemented in gdcm::FindPatientRootQuery, gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::WLMFindQuery.

12.32.4 Friends And Related Symbol Documentation

12.32.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References QueryFactory.

Referenced by QueryFactory.

12.32.5 Member Data Documentation

12.32.5.1 mHelpDescription

```
std::string gdcm::BaseRootQuery::mHelpDescription  [protected]
```

12.32.5.2 mImage

```
QueryImage gdcm::BaseRootQuery::mImage [protected]
```

12.32.5.3 mPatient

```
QueryPatient gdcm::BaseRootQuery::mPatient [protected]
```

12.32.5.4 mRootType

```
ERootType gdcm::BaseRootQuery::mRootType [protected]
```

12.32.5.5 mSeries

```
QuerySeries gdcm::BaseRootQuery::mSeries [protected]
```

12.32.5.6 mStudy

```
QueryStudy gdcm::BaseRootQuery::mStudy [protected]
```

The documentation for this class was generated from the following file:

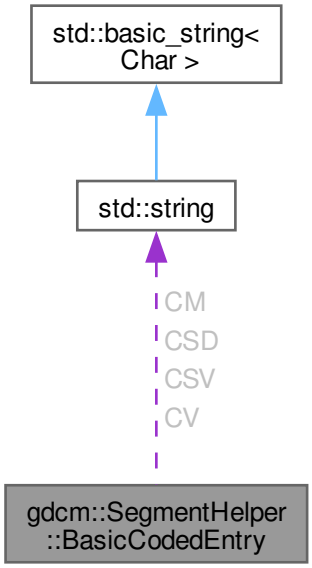
- gdcmBaseRootQuery.h

12.33 gdcm::SegmentHelper::BasicCodedEntry Struct Reference

This structure defines a basic coded entry with all of its attributes.

```
#include <gdcmSegmentHelper.h>
```

Collaboration diagram for gdcm::SegmentHelper::BasicCodedEntry:



Public Member Functions

- BasicCodedEntry ()

  Constructor.
- BasicCodedEntry (const char ∗a_CV, const char ∗a_CSD, const char ∗a_CM)

  constructor which defines type 1 attributes.
- BasicCodedEntry (const char ∗a_CV, const char ∗a_CSD, const char ∗a_CSV, const char ∗a_CM)

  constructor which defines attributes.
- bool IsEmpty (const bool checkOptionalAttributes=false) const

  Check if each attributes of the basic coded entry is defined.

Public Attributes

- std::string CM

  Coding Scheme Version attribute.
- std::string CSD

  Code Value attribute.
- std::string CSV

  Coding Scheme Designator attribute.
- std::string CV

12.33.1 Detailed Description

This structure defines a basic coded entry with all of its attributes.

See also

PS 3.3 section 8.8.

12.33.2 Constructor & Destructor Documentation

12.33.2.1 BasicCodedEntry() [1/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ()  [inline]
```

Constructor.

References CM, CSD, CSV, and CV.

12.33.2.2 BasicCodedEntry() [2/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (
            const char * a_CV,
            const char * a_CSD,
            const char * a_CM)  [inline]
```

constructor which defines type 1 attributes.

References CM, CSD, CSV, and CV.

12.33.2.3 BasicCodedEntry() [3/3]

```
gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (
            const char * a_CV,
            const char * a_CSD,
            const char * a_CSV,
            const char * a_CM)  [inline]
```

constructor which defines attributes.

References CM, CSD, CSV, and CV.

12.33.3 Member Function Documentation

12.33.3.1 IsEmpty()

```
bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (
            const bool checkOptionalAttributes = false) const
```

Check if each attributes of the basic coded entry is defined.

Parameters

checkOptionalAttributes	Check also type 1C attributes.

12.33.4 Member Data Documentation

12.33.4.1 CM

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CM
```

Coding Scheme Version attribute.

Referenced by BasicCodedEntry(), BasicCodedEntry(), and BasicCodedEntry().

12.33.4.2 CSD

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSD
```

Code Value attribute.

Referenced by BasicCodedEntry(), BasicCodedEntry(), and BasicCodedEntry().

12.33.4.3 CSV

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CSV
```

Coding Scheme Designator attribute.

Referenced by BasicCodedEntry(), BasicCodedEntry(), and BasicCodedEntry().

12.33.4.4 CV

```
std::string gdcm::SegmentHelper::BasicCodedEntry::CV
```

Referenced by BasicCodedEntry(), BasicCodedEntry(), and BasicCodedEntry().

The documentation for this struct was generated from the following file:

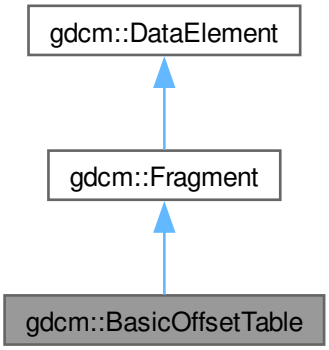
- gdcmSegmentHelper.h

12.34 gdcm::BasicOffsetTable Class Reference

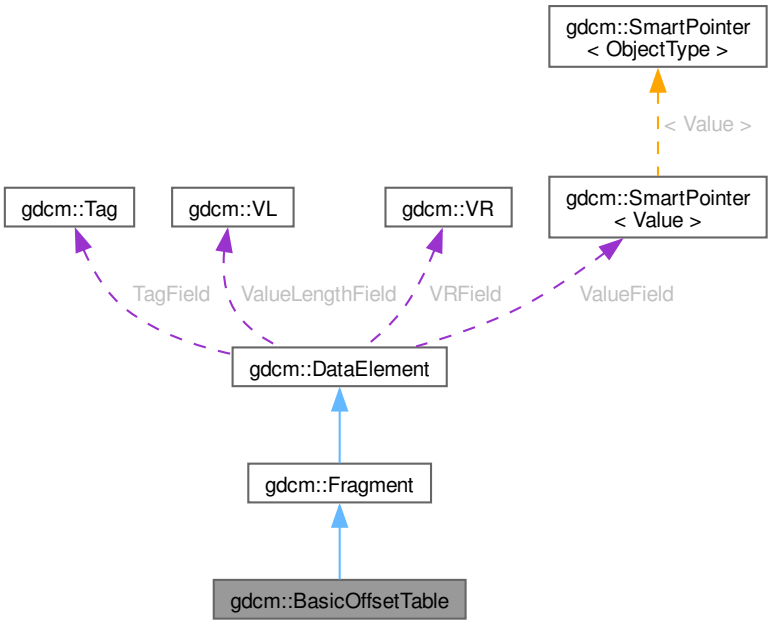
Class to represent a BasicOffsetTable.

```
#include <gdcmBasicOffsetTable.h>
```

Inheritance diagram for gdcm::BasicOffsetTable:



Collaboration diagram for gdcm::BasicOffsetTable:



Public Member Functions

- BasicOffsetTable ()
- template<typename TSwap>
  std::istream & Read (std::istream &is)

Public Member Functions inherited from gdcm::Fragment

- Fragment ()
- VL ComputeLength () const
- VL GetLength () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadBacktrack (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is)
- template<typename TSwap>
  std::ostream & Write (std::ostream &os) const

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.

- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.34.1 Detailed Description

Class to represent a BasicOffsetTable.

12.34.2 Constructor & Destructor Documentation

12.34.2.1 BasicOffsetTable()

```
gdcm::BasicOffsetTable::BasicOffsetTable () [inline]
```

References gdcm::Fragment::Fragment().

Referenced by operator<<.

12.34.3 Member Function Documentation

12.34.3.1 Read()

```
template<typename TSwap>
std::istream & gdcm::BasicOffsetTable::Read (
            std::istream & is) [inline]
```

References gdcmAssertAlwaysMacro, gdcm::ParseException::SetLastElement(), gdcm::DataElement::TagField, gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLengthField.

12.34.4 Friends And Related Symbol Documentation

12.34.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const BasicOffsetTable & val) [friend]
```

References BasicOffsetTable(), gdcm::DataElement::GetByteValue(), operator<<, gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLengthField.

Referenced by operator<<.

The documentation for this class was generated from the following file:

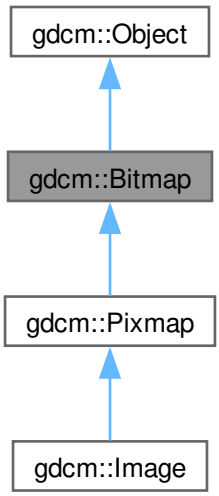
- gdcmBasicOffsetTable.h

12.35 gdcm::Bitmap Class Reference

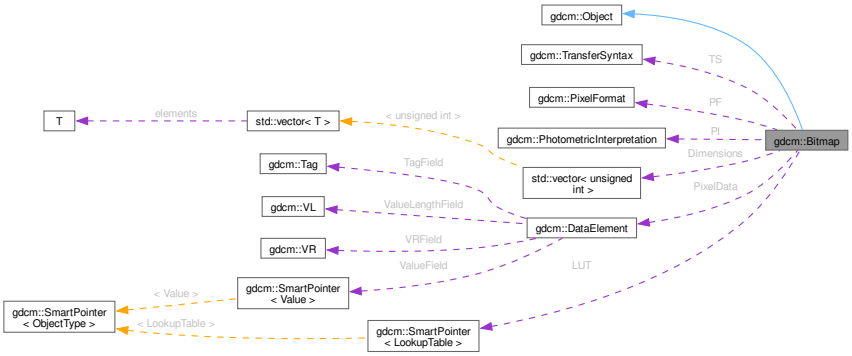
Bitmap class.

```
#include <gdcmBitmap.h>
```

Inheritance diagram for gdcm::Bitmap:



Collaboration diagram for gdcm::Bitmap:



Public Member Functions

- Bitmap ()
- ~Bitmap () override
- virtual bool AreOverlaysInPixelData () const
- void Clear ()
- bool GetBuffer (char *buffer) const

  Access the raw data.
- unsigned long GetBufferLength () const
- unsigned int GetColumns () const
- DataElement & GetDataElement ()
- const DataElement & GetDataElement () const
- unsigned int GetDimension (unsigned int idx) const
- const unsigned int * GetDimensions () const

  Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...
- LookupTable & GetLUT ()
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const

  INTERNAL do not use.
- unsigned int GetNumberOfDimensions () const

  Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.
- const PhotometricInterpretation & GetPhotometricInterpretation () const

  return the photometric interpretation
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const

  Get/Set PixelFormat.
- unsigned int GetPlanarConfiguration () const

  return the planar configuration
- unsigned int GetRows () const
- const TransferSyntax & GetTransferSyntax () const
- bool IsEmpty () const
- bool IsLossy () const

  Return whether or not the image was compressed using a lossy compressor or not.
- bool IsTransferSyntaxCompatible (TransferSyntax const &ts) const
- void Print (std::ostream &) const override
- void SetColumns (unsigned int col)
- void SetDataElement (DataElement const &de)
- void SetDimension (unsigned int idx, unsigned int dim)
- void SetDimensions (const unsigned int dims[3])
- void SetLossyFlag (bool f)

  Specifically set that the image was compressed using a lossy compression mechanism.
- void SetLUT (LookupTable const &lut)

  Set/Get LUT.
- void SetNeedByteSwap (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)
- void SetRows (unsigned int rows)
- void SetTransferSyntax (TransferSyntax const &ts)

  Transfer syntax.
- virtual bool UnusedBitsPresentInPixelData () const

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Protected Types

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions

- bool ComputeLossyFlag ()
- bool GetBuffer2 (std::ostream &os) const
- bool TryJPEG2000Codec (char *buffer, bool &lossyflag) const
- bool TryJPEG2000Codec2 (std::ostream &os) const
- bool TryJPEGCodec (char *buffer, bool &lossyflag) const
- bool TryJPEGCodec2 (std::ostream &os) const
- bool TryJPEGLSCodec (char *buffer, bool &lossyflag) const
- bool TryKAKADUCodec (char *buffer, bool &lossyflag) const
- bool TryPVRGCodec (char *buffer, bool &lossyflag) const
- bool TryRAWCodec (char *buffer, bool &lossyflag) const
- bool TryRLECodec (char *buffer, bool &lossyflag) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes

- std::vector< unsigned int > Dimensions
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- DataElement PixelData
- unsigned int PlanarConfiguration
- TransferSyntax TS

Friends

- class ImageChangeTransferSyntax
- class PixmapReader

12.35.1 Detailed Description

Bitmap class.

A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

12.35.2 Member Typedef Documentation

12.35.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr [protected]
```

12.35.3 Constructor & Destructor Documentation

12.35.3.1 Bitmap()

```
gdcm::Bitmap::Bitmap ()
```

12.35.3.2 ~Bitmap()

```
gdcm::Bitmap::~Bitmap () [override]
```

12.35.4 Member Function Documentation

12.35.4.1 AreOverlaysInPixelData()

```
virtual bool gdcm::Bitmap::AreOverlaysInPixelData () const [inline], [virtual]
```

Reimplemented in gdcm::Pixmap.

12.35.4.2 Clear()

```
void gdcm::Bitmap::Clear ()
```

12.35.4.3 ComputeLossyFlag()

```
bool gdcm::Bitmap::ComputeLossyFlag () [protected]
```

12.35.4.4 GetBuffer()

```
bool gdcm::Bitmap::GetBuffer (
            char * buffer) const
```

Access the raw data.

Examples

BasicImageAnonymizer.cs, ConvertToQImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, GetArray.cs, ReadMultiTimesException.cxx, and threadgdcm.cxx.

12.35.4.5 GetBuffer2()

```
bool gdcm::Bitmap::GetBuffer2 (
            std::ostream & os) const [protected]
```

12.35.4.6 GetBufferLength()

```
unsigned long gdcm::Bitmap::GetBufferLength () const
```

Return the length of the image after decompression WARNING for palette color: It will NOT take into account the Palette Color thus you need to multiply this length by 3 if computing the size of equivalent RGB image

Examples

BasicImageAnonymizer.cs, ConvertToQImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, GetArray.cs, PatchFile.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, and threadgdcm.cxx.

12.35.4.7 GetColumns()

```
unsigned int gdcm::Bitmap::GetColumns () const [inline]
```

References GetDimension().

12.35.4.8 GetDataElement() [1/2]

```
DataElement & gdcm::Bitmap::GetDataElement () [inline]
```

References PixelData.

12.35.4.9 GetDataElement() [2/2]

```
const DataElement & gdcm::Bitmap::GetDataElement () const  [inline]
```

Examples

ExtractIconFromFile.cxx.

References PixelData.

12.35.4.10 GetDimension()

```
unsigned int gdcm::Bitmap::GetDimension (
            unsigned int idx) const
```

Examples

BasicImageAnonymizer.cs, DecompressImage.cs, and GetArray.cs.

Referenced by GetColumns(), and GetRows().

12.35.4.11 GetDimensions()

```
const unsigned int * gdcm::Bitmap::GetDimensions () const
```

Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...

Examples

ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, HelloVizWorld.cxx, and threadgdcm.cxx.

12.35.4.12 GetLUT() [1/2]

```
LookupTable & gdcm::Bitmap::GetLUT ()  [inline]
```

References LUT.

12.35.4.13 GetLUT() [2/2]

```
const LookupTable & gdcm::Bitmap::GetLUT () const  [inline]
```

Examples

ExtractIconFromFile.cxx, ExtractImageRegionWithLUT.cs, and PrintLUT.cxx.

References LUT.

12.35.4.14 GetNeedByteSwap()

```
bool gdcm::Bitmap::GetNeedByteSwap () const  [inline]
```

INTERNAL do not use.

References NeedByteSwap.

12.35.4.15 GetNumberOfDimensions()

```
unsigned int gdcm::Bitmap::GetNumberOfDimensions () const
```

Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.

Examples

DecompressImage.cs, GetArray.cs, HelloVizWorld.cxx, and threadgdcm.cxx.

12.35.4.16 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::Bitmap::GetPhotometricInterpretation () const
```

return the photometric interpretation

Examples

ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, and HelloVizWorld.cxx.

12.35.4.17 GetPixelFormat() [1/2]

```
PixelFormat & gdcm::Bitmap::GetPixelFormat ()  [inline]
```

References PF.

12.35.4.18 GetPixelFormat() [2/2]

```
const PixelFormat & gdcm::Bitmap::GetPixelFormat () const  [inline]
```

Get/Set PixelFormat.

Examples

ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, RescaleImage.cs, TemplateEmptyImage.cxx, and threadgdcm.cxx.

References PF.

12.35.4.19 GetPlanarConfiguration()

```
unsigned int gdcm::Bitmap::GetPlanarConfiguration () const
```

return the planar configuration

12.35.4.20 GetRows()

```
unsigned int gdcm::Bitmap::GetRows () const  [inline]
```

References GetDimension().

12.35.4.21 GetTransferSyntax()

```
const TransferSyntax & gdcm::Bitmap::GetTransferSyntax () const  [inline]
```

Examples

ExtractIconFromFile.cxx.

References TS.

12.35.4.22 IsEmpty()

```
bool gdcm::Bitmap::IsEmpty () const  [inline]
```

References Dimensions.

12.35.4.23 IsLossy()

```
bool gdcm::Bitmap::IsLossy () const
```

Return whether or not the image was compressed using a lossy compressor or not.

12.35.4.24 IsTransferSyntaxCompatible()

```
bool gdcm::Bitmap::IsTransferSyntaxCompatible (
            TransferSyntax const & ts) const
```

12.35.4.25 Print()

```
void gdcm::Bitmap::Print (
            std::ostream & ) const  [override], [virtual]
```

Reimplemented from gdcm::Object.

Reimplemented in gdcm::Image, and gdcm::Pixmap.

Examples

ExtractIconFromFile.cxx.

12.35.4.26 SetColumns()

```
void gdcm::Bitmap::SetColumns (
            unsigned int col)  [inline]
```

References SetDimension().

12.35.4.27 SetDataElement()

```
void gdcm::Bitmap::SetDataElement (
            DataElement const & de)  [inline]
```

Examples

BasicImageAnonymizer.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.c DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

References PixelData.

12.35.4.28 SetDimension()

```
void gdcm::Bitmap::SetDimension (
            unsigned int idx,
            unsigned int dim)
```

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

Referenced by SetColumns(), and SetRows().

12.35.4.29 SetDimensions()

```
void gdcm::Bitmap::SetDimensions (
            const unsigned int dims[3])
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, and DecompressImage.cs.

12.35.4.30 SetLossyFlag()

```
void gdcm::Bitmap::SetLossyFlag (
            bool f)  [inline]
```

Specifically set that the image was compressed using a lossy compression mechanism.

References LossyFlag.

12.35.4.31 SetLUT()

```
void gdcm::Bitmap::SetLUT (
            LookupTable const & lut)  [inline]
```

Set/Get LUT.

References LUT, and gdcm::Object::SmartPointer.

12.35.4.32 SetNeedByteSwap()

```
void gdcm::Bitmap::SetNeedByteSwap (
            bool b)  [inline]
```

References NeedByteSwap.

12.35.4.33 SetNumberOfDimensions()

```
void gdcm::Bitmap::SetNumberOfDimensions (
            unsigned int dim)
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

12.35.4.34 SetPhotometricInterpretation()

```
void gdcm::Bitmap::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi)
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

12.35.4.35 SetPixelFormat()

```
void gdcm::Bitmap::SetPixelFormat (
            PixelFormat const & pf)  [inline]
```

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

References PF.

12.35.4.36 SetPlanarConfiguration()

```
void gdcm::Bitmap::SetPlanarConfiguration (
            unsigned int pc)
```

Warning

you need to call SetPixelFormat first (before SetPlanarConfiguration) for consistency checking

12.35.4.37 SetRows()

```
void gdcm::Bitmap::SetRows (
            unsigned int rows)  [inline]
```

References SetDimension().

12.35.4.38 SetTransferSyntax()

```
void gdcm::Bitmap::SetTransferSyntax (
            TransferSyntax const & ts)  [inline]
```

Transfer syntax.

Examples

BasicImageAnonymizer.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, MergeTwoFiles.cxx, and MpegVideoInfo.cs.

References TS.

12.35.4.39 TryJPEG2000Codec()

```
bool gdcm::Bitmap::TryJPEG2000Codec (
            char * buffer,
            bool & lossyflag) const  [protected]
```

12.35.4.40 TryJPEG2000Codec2()

```
bool gdcm::Bitmap::TryJPEG2000Codec2 (
            std::ostream & os) const  [protected]
```

12.35.4.41 TryJPEGCodec()

```
bool gdcm::Bitmap::TryJPEGCodec (
            char * buffer,
            bool & lossyflag) const  [protected]
```

12.35.4.42 TryJPEGCodec2()

```
bool gdcm::Bitmap::TryJPEGCodec2 (
            std::ostream & os) const  [protected]
```

12.35.4.43 TryJPEGLSCodec()

```
bool gdcm::Bitmap::TryJPEGLSCodec (
            char * buffer,
            bool & lossyflag) const  [protected]
```

12.35.4.44 TryKAKADUCodec()

```
bool gdcm::Bitmap::TryKAKADUCodec (
            char * buffer,
            bool & lossyflag) const  [protected]
```

12.35.4.45 TryPVRGCodec()

```
bool gdcm::Bitmap::TryPVRGCodec (
            char * buffer,
            bool & lossyflag) const  [protected]
```

12.35.4.46 TryRAWCodec()

```
bool gdcm::Bitmap::TryRAWCodec (
            char * buffer,
            bool & lossyflag) const  [protected]
```

12.35.4.47 TryRLECodec()

```
bool gdcm::Bitmap::TryRLECodec (
            char * buffer,
            bool & lossyflag) const  [protected]
```

12.35.4.48 UnusedBitsPresentInPixelData()

```
virtual bool gdcm::Bitmap::UnusedBitsPresentInPixelData () const  [inline], [virtual]
```

Reimplemented in gdcm::Pixmap.

12.35.5 Friends And Related Symbol Documentation

12.35.5.1 ImageChangeTransferSyntax

```
friend class ImageChangeTransferSyntax  [friend]
```

References ImageChangeTransferSyntax.

Referenced by ImageChangeTransferSyntax.

12.35.5.2 PixmapReader

```
friend class PixmapReader [friend]
```

References PixmapReader.

Referenced by PixmapReader.

12.35.6 Member Data Documentation

12.35.6.1 Dimensions

```
std::vector<unsigned int> gdcm::Bitmap::Dimensions [protected]
```

Referenced by IsEmpty().

12.35.6.2 LossyFlag

```
bool gdcm::Bitmap::LossyFlag [protected]
```

Referenced by SetLossyFlag().

12.35.6.3 LUT

```
LUTPtr gdcm::Bitmap::LUT [protected]
```

Referenced by GetLUT(), GetLUT(), and SetLUT().

12.35.6.4 NeedByteSwap

```
bool gdcm::Bitmap::NeedByteSwap [protected]
```

Referenced by GetNeedByteSwap(), and SetNeedByteSwap().

12.35.6.5 NumberOfDimensions

```
unsigned int gdcm::Bitmap::NumberOfDimensions [protected]
```

12.35.6.6 PF

```
PixelFormat gdcm::Bitmap::PF [protected]
```

Referenced by GetPixelFormat(), GetPixelFormat(), and SetPixelFormat().

12.35.6.7 PI

```
PhotometricInterpretation gdcm::Bitmap::PI  [protected]
```

12.35.6.8 PixelData

```
DataElement gdcm::Bitmap::PixelData  [protected]
```

Referenced by GetDataElement(), GetDataElement(), and SetDataElement().

12.35.6.9 PlanarConfiguration

```
unsigned int gdcm::Bitmap::PlanarConfiguration  [protected]
```

12.35.6.10 TS

```
TransferSyntax gdcm::Bitmap::TS  [protected]
```

Referenced by GetTransferSyntax(), and SetTransferSyntax().

The documentation for this class was generated from the following file:

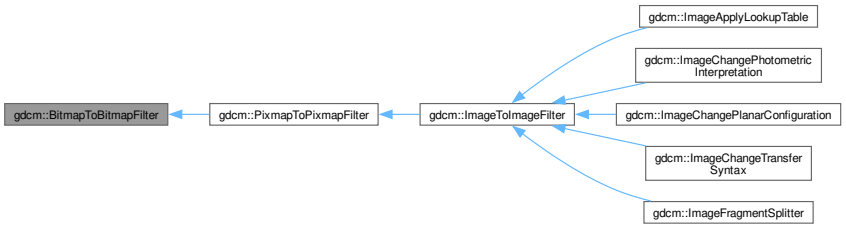
- gdcmBitmap.h

12.36 gdcm::BitmapToBitmapFilter Class Reference

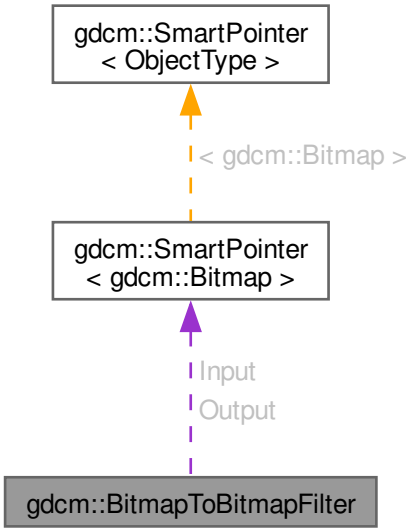
BitmapToBitmapFilter class.

```
#include <gdcmBitmapToBitmapFilter.h>
```

Inheritance diagram for gdcm::BitmapToBitmapFilter:



Collaboration diagram for gdcm::BitmapToBitmapFilter:



Public Member Functions

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Protected Attributes

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

12.36.1 Detailed Description

BitmapToBitmapFilter class.

Super class for all filter taking an image and producing an output image

12.36.2 Constructor & Destructor Documentation

12.36.2.1 BitmapToBitmapFilter()

```
gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ()
```

12.36.2.2 ~BitmapToBitmapFilter()

```
gdcm::BitmapToBitmapFilter::~BitmapToBitmapFilter ()  [default]
```

12.36.3 Member Function Documentation

12.36.3.1 GetOutput()

```
const Bitmap & gdcm::BitmapToBitmapFilter::GetOutput () const  [inline]
```

Get Output image.

References Output.

12.36.3.2 GetOutputAsBitmap()

```
const Bitmap & gdcm::BitmapToBitmapFilter::GetOutputAsBitmap () const
```

12.36.3.3 SetInput()

```
void gdcm::BitmapToBitmapFilter::SetInput (
            const Bitmap & image)
```

Set input image.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, and StandardizeFiles.cs.

12.36.4 Member Data Documentation

12.36.4.1 Input

```
SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input  [protected]
```

12.36.4.2 Output

```
SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output  [protected]
```

Referenced by GetOutput().

The documentation for this class was generated from the following file:

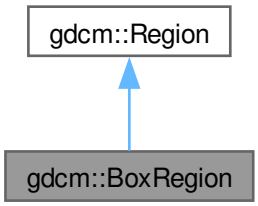
- gdcmBitmapToBitmapFilter.h

12.37 gdcm::BoxRegion Class Reference

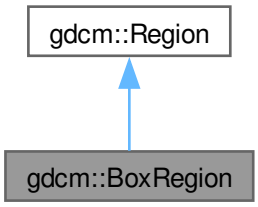
Class for manipulation box region.

```
#include <gdcmBoxRegion.h>
```

Inheritance diagram for gdcm::BoxRegion:



Collaboration diagram for gdcm::BoxRegion:



Public Member Functions

- BoxRegion ()
- BoxRegion (const BoxRegion &)

  copy/cstor and al.
- ~BoxRegion () override
- size_t Area () const override

  compute the area
- Region * Clone () const override
- BoxRegion ComputeBoundingBox () override

  Return the Axis-Aligned minimum bounding box for all regions.
- bool Empty () const override

  return whether this domain is empty:
- unsigned int GetXMax () const
- unsigned int GetXMin () const

  Get domain.
- unsigned int GetYMax () const
- unsigned int GetYMin () const
- unsigned int GetZMax () const
- unsigned int GetZMin () const
- bool IsValid () const override

  return whether this is valid domain
- void operator= (const BoxRegion &)
- void Print (std::ostream &os=std::cout) const override

  Print.
- void SetDomain (unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax)

  Set domain.

Public Member Functions inherited from gdcm::Region

- Region ()
- virtual ~Region ()

Static Public Member Functions

- static BoxRegion BoundingBox (BoxRegion const &b1, BoxRegion const &b2)

  Helper class to compute the bounding box of two BoxRegion.

12.37.1 Detailed Description

Class for manipulation box region.

This is a very simple implementation of the Region class. It only support 3D box type region. It assumes the 3D Box does not have a tilt Origin is as (0,0,0)

Examples

ExtractImageRegion.cs, and ExtractImageRegionWithLUT.cs.

12.37.2 Constructor & Destructor Documentation

12.37.2.1 BoxRegion() [1/2]

```
gdcm::BoxRegion::BoxRegion ()
```

Referenced by BoxRegion(), BoundingBox(), ComputeBoundingBox(), and operator=().

12.37.2.2 ~BoxRegion()

```
gdcm::BoxRegion::~BoxRegion ()  [override]
```

12.37.2.3 BoxRegion() [2/2]

```
gdcm::BoxRegion::BoxRegion (
            const BoxRegion & )
```

copy/cstor and al.

References BoxRegion().

12.37.3 Member Function Documentation

12.37.3.1 Area()

```
size_t gdcm::BoxRegion::Area () const  [override], [virtual]
```

compute the area

Implements gdcm::Region.

12.37.3.2 BoundingBox()

```
BoxRegion gdcm::BoxRegion::BoundingBox (
            BoxRegion const & b1,
            BoxRegion const & b2)  [static]
```

Helper class to compute the bounding box of two BoxRegion.

References BoxRegion().

12.37.3.3 Clone()

```
Region * gdcm::BoxRegion::Clone () const [override], [virtual]
```

Implements gdcm::Region.

References gdcm::Region::Region().

12.37.3.4 ComputeBoundingBox()

```
BoxRegion gdcm::BoxRegion::ComputeBoundingBox () [override], [virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implements gdcm::Region.

References BoxRegion().

12.37.3.5 Empty()

```
bool gdcm::BoxRegion::Empty () const [override], [virtual]
```

return whether this domain is empty:

Implements gdcm::Region.

12.37.3.6 GetXMax()

```
unsigned int gdcm::BoxRegion::GetXMax () const
```

12.37.3.7 GetXMin()

```
unsigned int gdcm::BoxRegion::GetXMin () const
```

Get domain.

12.37.3.8 GetYMax()

```
unsigned int gdcm::BoxRegion::GetYMax () const
```

12.37.3.9 GetYMin()

```
unsigned int gdcm::BoxRegion::GetYMin () const
```

12.37.3.10 GetZMax()

```
unsigned int gdcm::BoxRegion::GetZMax () const
```

12.37.3.11 GetZMin()

```
unsigned int gdcm::BoxRegion::GetZMin () const
```

12.37.3.12 IsValid()

```
bool gdcm::BoxRegion::IsValid () const  [override], [virtual]
```

return whether this is valid domain

Implements gdcm::Region.

12.37.3.13 operator=()

```
void gdcm::BoxRegion::operator= (
            const BoxRegion & )
```

References BoxRegion().

12.37.3.14 Print()

```
void gdcm::BoxRegion::Print (
            std::ostream & os = std::cout) const  [override], [virtual]
```

Print.

Reimplemented from gdcm::Region.

12.37.3.15 SetDomain()

```
void gdcm::BoxRegion::SetDomain (
            unsigned int xmin,
            unsigned int xmax,
            unsigned int ymin,
            unsigned int ymax,
            unsigned int zmin,
            unsigned int zmax)
```

Set domain.

Examples

ExtractImageRegion.cs, and ExtractImageRegionWithLUT.cs.

The documentation for this class was generated from the following file:

- gdcmBoxRegion.h

12.38 gdcm::ByteBuffer Class Reference

ByteBuffer.

```
#include <gdcmByteBuffer.h>
```

Public Member Functions

- ByteBuffer ()
- char * Get (int len)
- const char * GetStart () const
- void ShiftEnd (int len)
- void UpdatePosition ()

12.38.1 Detailed Description

ByteBuffer.

Detailed description here

Note

looks like a std::streambuf or std::filebuf class with the get and peek pointer

12.38.2 Constructor & Destructor Documentation

12.38.2.1 ByteBuffer()

```
gdcm::ByteBuffer::ByteBuffer () [inline]
```

12.38.3 Member Function Documentation

12.38.3.1 Get()

```
char * gdcm::ByteBuffer::Get (
            int len) [inline]
```

12.38.3.2 GetStart()

```
const char * gdcm::ByteBuffer::GetStart () const [inline]
```

12.38.3.3 ShiftEnd()

```
void gdcm::ByteBuffer::ShiftEnd (
            int len)  [inline]
```

12.38.3.4 UpdatePosition()

```
void gdcm::ByteBuffer::UpdatePosition ()  [inline]
```

The documentation for this class was generated from the following file:

- gdcmByteBuffer.h

12.39 gdcm::ByteSwap< T > Class Template Reference

ByteSwap.

```
#include <gdcmByteSwap.h>
```

Static Public Member Functions

- static void Swap (T &p)
- static void SwapFromSwapCodeIntoSystem (T &p, SwapCode const &sc)
- static void SwapRange (T *p, unsigned int num)
- static void SwapRangeFromSwapCodeIntoSystem (T *p, SwapCode const &sc, std::streamoff num)
- static bool SystemIsBigEndian ()
- static bool SystemIsLittleEndian ()

12.39.1 Detailed Description

template<class T>
class gdcm::ByteSwap< T >

ByteSwap.

Perform machine dependent byte swapping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian). TODO: bswap_32 / bswap_64 ...

12.39.2 Member Function Documentation

12.39.2.1 Swap()

```
template<class T>
void gdcm::ByteSwap< T >::Swap (
            T & p)  [static]
```

12.39.2.2 SwapFromSwapCodeIntoSystem()

```
template<class T>
void gdcm::ByteSwap< T >::SwapFromSwapCodeIntoSystem (
            T & p,
            SwapCode const & sc)  [static]
```

Examples

TestByteSwap.cxx.

12.39.2.3 SwapRange()

```
template<class T>
void gdcm::ByteSwap< T >::SwapRange (
            T * p,
            unsigned int num)  [static]
```

12.39.2.4 SwapRangeFromSwapCodeIntoSystem()

```
template<class T>
void gdcm::ByteSwap< T >::SwapRangeFromSwapCodeIntoSystem (
            T * p,
            SwapCode const & sc,
            std::streamoff num)  [static]
```

Examples

TestByteSwap.cxx.

12.39.2.5 SystemIsBigEndian()

```
template<class T>
bool gdcm::ByteSwap< T >::SystemIsBigEndian ()  [static]
```

Query the machine Endian-ness.

Examples

TestByteSwap.cxx.

12.39.2.6 SystemIsLittleEndian()

```
template<class T>
bool gdcm::ByteSwap< T >::SystemIsLittleEndian () [static]
```

Examples

TestByteSwap.cxx.

The documentation for this class was generated from the following file:

- gdcmByteSwap.h

12.40 gdcm::ByteSwapFilter Class Reference

ByteSwapFilter.

```
#include <gdcmByteSwapFilter.h>
```

Public Member Functions

- ByteSwapFilter (const ByteSwapFilter &)=delete
- ByteSwapFilter (DataSet &ds)
- ~ByteSwapFilter ()=default
- bool ByteSwap ()
- ByteSwapFilter & operator= (const ByteSwapFilter &)=delete
- void SetByteSwapTag (bool b)

12.40.1 Detailed Description

ByteSwapFilter.

In place byte-swapping of a dataset FIXME: FL status ??

12.40.2 Constructor & Destructor Documentation

12.40.2.1 ByteSwapFilter() [1/2]

```
gdcm::ByteSwapFilter::ByteSwapFilter (
            DataSet & ds) [inline]
```

Referenced by ByteSwapFilter(), and operator=().

12.40.2.2 ~ByteSwapFilter()

```
gdcm::ByteSwapFilter::~ByteSwapFilter ()  [default]
```

12.40.2.3 ByteSwapFilter() [2/2]

```
gdcm::ByteSwapFilter::ByteSwapFilter (
            const ByteSwapFilter & )  [delete]
```

References ByteSwapFilter().

12.40.3 Member Function Documentation

12.40.3.1 ByteSwap()

```
bool gdcm::ByteSwapFilter::ByteSwap ()
```

Referenced by gdcm::Item::Read().

12.40.3.2 operator=()

```
ByteSwapFilter & gdcm::ByteSwapFilter::operator= (
            const ByteSwapFilter & )  [delete]
```

References ByteSwapFilter().

12.40.3.3 SetByteSwapTag()

```
void gdcm::ByteSwapFilter::SetByteSwapTag (
            bool b)  [inline]
```

Referenced by gdcm::Item::Read().

The documentation for this class was generated from the following file:

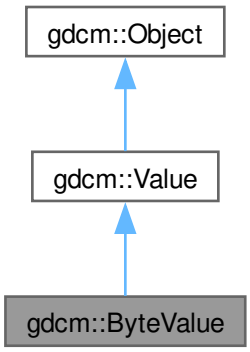
- gdcmByteSwapFilter.h

12.41 gdcm::ByteValue Class Reference

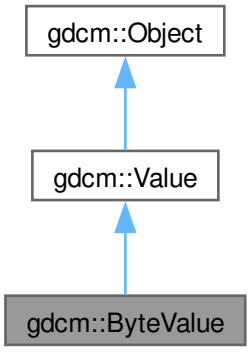
Class to represent binary value (array of bytes).

```
#include <gdcmByteValue.h>
```

Inheritance diagram for gdcm::ByteValue:



Collaboration diagram for gdcm::ByteValue:



Public Member Functions

- ByteValue (const char *array=nullptr, VL const &vl=0)
- ByteValue (std::vector< char > &v)
- ~ByteValue () override
- void Append (ByteValue const &bv)
- void Clear () override
- VL ComputeLength () const
- void Fill (char c)
- bool GetBuffer (char *buffer, unsigned long length) const
- VL GetLength () const override
- const char * GetPointer () const
- void * GetVoidPointer ()
- const void * GetVoidPointer () const
- bool IsEmpty () const
- bool IsPrintable (VL length) const

  Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I don't think this function is working since it does not handle UNICODE or character set...
- operator const std::vector< char > & () const
- ByteValue & operator= (const ByteValue &val)
- bool operator== (const ByteValue &val) const
- bool operator== (const Value &val) const override
- void PrintASCII (std::ostream &os, VL maxlength) const
- void PrintASCIIXML (std::ostream &os) const
- void PrintGroupLength (std::ostream &os)
- void PrintHex (std::ostream &os, VL maxlength) const
- void PrintHexXML (std::ostream &os) const
- void PrintPNXML (std::ostream &os) const
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap, typename TType>
  std::istream & Read (std::istream &is, bool readvalues=true)
- void SetLength (VL vl) override
- template<typename TSwap>
  std::ostream const & Write (std::ostream &os) const
- template<typename TSwap, typename TType>
  std::ostream const & Write (std::ostream &os) const
- bool WriteBuffer (std::ostream &os) const

Public Member Functions inherited from gdcm::Value

- Value ()=default
- ~Value () override=default

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Protected Member Functions

- void Print (std::ostream &os) const override
- void SetLengthOnly (VL vl) override

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.41.1 Detailed Description

Class to represent binary value (array of bytes).

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, ReadExplicitLengthSQIVR.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.41.2 Constructor & Destructor Documentation

12.41.2.1 ByteValue() [1/2]

```
gdcm::ByteValue::ByteValue (
            const char * array = nullptr,
            VL const & vl = 0)  [inline]
```

Referenced by Append(), operator=(), operator==(), and operator==().

12.41.2.2 ByteValue() [2/2]

```
gdcm::ByteValue::ByteValue (
            std::vector< char > & v)  [inline]
```

Warning

casting to uint32_t

12.41.2.3 ~ByteValue()

```
gdcm::ByteValue::~ByteValue ()  [inline], [override]
```

12.41.3 Member Function Documentation

12.41.3.1 Append()

```
void gdcm::ByteValue::Append (
            ByteValue const & bv)
```

References ByteValue().

12.41.3.2 Clear()

```
void gdcm::ByteValue::Clear () [inline], [override], [virtual]
```

Implements gdcm::Value.

12.41.3.3 ComputeLength()

```
VL gdcm::ByteValue::ComputeLength () const [inline]
```

Referenced by gdcm::Fragment::Write().

12.41.3.4 Fill()

```
void gdcm::ByteValue::Fill (
            char c) [inline]
```

Examples

DuplicatePCDE.cxx.

12.41.3.5 GetBuffer()

```
bool gdcm::ByteValue::GetBuffer (
            char * buffer,
            unsigned long length) const
```

Examples

ExtractEncapsulatedFile.cs, and FixJAIBugJPEGLS.cxx.

12.41.3.6 GetLength()

```
VL gdcm::ByteValue::GetLength () const [inline], [override], [virtual]
```

Implements gdcm::Value.

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::CSAElement::operator<<, gdcm::SequenceOfFragments::ReadValue(), gdcm::Element< TVR, TVM >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, E gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueN gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< TVR, TVM >::SetNoSwap(), gdcm::Element< TVR, VM::VM1_n >::SetNoSwap(), and gdcm::Fragment::Write().

12.41.3.7 GetPointer()

```
const char * gdcm::ByteValue::GetPointer () const [inline]
```

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx ExtractIconFromFile.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::CSAElement::operator<<, gdcm::SequenceOfFragments::ReadValue(), gdcm::Element< TVR, TVM >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, E gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueN gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< TVR, TVM >::SetNoSwap(), and gdcm::Element< TVR, VM::VM1_n >::SetNoSwap().

12.41.3.8 GetVoidPointer() [1/2]

```
void * gdcm::ByteValue::GetVoidPointer () [inline]
```

12.41.3.9 GetVoidPointer() [2/2]

```
const void * gdcm::ByteValue::GetVoidPointer () const [inline]
```

Examples

FixBrokenJ2K.cxx.

Referenced by Read(), and gdcm::Element< TVR, VM::VM1_n >::Set().

12.41.3.10 IsEmpty()

```
bool gdcm::ByteValue::IsEmpty () const  [inline]
```

12.41.3.11 IsPrintable()

```
bool gdcm::ByteValue::IsPrintable (
            VL length) const  [inline]
```

Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I don't think this function is working since it does not handle UNICODE or character set...

Referenced by Print().

12.41.3.12 operator const std::vector< char > &()

```
gdcm::ByteValue::operator const std::vector< char > & () const  [inline]
```

12.41.3.13 operator=()

```
ByteValue & gdcm::ByteValue::operator= (
            const ByteValue & val)  [inline]
```

References ByteValue().

12.41.3.14 operator==() [1/2]

```
bool gdcm::ByteValue::operator== (
            const ByteValue & val) const  [inline]
```

References ByteValue().

12.41.3.15 operator==() [2/2]

```
bool gdcm::ByteValue::operator== (
            const Value & val) const  [inline], [override], [virtual]
```

Implements gdcm::Value.

References ByteValue(), and gdcm::Value::Value().

12.41.3.16 Print()

```
void gdcm::ByteValue::Print (
            std::ostream & os) const  [inline], [override], [protected], [virtual]
```

Reimplemented from gdcm::Object.

References IsPrintable().

12.41.3.17 PrintASCII()

```
void gdcm::ByteValue::PrintASCII (
            std::ostream & os,
            VL maxlength) const
```

12.41.3.18 PrintASCIIXML()

```
void gdcm::ByteValue::PrintASCIIXML (
            std::ostream & os) const
```

12.41.3.19 PrintGroupLength()

```
void gdcm::ByteValue::PrintGroupLength (
            std::ostream & os)  [inline]
```

12.41.3.20 PrintHex()

```
void gdcm::ByteValue::PrintHex (
            std::ostream & os,
            VL maxlength) const
```

12.41.3.21 PrintHexXML()

```
void gdcm::ByteValue::PrintHexXML (
            std::ostream & os) const
```

12.41.3.22 PrintPNXML()

```
void gdcm::ByteValue::PrintPNXML (
            std::ostream & os) const
```

To Print Values in Native DICOM format

12.41.3.23 Read() [1/2]

```
template<typename TSwap>
std::istream & gdcm::ByteValue::Read (
            std::istream & is)  [inline]
```

References Read().

12.41.3.24 Read() [2/2]

```
template<typename TSwap, typename TType>
std::istream & gdcm::ByteValue::Read (
            std::istream & is,
            bool readvalues = true)  [inline]
```

References GetVoidPointer().

Referenced by Read().

12.41.3.25 SetLength()

```
void gdcm::ByteValue::SetLength (
            VL vl)  [override], [virtual]
```

Implements gdcm::Value.

12.41.3.26 SetLengthOnly()

```
void gdcm::ByteValue::SetLengthOnly (
            VL vl)  [inline], [override], [protected], [virtual]
```

Reimplemented from gdcm::Value.

12.41.3.27 Write() [1/2]

```
template<typename TSwap>
std::ostream const & gdcm::ByteValue::Write (
            std::ostream & os) const  [inline]
```

References Write().

12.41.3.28 Write() [2/2]

```
template<typename TSwap, typename TType>
std::ostream const & gdcm::ByteValue::Write (
            std::ostream & os) const  [inline]
```

Referenced by Write(), and gdcm::Fragment::Write().

12.41.3.29 WriteBuffer()

```
bool gdcm::ByteValue::WriteBuffer (
            std::ostream & os) const  [inline]
```

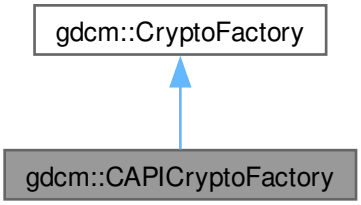
The documentation for this class was generated from the following file:

- gdcmByteValue.h

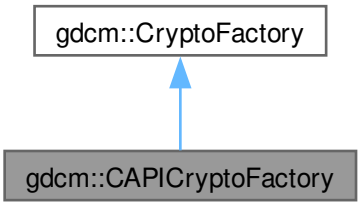
12.42 gdcm::CAPICryptoFactory Class Reference

```
#include <gdcmCAPICryptoFactory.h>
```

Inheritance diagram for gdcm::CAPICryptoFactory:



Collaboration diagram for gdcm::CAPICryptoFactory:



Public Member Functions

- CAPICryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Additional Inherited Members

Public Types inherited from gdcm::CryptoFactory

- enum CryptoLib {
  DEFAULT = 0 ,
  OPENSSL = 1 ,
  CAPI = 2 ,
  OPENSSLP7 = 3 }

Static Public Member Functions inherited from gdcm::CryptoFactory

- static CryptoFactory ∗ GetFactoryInstance (CryptoLib id=DEFAULT)

Protected Member Functions inherited from gdcm::CryptoFactory

- CryptoFactory ()=default
- CryptoFactory (CryptoLib id)
- ∼CryptoFactory ()=default

12.42.1 Constructor & Destructor Documentation

12.42.1.1 CAPICryptoFactory()

```
gdcm::CAPICryptoFactory::CAPICryptoFactory (
            CryptoLib id)
```

Referenced by CreateCMSProvider().

12.42.2 Member Function Documentation

12.42.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax * gdcm::CAPICryptoFactory::CreateCMSProvider ()  [virtual]
```

Implements gdcm::CryptoFactory.

References CAPICryptoFactory().

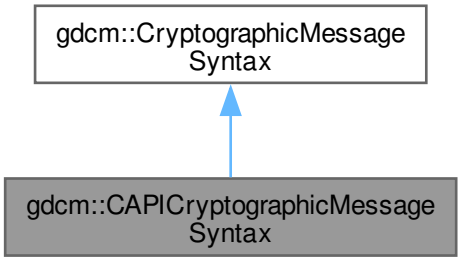
The documentation for this class was generated from the following file:

- gdcmCAPICryptoFactory.h

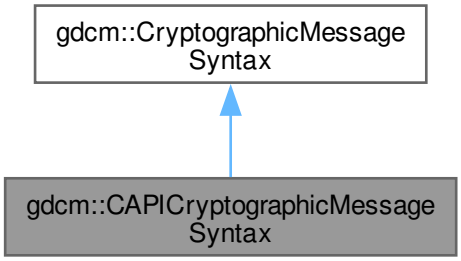
12.43 gdcm::CAPICryptographicMessageSyntax Class Reference

```
#include <gdcmCAPICryptographicMessageSyntax.h>
```

Inheritance diagram for gdcm::CAPICryptographicMessageSyntax:



Collaboration diagram for gdcm::CAPICryptographicMessageSyntax:



Public Member Functions

- CAPICryptographicMessageSyntax ()
- ~CAPICryptographicMessageSyntax ()
- bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const

  decrypt content from a CMS envelopedData structure
- bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const

  create a CMS envelopedData structure
- CipherTypes GetCipherType () const

- bool GetInitialized () const
- bool ParseCertificateFile (const char ∗filename)
- bool ParseKeyFile (const char ∗filename)
- void SetCipherType (CipherTypes type)
- bool SetPassword (const char ∗pass, size_t passLen)

Public Member Functions inherited from gdcm::CryptographicMessageSyntax

- CryptographicMessageSyntax ()=default
- CryptographicMessageSyntax (const CryptographicMessageSyntax &)=delete
- virtual ∼CryptographicMessageSyntax ()=default
- void operator= (const CryptographicMessageSyntax &)=delete

Additional Inherited Members

Public Types inherited from gdcm::CryptographicMessageSyntax

- enum CipherTypes {
  DES3_CIPHER ,
  AES128_CIPHER ,
  AES192_CIPHER ,
  AES256_CIPHER }

12.43.1 Constructor & Destructor Documentation

12.43.1.1 CAPICryptographicMessageSyntax()

```
gdcm::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax ()
```

12.43.1.2 ∼CAPICryptographicMessageSyntax()

```
gdcm::CAPICryptographicMessageSyntax::∼CAPICryptographicMessageSyntax ()
```

12.43.2 Member Function Documentation

12.43.2.1 Decrypt()

```
bool gdcm::CAPICryptographicMessageSyntax::Decrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len) const  [virtual]
```

decrypt content from a CMS envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

12.43.2.2 Encrypt()

```
bool gdcm::CAPICryptographicMessageSyntax::Encrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len) const  [virtual]
```

create a CMS envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

12.43.2.3 GetCipherType()

```
CipherTypes gdcm::CAPICryptographicMessageSyntax::GetCipherType () const  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.43.2.4 GetInitialized()

```
bool gdcm::CAPICryptographicMessageSyntax::GetInitialized () const  [inline]
```

12.43.2.5 ParseCertificateFile()

```
bool gdcm::CAPICryptographicMessageSyntax::ParseCertificateFile (
            const char * filename)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.43.2.6 ParseKeyFile()

```
bool gdcm::CAPICryptographicMessageSyntax::ParseKeyFile (
            const char * filename)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.43.2.7 SetCipherType()

```
void gdcm::CAPICryptographicMessageSyntax::SetCipherType (
            CipherTypes type)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.43.2.8 SetPassword()

```
bool gdcm::CAPICryptographicMessageSyntax::SetPassword (
            const char * pass,
            size_t passLen)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

The documentation for this class was generated from the following file:

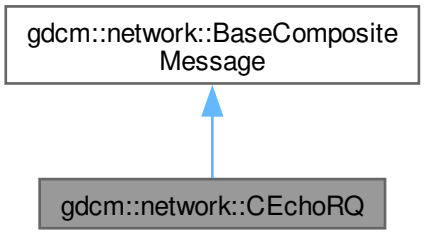
- gdcmCAPICryptographicMessageSyntax.h

12.44 gdcm::network::CEchoRQ Class Reference

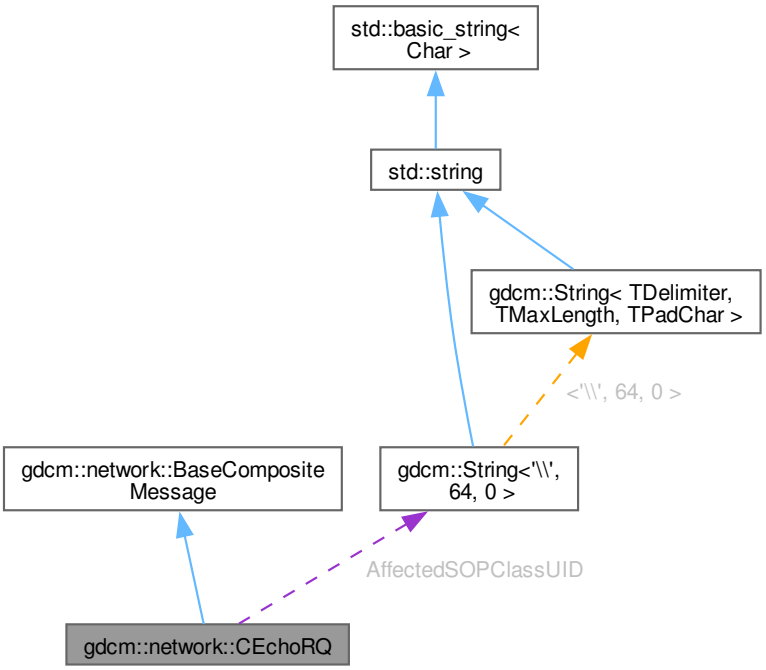
CEchoRQ.

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRQ:



Collaboration diagram for gdcm::network::CEchoRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery) override

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default

Public Attributes

- UIComp AffectedSOPClassUID
- uint16_t MessageID

12.44.1 Detailed Description

CEchoRQ.

this file defines the messages for the cecho action

12.44.2 Member Function Documentation

12.44.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CEchoRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery)  [override], [virtual]
```

Implements gdcm::network::BaseCompositeMessage.

12.44.3 Member Data Documentation

12.44.3.1 AffectedSOPClassUID

```
UIComp gdcm::network::CEchoRQ::AffectedSOPClassUID
```

12.44.3.2 MessageID

```
uint16_t gdcm::network::CEchoRQ::MessageID
```

The documentation for this class was generated from the following files:

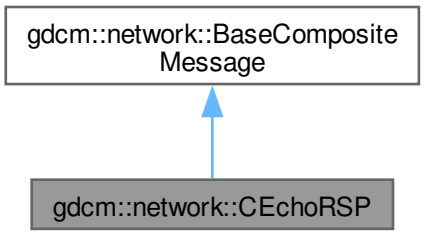
- gdcmCEchoMessages.h
- gdcmDIMSE.h

12.45 gdcm::network::CEchoRSP Class Reference

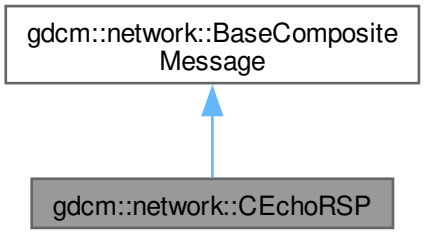
CEchoRSP this file defines the messages for the cecho action.

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for gdcm::network::CEchoRSP:



Collaboration diagram for gdcm::network::CEchoRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

12.45.1 Detailed Description

CEchoRSP this file defines the messages for the cecho action.

12.45.2 Member Function Documentation

12.45.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CEchoRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

- gdcmCEchoMessages.h

12.46 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

12.46.1 Detailed Description

PS 3.4 - 2009 Table B.2-1 C-STORE STATUS

The documentation for this class was generated from the following file:

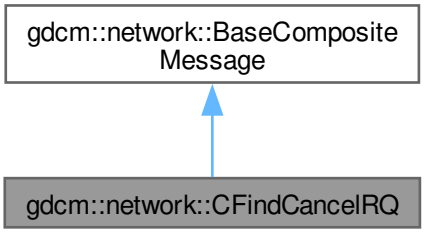
- gdcmDIMSE.h

12.47 gdcm::network::CFindCancelRQ Class Reference

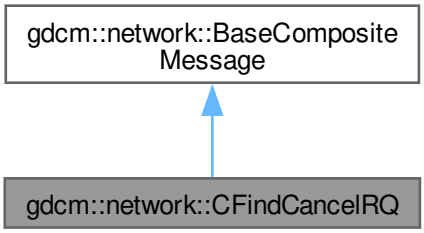
CFindCancelRQ this file defines the messages for the cfind action.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindCancelRQ:



Collaboration diagram for gdcm::network::CFindCancelRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

12.47.1 Detailed Description

CFindCancelRQ this file defines the messages for the cfind action.

12.47.2 Member Function Documentation

12.47.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CFindCancelRQ::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

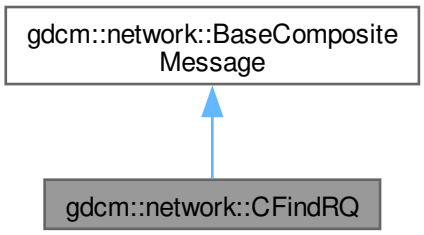
- gdcmCFindMessages.h

12.48 gdcm::network::CFindRQ Class Reference

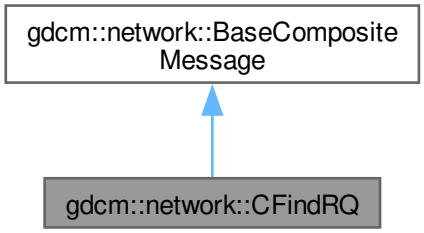
CFindRQ.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindRQ:



Collaboration diagram for gdcm::network::CFindRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery) override

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default

12.48.1 Detailed Description

CFindRQ.

this file defines the messages for the cfind action

12.48.2 Member Function Documentation

12.48.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CFindRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery)  [override], [virtual]
```

Implements gdcm::network::BaseCompositeMessage.

The documentation for this class was generated from the following file:

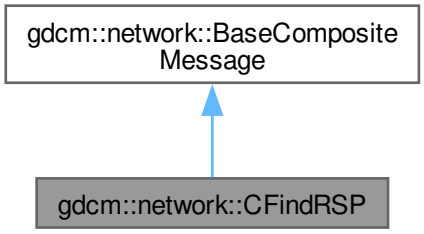
- gdcmCFindMessages.h

12.49 gdcm::network::CFindRSP Class Reference

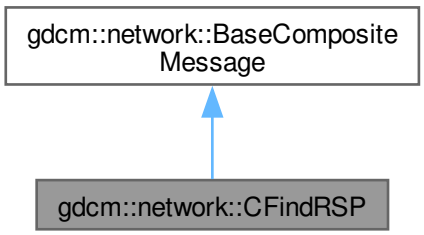
CFindRSP this file defines the messages for the cfind action.

```
#include <gdcmCFindMessages.h>
```

Inheritance diagram for gdcm::network::CFindRSP:



Collaboration diagram for gdcm::network::CFindRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

12.49.1 Detailed Description

CFindRSP this file defines the messages for the cfind action.

12.49.2 Member Function Documentation

12.49.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CFindRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

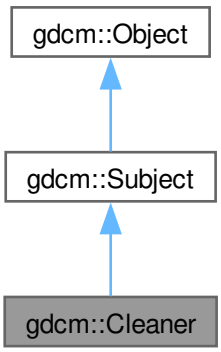
- gdcmCFindMessages.h

12.50 gdcm::Cleaner Class Reference

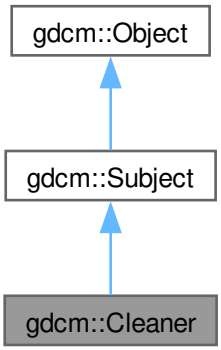
Cleaner.

```
#include <gdcmCleaner.h>
```

Inheritance diagram for gdcm::Cleaner:



Collaboration diagram for gdcm::Cleaner:



Public Member Functions

- Cleaner ()
- ~Cleaner () override
- bool Clean ()

  main loop
- bool Empty (DPath const &dpath)
- bool Empty (PrivateTag const &pt)
- bool Empty (Tag const &t)
- bool Empty (VR const &vr)
- void EmptyWhenScrubFails (bool empty)

  Should I empty instead of scrub upon failure.
- File & GetFile ()
- bool Preserve (DPath const &dpath)
- bool Remove (DPath const &dpath)
- bool Remove (PrivateTag const &pt)
- bool Remove (Tag const &t)
- bool Remove (VR const &vr)
- void RemoveAllGroupLength (bool remove)

  Should I remove all group length (deprecated). Default: true.
- void RemoveAllIllegal (bool remove)

  Should I remove all illegal attribute. Default: true.
- void RemoveAllMissingPrivateCreator (bool remove)
- bool RemoveMissingPrivateCreator (Tag const &t)
- bool Scrub (DPath const &dpath)
- bool Scrub (PrivateTag const &pt)
- bool Scrub (Tag const &t)

  Clean digital tash (typically SIEMENS CSA header):
- bool Scrub (VR const &vr)
- void SetFile (const File &f)

  Set/Get File.

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static SmartPointer< Cleaner > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.50.1 Detailed Description

Cleaner.

This class implement the Subject/Observer pattern trigger the following event:

- AnonymizeEvent
- IterationEvent
- StartEvent
- EndEvent

Examples

Cleaner.cs.

12.50.2 Constructor & Destructor Documentation

12.50.2.1 Cleaner()

```
gdcm::Cleaner::Cleaner ()
```

Referenced by New().

12.50.2.2 ~Cleaner()

```
gdcm::Cleaner::~Cleaner ()  [override]
```

12.50.3 Member Function Documentation

12.50.3.1 Clean()

```
bool gdcm::Cleaner::Clean ()
```

main loop

Examples

Cleaner.cs.

12.50.3.2 Empty() [1/4]

```
bool gdcm::Cleaner::Empty (
            DPath const & dpath)
```

12.50.3.3 Empty() [2/4]

```
bool gdcm::Cleaner::Empty (
            PrivateTag const & pt)
```

12.50.3.4 Empty() [3/4]

```
bool gdcm::Cleaner::Empty (
            Tag const & t)
```

Examples

Cleaner.cs.

12.50.3.5 Empty() [4/4]

```
bool gdcm::Cleaner::Empty (
            VR const & vr)
```

12.50.3.6 EmptyWhenScrubFails()

```
void gdcm::Cleaner::EmptyWhenScrubFails (
            bool empty)
```

Should I empty instead of scrub upon failure.

12.50.3.7 GetFile()

```
File & gdcm::Cleaner::GetFile ()  [inline]
```

Examples

Cleaner.cs.

12.50.3.8 New()

```
SmartPointer< Cleaner > gdcm::Cleaner::New ()  [inline], [static]
```

for wrapped language: instantiate a reference counted object

References Cleaner().

12.50.3.9 Preserve()

```
bool gdcm::Cleaner::Preserve (
            DPath const & dpath)
```

Examples

Cleaner.cs.

12.50.3.10 Remove() [1/4]

```
bool gdcm::Cleaner::Remove (
            DPath const & dpath)
```

12.50.3.11 Remove() [2/4]

```
bool gdcm::Cleaner::Remove (
            PrivateTag const & pt)
```

12.50.3.12 Remove() [3/4]

```
bool gdcm::Cleaner::Remove (
            Tag const & t)
```

Examples

Cleaner.cs.

12.50.3.13 Remove() [4/4]

```
bool gdcm::Cleaner::Remove (
            VR const & vr)
```

12.50.3.14 RemoveAllGroupLength()

```
void gdcm::Cleaner::RemoveAllGroupLength (
            bool remove)
```

Should I remove all group length (deprecated). Default: true.

12.50.3.15 RemoveAllIllegal()

```
void gdcm::Cleaner::RemoveAllIllegal (
            bool remove)
```

Should I remove all illegal attribute. Default: true.

12.50.3.16 RemoveAllMissingPrivateCreator()

```
void gdcm::Cleaner::RemoveAllMissingPrivateCreator (
            bool remove)
```

Should I remove all private tag for which no private creator is found. Default: true

12.50.3.17 RemoveMissingPrivateCreator()

```
bool gdcm::Cleaner::RemoveMissingPrivateCreator (
            Tag const & t)
```

Specify a private tag (odd number) without a private creator (root level only for now):

12.50.3.18 Scrub() [1/4]

```
bool gdcm::Cleaner::Scrub (
            DPath const & dpath)
```

12.50.3.19 Scrub() [2/4]

```
bool gdcm::Cleaner::Scrub (
            PrivateTag const & pt)
```

12.50.3.20 Scrub() [3/4]

```
bool gdcm::Cleaner::Scrub (
            Tag const & t)
```

Clean digital tash (typically SIEMENS CSA header):

Examples

Cleaner.cs.

12.50.3.21 Scrub() [4/4]

```
bool gdcm::Cleaner::Scrub (
            VR const & vr)
```

12.50.3.22 SetFile()

```
void gdcm::Cleaner::SetFile (
            const File & f)  [inline]
```

Set/Get File.

Examples

Cleaner.cs.

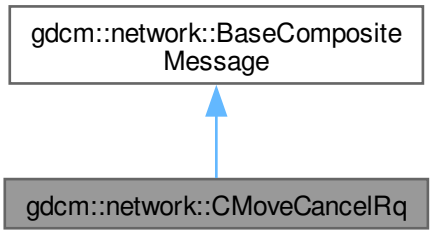
The documentation for this class was generated from the following file:

- gdcmCleaner.h

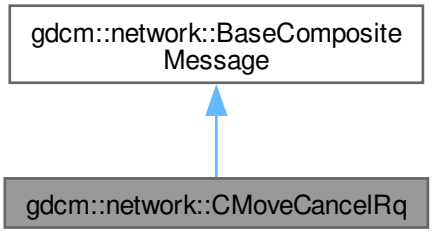
12.51 gdcm::network::CMoveCancelRq Class Reference

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveCancelRq:



Collaboration diagram for gdcm::network::CMoveCancelRq:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

12.51.1 Member Function Documentation

12.51.1.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

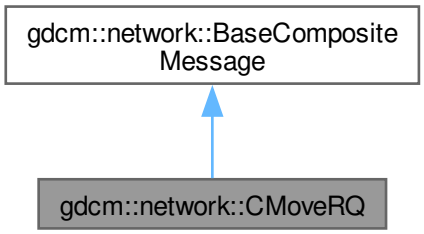
- gdcmCMoveMessages.h

12.52 gdcm::network::CMoveRQ Class Reference

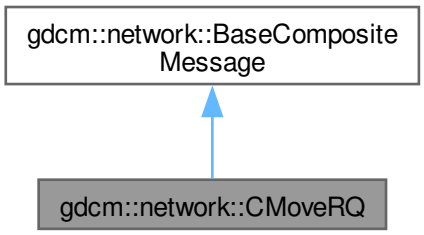
CMoveRQ.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRQ:



Collaboration diagram for gdcm::network::CMoveRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery) override

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default

12.52.1 Detailed Description

CMoveRQ.

this file defines the messages for the cmove action

12.52.2 Member Function Documentation

12.52.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CMoveRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery)  [override], [virtual]
```

Implements gdcm::network::BaseCompositeMessage.

The documentation for this class was generated from the following file:

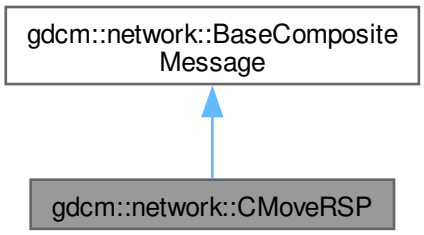
- gdcmCMoveMessages.h

12.53 gdcm::network::CMoveRSP Class Reference

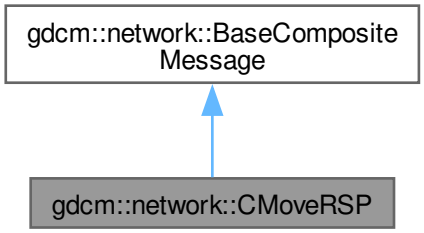
CMoveRSP this file defines the messages for the cmove action.

```
#include <gdcmCMoveMessages.h>
```

Inheritance diagram for gdcm::network::CMoveRSP:



Collaboration diagram for gdcm::network::CMoveRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

12.53.1 Detailed Description

CMoveRSP this file defines the messages for the cmove action.

12.53.2 Member Function Documentation

12.53.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::CMoveRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

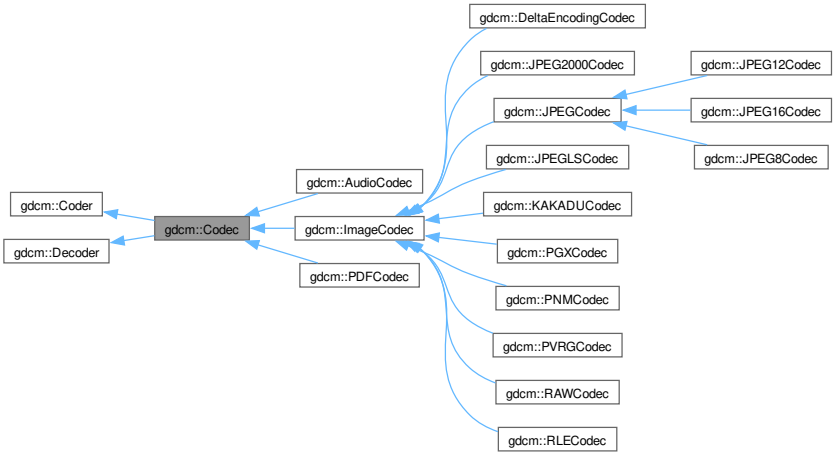
- gdcmCMoveMessages.h

12.54 gdcm::Codec Class Reference

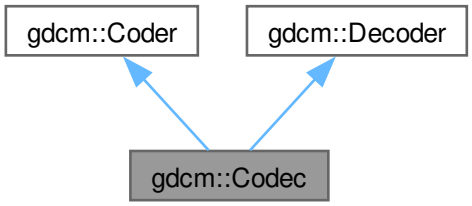
Codec class.

`#include <gdcmCodec.h>`

Inheritance diagram for gdcm::Codec:



Collaboration diagram for gdcm::Codec:



Additional Inherited Members

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default
- virtual bool CanCode (TransferSyntax const &) const =0

  Return whether this coder support this transfer syntax (can code it).
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default
- virtual bool CanDecode (TransferSyntax const &) const =0

  Return whether this decoder support this transfer syntax (can decode it).
- virtual bool Decode (DataElement const &, DataElement &)

  Decode.

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Protected Member Functions inherited from gdcm::Decoder

- virtual bool DecodeByStreams (std::istream &, std::ostream &)

12.54.1 Detailed Description

Codec class.

The documentation for this class was generated from the following file:

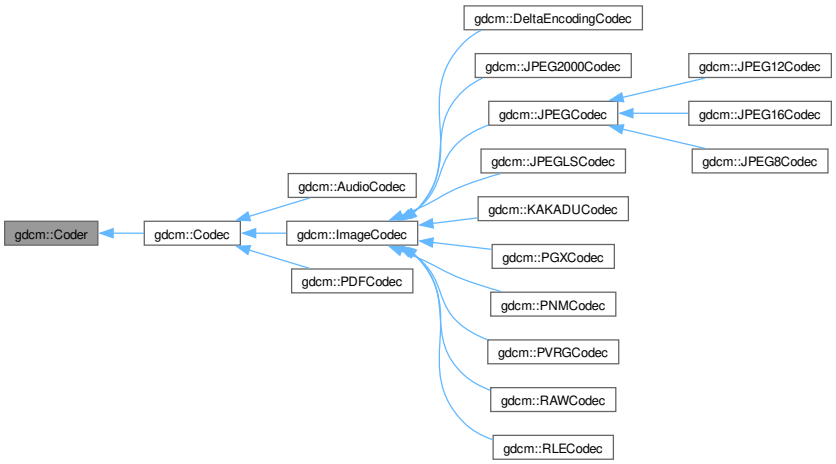
- gdcmCodec.h

12.55 gdcm::Coder Class Reference

Coder.

```
#include <gdcmCoder.h>
```

Inheritance diagram for gdcm::Coder:



Public Member Functions

- virtual ~Coder ()=default
- virtual bool CanCode (TransferSyntax const &) const =0

  Return whether this coder support this transfer syntax (can code it).
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Protected Member Functions

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

12.55.1 Detailed Description

Coder.

12.55.2 Constructor & Destructor Documentation

12.55.2.1 ~Coder()

```
virtual gdcm::Coder::~Coder ()  [virtual], [default]
```

12.55.3 Member Function Documentation

12.55.3.1 CanCode()

```
virtual bool gdcm::Coder::CanCode (
            TransferSyntax const & ) const  [pure virtual]
```

Return whether this coder support this transfer syntax (can code it).

Implemented in gdcm::AudioCodec, gdcm::ImageCodec, gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PDFCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.55.3.2 Code()

```
virtual bool gdcm::Coder::Code (
            DataElement const & in_,
            DataElement & out_)  [inline], [virtual]
```

Code.

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.55.3.3 InternalCode()

```
virtual bool gdcm::Coder::InternalCode (
            const char * bv,
            unsigned long len,
            std::ostream & os)  [inline], [protected], [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

The documentation for this class was generated from the following file:

- gdcmCoder.h

12.56 gdcm::CodeString Class Reference

CodeString.

```
#include <gdcmCodeString.h>
```

Public Types

- typedef InternalClass::const_iterator const_iterator
- typedef InternalClass::const_reference const_reference
- typedef InternalClass::const_reverse_iterator const_reverse_iterator
- typedef InternalClass::difference_type difference_type
- typedef InternalClass::iterator iterator
- typedef InternalClass::pointer pointer
- typedef InternalClass::reference reference
- typedef InternalClass::reverse_iterator reverse_iterator
- typedef InternalClass::size_type size_type
- typedef InternalClass::value_type value_type

Public Member Functions

- CodeString ()

  CodeString constructors.
- CodeString (const InternalClass &s, size_type pos=0, size_type n=InternalClass::npos)
- CodeString (const value_type *s)
- CodeString (const value_type *s, size_type n)
- std::string GetAsString () const

  Return the full code string as std::string.
- bool IsValid () const

  Check if CodeString obj is correct..
- size_type Size () const

  Return the size of the string.

Protected Member Functions

- std::string TrimInternal () const

Friends

- bool operator!= (const CodeString &ref, const CodeString &cs)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- bool operator== (const CodeString &ref, const CodeString &cs)

12.56.1 Detailed Description

CodeString.

This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

Note

the cstor of CodeString will Trim the string on the fly so as to remove the extra leading and ending spaces. However it will not perform validation on the fly (CodeString obj can contains invalid char such as lower cases). This design was chosen to be a little tolerant to broken DICOM implementation, and thus allow user to compare lower case CS from there input file without the need to first rewrite them to get rid of invalid character (validation is a different operation from searching, querying).

Warning

when writing out DICOM file it is highly recommended to perform the IsValid() call, at least to check that the length of the string match the definition in the standard.

12.56.2 Member Typedef Documentation

12.56.2.1 const_iterator

```
typedef InternalClass::const_iterator gdcm::CodeString::const_iterator
```

12.56.2.2 const_reference

```
typedef InternalClass::const_reference gdcm::CodeString::const_reference
```

12.56.2.3 const_reverse_iterator

```
typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator
```

12.56.2.4 difference_type

```
typedef InternalClass::difference_type gdcm::CodeString::difference_type
```

12.56.2.5 iterator

```
typedef InternalClass::iterator gdcm::CodeString::iterator
```

12.56.2.6 pointer

```
typedef InternalClass::pointer gdcm::CodeString::pointer
```

12.56.2.7 reference

```
typedef InternalClass::reference gdcm::CodeString::reference
```

12.56.2.8 reverse_iterator

```
typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator
```

12.56.2.9 size_type

```
typedef InternalClass::size_type gdcm::CodeString::size_type
```

12.56.2.10 value_type

```
typedef InternalClass::value_type gdcm::CodeString::value_type
```

12.56.3 Constructor & Destructor Documentation

12.56.3.1 CodeString() [1/4]

```
gdcm::CodeString::CodeString () [inline]
```

CodeString constructors.

Referenced by operator!=, operator<<, and operator==.

12.56.3.2 CodeString() [2/4]

```
gdcm::CodeString::CodeString (
            const value_type * s)  [inline]
```

12.56.3.3 CodeString() [3/4]

```
gdcm::CodeString::CodeString (
            const value_type * s,
            size_type n)  [inline]
```

12.56.3.4 CodeString() [4/4]

```
gdcm::CodeString::CodeString (
            const InternalClass & s,
            size_type pos = 0,
            size_type n = InternalClass::npos)  [inline]
```

12.56.4 Member Function Documentation

12.56.4.1 GetAsString()

```
std::string gdcm::CodeString::GetAsString () const  [inline]
```

Return the full code string as std::string.

12.56.4.2 IsValid()

```
bool gdcm::CodeString::IsValid () const
```

Check if CodeString obj is correct..

12.56.4.3 Size()

```
size_type gdcm::CodeString::Size () const  [inline]
```

Return the size of the string.

12.56.4.4 TrimInternal()

```
std::string gdcm::CodeString::TrimInternal () const  [inline], [protected]
```

12.56.5 Friends And Related Symbol Documentation

12.56.5.1 operator"!=

```
bool operator!= (
            const CodeString & ref,
            const CodeString & cs)  [friend]
```

References CodeString().

12.56.5.2 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const CodeString & str)  [friend]
```

References CodeString().

12.56.5.3 operator==

```
bool operator== (
            const CodeString & ref,
            const CodeString & cs)  [friend]
```

References CodeString().

The documentation for this class was generated from the following file:

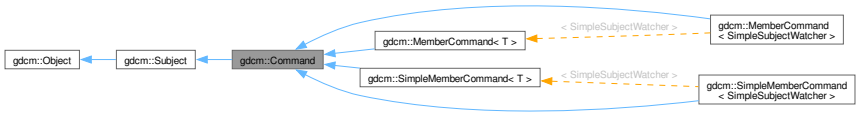
- gdcmCodeString.h

12.57 gdcm::Command Class Reference

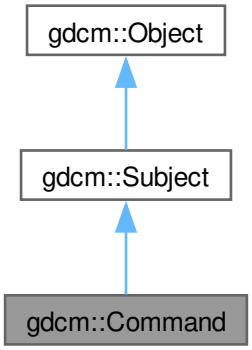
Command superclass for callback/observer methods.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::Command:



Collaboration diagram for gdcm::Command:



Public Member Functions

- Command (const Command &)=delete
- virtual void Execute (const Subject *caller, const Event &event)=0
- virtual void Execute (Subject *caller, const Event &event)=0

  Abstract method that defines the action to be taken by the command.
- void operator= (const Command &)=delete

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Protected Member Functions

- Command ()
- ~Command () override

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.57.1 Detailed Description

Command superclass for callback/observer methods.

See also

Subject

12.57.2 Constructor & Destructor Documentation

12.57.2.1 Command() [1/2]

```
gdcm::Command::Command (
            const Command & )  [delete]
```

References Command().

Referenced by Command(), and operator=().

12.57.2.2 Command() [2/2]

```
gdcm::Command::Command ()  [protected]
```

12.57.2.3 ~Command()

```
gdcm::Command::~Command ()  [override], [protected]
```

12.57.3 Member Function Documentation

12.57.3.1 Execute() [1/2]

```
virtual void gdcm::Command::Execute (
            const Subject * caller,
            const Event & event)  [pure virtual]
```

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const Object

Implemented in gdcm::MemberCommand< T >, gdcm::MemberCommand< SimpleSubjectWatcher >, gdcm::SimpleMemberCommand
and gdcm::SimpleMemberCommand< SimpleSubjectWatcher >.

References gdcm::Subject::Subject().

12.57.3.2 Execute() [2/2]

```
virtual void gdcm::Command::Execute (
            Subject * caller,
            const Event & event)  [pure virtual]
```

Abstract method that defines the action to be taken by the command.

Implemented in gdcm::MemberCommand< T >, gdcm::MemberCommand< SimpleSubjectWatcher >, gdcm::SimpleMemberCommand
and gdcm::SimpleMemberCommand< SimpleSubjectWatcher >.

References gdcm::Subject::Subject().

12.57.3.3 operator=()

```
void gdcm::Command::operator= (
            const Command & )  [delete]
```

References Command().

The documentation for this class was generated from the following file:

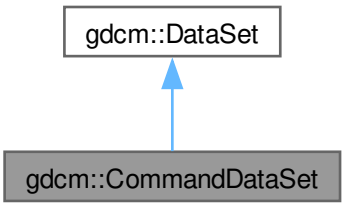
- gdcmCommand.h

12.58 gdcm::CommandDataSet Class Reference

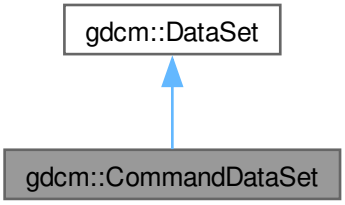
Class to represent a Command DataSet.

```
#include <gdcmCommandDataSet.h>
```

Inheritance diagram for gdcm::CommandDataSet:



Collaboration diagram for gdcm::CommandDataSet:



Public Member Functions

- CommandDataSet ()=default
- ~CommandDataSet ()=default
- void Insert (const DataElement &de)
- std::istream & Read (std::istream &is)

  Read.
- void Replace (const DataElement &de)
- std::ostream & Write (std::ostream &os) const

  Write.

Public Member Functions inherited from gdcm::DataSet

- Iterator Begin ()
- ConstIterator Begin () const
- void Clear ()
- template<typename TDE>
  unsigned int ComputeGroupLength (Tag const &tag) const
- Iterator End ()
- ConstIterator End () const
- bool FindDataElement (const PrivateTag &t) const

  Look up if private tag 't' is present in the dataset:
- bool FindDataElement (const Tag &t) const
- const DataElement & FindNextDataElement (const Tag &t) const
- const DataElement & GetDataElement (const PrivateTag &t) const

  Return the dataelement.
- const DataElement & GetDataElement (const Tag &t) const
- DataElementSet & GetDES ()
- const DataElementSet & GetDES () const
- template<typename TDE>
  VL GetLength () const
- MediaStorage GetMediaStorage () const
- std::string GetPrivateCreator (const Tag &t) const
- PrivateTag GetPrivateTag (const Tag &t) const

  Return the private tag of the private tag 't', private creator will be set to empty if not found.
- void Insert (const DataElement &de)
- bool IsEmpty () const

  Returns if the dataset is empty.
- const DataElement & operator() (uint16_t group, uint16_t element) const
- DataSet & operator= (DataSet const &)=default
- const DataElement & operator[] (const Tag &t) const
- void Print (std::ostream &os, std::string const &indent="") const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadNested (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedPrivateTags (std::istream &is, const std::set< PrivateTag > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedPrivateTagsWithLength (std::istream &is, const std::set< PrivateTag > &tags, VL &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedTags (std::istream &is, const std::set< Tag > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedTagsWithLength (std::istream &is, const std::set< Tag > &tags, VL &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadUpToTag (std::istream &is, const Tag &t, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadUpToTagWithLength (std::istream &is, const Tag &t, std::set< Tag > const &skiptags, VL &length)

- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- SizeType Remove (const Tag &tag)

  Completely remove a dataelement from the dataset.
- void Replace (const DataElement &de)

  Replace a dataelement with another one.
- void ReplaceEmpty (const DataElement &de)

  Only replace a DICOM attribute when it is missing or empty.
- SizeType Size () const
- template<typename TDE, typename TSwap>
  std::ostream const & Write (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &_os, const CommandDataSet &_val)

Additional Inherited Members

Public Types inherited from gdcm::DataSet

- typedef DataElementSet::const_iterator ConstIterator
- typedef std::set< DataElement > DataElementSet
- typedef DataElementSet::iterator Iterator
- typedef DataElementSet::size_type SizeType

Protected Member Functions inherited from gdcm::DataSet

- Tag ComputeDataElement (const PrivateTag &t) const
- const DataElement & GetDEEnd () const
- void InsertDataElement (const DataElement &de)

12.58.1 Detailed Description

Class to represent a Command DataSet.

See also

DataSet

12.58.2 Constructor & Destructor Documentation

12.58.2.1 CommandDataSet()

```
gdcm::CommandDataSet::CommandDataSet () [default]
```

Referenced by ~CommandDataSet(), and operator<<.

12.58.2.2 ~CommandDataSet()

```
gdcm::CommandDataSet::~CommandDataSet () [default]
```

References CommandDataSet(), and operator<<.

12.58.3 Member Function Documentation

12.58.3.1 Insert()

```
void gdcm::CommandDataSet::Insert (
            const DataElement & de) [inline]
```

References gdcmErrorMacro, gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), and gdcm::DataSet::InsertDataElement().

Referenced by Replace().

12.58.3.2 Read()

```
std::istream & gdcm::CommandDataSet::Read (
            std::istream & is)
```

Read.

12.58.3.3 Replace()

```
void gdcm::CommandDataSet::Replace (
            const DataElement & de) [inline]
```

References gdcm::DataElement::GetTag(), Insert(), and gdcm::DataSet::Remove().

12.58.3.4 Write()

```
std::ostream & gdcm::CommandDataSet::Write (
            std::ostream & os) const
```

Write.

12.58.4 Friends And Related Symbol Documentation

12.58.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const CommandDataSet & _val)  [friend]
```

References CommandDataSet(), and gdcm::DataSet::Print().

Referenced by ~CommandDataSet().

The documentation for this class was generated from the following file:

- gdcmCommandDataSet.h

12.59 gdcm::network::CompositeMessageFactory Class Reference

CompositeMessageFactory.

```
#include <gdcmCompositeMessageFactory.h>
```

Static Public Member Functions

- static std::vector< PresentationDataValue > ConstructCEchoRQ (const ULConnection &inConnection)
- static std::vector< PresentationDataValue > ConstructCFindRQ (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector< PresentationDataValue > ConstructCMoveRQ (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)
- static std::vector< PresentationDataValue > ConstructCStoreRQ (const ULConnection &inConnection, const File &file, bool writeDataSet=true)
- static std::vector< PresentationDataValue > ConstructCStoreRSP (const DataSet *inDataSet, const BasePDU *inPC)

12.59.1 Detailed Description

CompositeMessageFactory.

This class constructs PDataPDUs, but that have been specifically constructed for the composite DICOM services (C-Echo, C-Find, C-Get, C-Move, and C-Store). It will also handle parsing the incoming data to determine which of the CompositePDUs the incoming data is, and so therefore allowing the scu to determine what to do with incoming data (if acting as a storescp server, for instance).

12.59.2 Member Function Documentation

12.59.2.1 ConstructCEchoRQ()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCEchoRQ (
            const ULConnection & inConnection)  [static]
```

12.59.2.2 ConstructCFindRQ()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCFindRQ (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery)  [static]
```

12.59.2.3 ConstructCMoveRQ()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCMoveRQ (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery)  [static]
```

12.59.2.4 ConstructCStoreRQ()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCStoreRQ (
            const ULConnection & inConnection,
            const File & file,
            bool writeDataSet = true)  [static]
```

12.59.2.5 ConstructCStoreRSP()

```
std::vector< PresentationDataValue > gdcm::network::CompositeMessageFactory::ConstructCStoreRSP (
            const DataSet * inDataSet,
            const BasePDU * inPC)  [static]
```

The documentation for this class was generated from the following file:

- gdcmCompositeMessageFactory.h

12.60 gdcm::CompositeNetworkFunctions Class Reference

Composite Network Functions.

```
#include <gdcmCompositeNetworkFunctions.h>
```

Public Types

- typedef std::vector< KeyValuePairType > KeyValuePairArrayType
- typedef std::pair< Tag, std::string > KeyValuePairType

Static Public Member Functions

- static bool CEcho (const char *remote, uint16_t portno, const char *aetitle=nullptr, const char *call=nullptr)
- static bool CFind (const char *remote, uint16_t portno, const BaseRootQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle=nullptr, const char *call=nullptr)
- static bool CMove (const char *remote, uint16_t portno, const BaseRootQuery *query, uint16_t portscp, const char *aetitle=nullptr, const char *call=nullptr, const char *outputdir=nullptr)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const DataSet &queryds, EQueryType queryType=eFind)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, EQueryType queryType=eFind)
- static bool CStore (const char *remote, uint16_t portno, const Directory::FilenamesType &filenames, const char *aetitle=nullptr, const char *call=nullptr)

12.60.1 Detailed Description

Composite Network Functions.

These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- C-ECHO SCU
- C-FIND SCU
- C-STORE SCU
- C-MOVE SCU (+internal C-STORE SCP)

Examples

SendFileSCU.cs.

12.60.2 Member Typedef Documentation

12.60.2.1 KeyValuePairArrayType

```
typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType
```

12.60.2.2 KeyValuePairType

```
typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType
```

12.60.3 Member Function Documentation

12.60.3.1 CEcho()

```
bool gdcm::CompositeNetworkFunctions::CEcho (
            const char * remote,
            uint16_t portno,
            const char * aetitle = nullptr,
            const char * call = nullptr)  [static]
```

The most basic network function. Use this function to ensure that the remote server is responding on the given IP and port number as expected.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

Examples

SendFileSCU.cs.

12.60.3.2 CFind()

```
bool gdcm::CompositeNetworkFunctions::CFind (
            const char * remote,
            uint16_t portno,
            const BaseRootQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle = nullptr,
            const char * call = nullptr)  [static]
```

This function will use the provided query to determine what files a remote server contains that match the query strings. The return is a vector of datasets that contain tags as reported by the server. If the dataset is empty, then it is possible that an error condition was encountered; in which case, the user should monitor the error and warning streams.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

12.60.3.3 CMove()

```
bool gdcm::CompositeNetworkFunctions::CMove (
            const char * remote,
            uint16_t portno,
            const BaseRootQuery * query,
            uint16_t portscp,
            const char * aetitle = nullptr,
            const char * call = nullptr,
            const char * outputdir = nullptr)  [static]
```

This function will use the provided query to get files from a remote server. NOTE that this functionality is essentially equivalent to C-GET in the DICOM standard; however, C-GET has been deprecated, so this function allows for the user to ask a remote server for files matching a query and return them to the local machine. Files will be written to the given output directory. If the operation succeeds, the function returns true. This function is a prime candidate for being overwritten by expert users; if the datasets should remain in memory, for instance, that behavior can be changed by creating a user-level version of this function.

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0 when
outputdir	is not set default to current dir ('.')

Returns

true if it worked.

12.60.3.4 ConstructQuery() [1/2]

```
BaseRootQuery * gdcm::CompositeNetworkFunctions::ConstructQuery (
            ERootType inRootType,
            EQueryLevel inQueryLevel,
            const DataSet & queryds,
            EQueryType queryType = eFind)  [static]
```

This function will take a list of strings and tags and fill in a query that can be used for either CFind or CMove (depending on the input boolean

Parameters

inMove).	Note that the caller is responsible for deleting the constructed query. This function is used to build both a move and a find query (true for inMove if it's move, false if it's find)

References gdcm::eFind.

12.60.3.5 ConstructQuery() [2/2]

```
BaseRootQuery * gdcm::CompositeNetworkFunctions::ConstructQuery (
            ERootType inRootType,
            EQueryLevel inQueryLevel,
            const KeyValuePairArrayType & keys,
            EQueryType queryType = eFind)  [static]
```

Deprecated

References gdcm::eFind.

12.60.3.6 CStore()

```
bool gdcm::CompositeNetworkFunctions::CStore (
            const char * remote,
            uint16_t portno,
            const Directory::FilenamesType & filenames,
            const char * aetitle = nullptr,
            const char * call = nullptr)  [static]
```

This function will place the provided files into the remote server. The function returns true if it worked for all files.

Warning

the server side can refuse an association on a given file

Parameters

aetitle	when not set will default to 'GDCMSCU'
call	when not set will default to 'ANY-SCP'

Warning

This is an error to set remote to NULL or portno to 0

Returns

true if it worked for all files

Examples

SendFileSCU.cs.

The documentation for this class was generated from the following file:

- gdcmCompositeNetworkFunctions.h

12.61 gdcm::ConstCharWrapper Class Reference

Do not use me.

```
#include <gdcmConstCharWrapper.h>
```

Public Member Functions

- ConstCharWrapper (const char *i=0)
- operator const char * () const

12.61.1 Detailed Description

Do not use me.

12.61.2 Constructor & Destructor Documentation

12.61.2.1 ConstCharWrapper()

```
gdcm::ConstCharWrapper::ConstCharWrapper (
            const char * i = 0)  [inline]
```

12.61.3 Member Function Documentation

12.61.3.1 operator const char *()

```
gdcm::ConstCharWrapper::operator const char * () const  [inline]
```

The documentation for this class was generated from the following file:

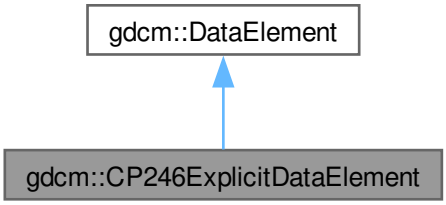
- gdcmConstCharWrapper.h

12.62 gdcm::CP246ExplicitDataElement Class Reference

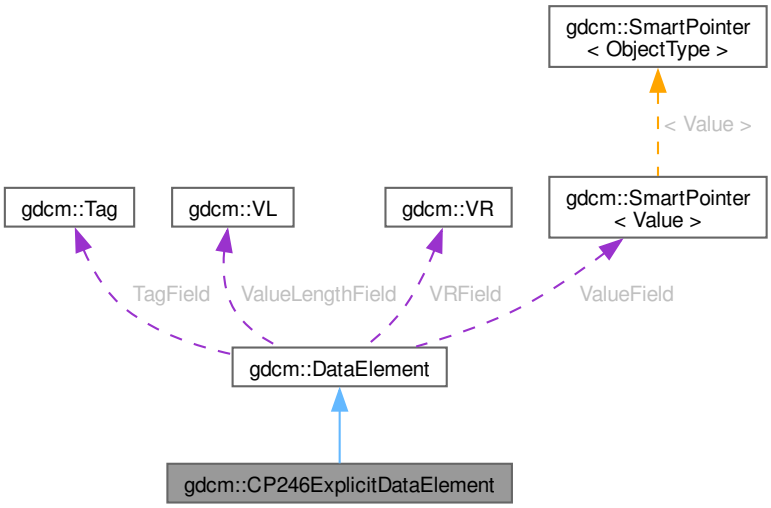
Class to read/write a DataElement as CP246Explicit Data Element.

```
#include <gdcmCP246ExplicitDataElement.h>
```

Inheritance diagram for gdcm::CP246ExplicitDataElement:



Collaboration diagram for gdcm::CP246ExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const

- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.62.1 Detailed Description

Class to read/write a DataElement as CP246Explicit Data Element.

Note

Some system are producing SQ, declare them as UN, but encode the SQ as 'Explicit' instead of Implicit

12.62.2 Member Function Documentation

12.62.2.1 GetLength()

```
VL gdcm::CP246ExplicitDataElement::GetLength () const
```

12.62.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::CP246ExplicitDataElement::Read (
            std::istream & is)
```

12.62.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::CP246ExplicitDataElement::ReadPreValue (
            std::istream & is)
```

12.62.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::CP246ExplicitDataElement::ReadValue (
            std::istream & is,
            bool readvalues = true)
```

12.62.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::CP246ExplicitDataElement::ReadWithLength (
            std::istream & is,
            VL & length)
```

The documentation for this class was generated from the following file:

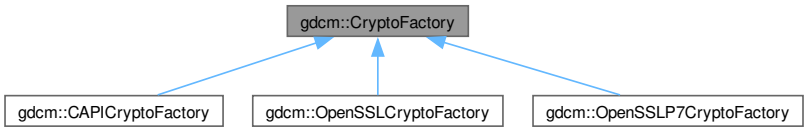
- gdcmCP246ExplicitDataElement.h

12.63 gdcm::CryptoFactory Class Reference

Class to do handle the crypto factory.

```
#include <gdcmCryptoFactory.h>
```

Inheritance diagram for gdcm::CryptoFactory:



Public Types

- enum CryptoLib {
  DEFAULT = 0 ,
  OPENSSL = 1 ,
  CAPI = 2 ,
  OPENSSLP7 = 3 }

Public Member Functions

- virtual CryptographicMessageSyntax ∗ CreateCMSProvider ()=0

Static Public Member Functions

- static CryptoFactory ∗ GetFactoryInstance (CryptoLib id=DEFAULT)

Protected Member Functions

- CryptoFactory ()=default
- CryptoFactory (CryptoLib id)
- ~CryptoFactory ()=default

12.63.1 Detailed Description

Class to do handle the crypto factory.

GDCM needs to access in a platform independent way the user specified crypto engine. It can be:

- CAPI (windows only)
- OPENSSL (portable)
- OPENSSLP7 (portable) By default the factory will try: CAPI if on windows OPENSSL if possible OPENSSLP7 when older OpenSSL is used.

12.63.2 Member Enumeration Documentation

12.63.2.1 CryptoLib

```
enum gdcm::CryptoFactory::CryptoLib
```

Enumerator

DEFAULT	
OPENSSL	
CAPI	
OPENSSLP7	

12.63.3 Constructor & Destructor Documentation

12.63.3.1 CryptoFactory() [1/2]

```
gdcm::CryptoFactory::CryptoFactory (
            CryptoLib id)  [inline], [protected]
```

Referenced by gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory(), gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory() and GetFactoryInstance().

12.63.3.2 CryptoFactory() [2/2]

```
gdcm::CryptoFactory::CryptoFactory () [protected], [default]
```

12.63.3.3 ~CryptoFactory()

```
gdcm::CryptoFactory::~CryptoFactory () [protected], [default]
```

12.63.4 Member Function Documentation

12.63.4.1 CreateCMSProvider()

```
virtual CryptographicMessageSyntax * gdcm::CryptoFactory::CreateCMSProvider () [pure virtual]
```

Implemented in gdcm::CAPICryptoFactory, gdcm::OpenSSLCryptoFactory, and gdcm::OpenSSLP7CryptoFactory.

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

12.63.4.2 GetFactoryInstance()

```
CryptoFactory * gdcm::CryptoFactory::GetFactoryInstance (
            CryptoLib id = DEFAULT) [static]
```

References CryptoFactory(), and DEFAULT.

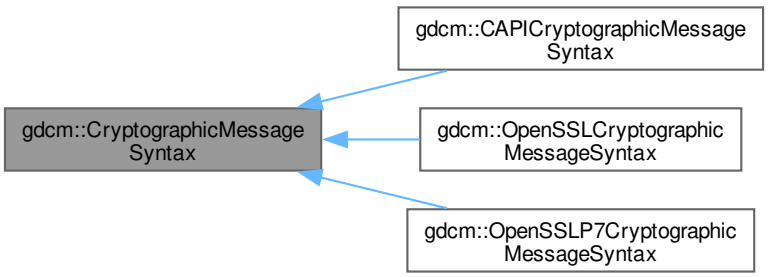
The documentation for this class was generated from the following file:

- gdcmCryptoFactory.h

12.64 gdcm::CryptographicMessageSyntax Class Reference

```
#include <gdcmCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcm::CryptographicMessageSyntax:



Public Types

- enum CipherTypes {
  DES3_CIPHER ,
  AES128_CIPHER ,
  AES192_CIPHER ,
  AES256_CIPHER }

Public Member Functions

- CryptographicMessageSyntax ()=default
- CryptographicMessageSyntax (const CryptographicMessageSyntax &)=delete
- virtual ~CryptographicMessageSyntax ()=default
- virtual bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const =0

  decrypt content from a CMS envelopedData structure
- virtual bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const =0

  create a CMS envelopedData structure
- virtual CipherTypes GetCipherType () const =0
- void operator= (const CryptographicMessageSyntax &)=delete
- virtual bool ParseCertificateFile (const char *filename)=0
- virtual bool ParseKeyFile (const char *filename)=0
- virtual void SetCipherType (CipherTypes type)=0
- virtual bool SetPassword (const char *pass, size_t passLen)=0

12.64.1 Member Enumeration Documentation

12.64.1.1 CipherTypes

```
enum gdcm::CryptographicMessageSyntax::CipherTypes
```

Enumerator

DES3_CIPHER	
AES128_CIPHER	
AES192_CIPHER	
AES256_CIPHER	

12.64.2 Constructor & Destructor Documentation

12.64.2.1 CryptographicMessageSyntax() [1/2]

```
gdcm::CryptographicMessageSyntax::CryptographicMessageSyntax ()  [default]
```

Referenced by CryptographicMessageSyntax(), and operator=().

12.64.2.2 ~CryptographicMessageSyntax()

```
virtual gdcm::CryptographicMessageSyntax::~CryptographicMessageSyntax () [virtual], [default]
```

12.64.2.3 CryptographicMessageSyntax() [2/2]

```
gdcm::CryptographicMessageSyntax::CryptographicMessageSyntax (
            const CryptographicMessageSyntax & ) [delete]
```

References CryptographicMessageSyntax().

12.64.3 Member Function Documentation

12.64.3.1 Decrypt()

```
virtual bool gdcm::CryptographicMessageSyntax::Decrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len) const [pure virtual]
```

decrypt content from a CMS envelopedData structure

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

12.64.3.2 Encrypt()

```
virtual bool gdcm::CryptographicMessageSyntax::Encrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len) const [pure virtual]
```

create a CMS envelopedData structure

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

12.64.3.3 GetCipherType()

```
virtual CipherTypes gdcm::CryptographicMessageSyntax::GetCipherType () const [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

12.64.3.4 operator=()

```
void gdcm::CryptographicMessageSyntax::operator= (
            const CryptographicMessageSyntax & )  [delete]
```

References CryptographicMessageSyntax().

12.64.3.5 ParseCertificateFile()

```
virtual bool gdcm::CryptographicMessageSyntax::ParseCertificateFile (
            const char * filename)  [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

Examples

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

12.64.3.6 ParseKeyFile()

```
virtual bool gdcm::CryptographicMessageSyntax::ParseKeyFile (
            const char * filename)  [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

12.64.3.7 SetCipherType()

```
virtual void gdcm::CryptographicMessageSyntax::SetCipherType (
            CipherTypes type)  [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

12.64.3.8 SetPassword()

```
virtual bool gdcm::CryptographicMessageSyntax::SetPassword (
            const char * pass,
            size_t passLen)  [pure virtual]
```

Implemented in gdcm::CAPICryptographicMessageSyntax, gdcm::OpenSSLCryptographicMessageSyntax, and gdcm::OpenSSLP7CryptographicMessageSyntax.

The documentation for this class was generated from the following file:

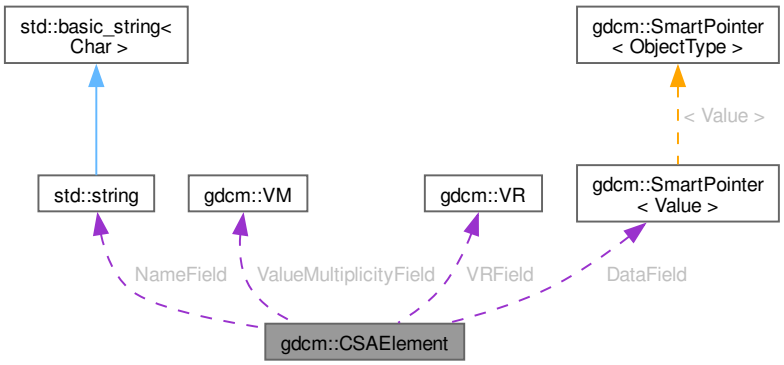
- gdcmCryptographicMessageSyntax.h

12.65 gdcm::CSAElement Class Reference

Class to represent a CSA Element.

```
#include <gdcmCSAElement.h>
```

Collaboration diagram for gdcm::CSAElement:



Public Member Functions

- CSAElement (const CSAElement &_val)
- CSAElement (unsigned int kf=0)
- const ByteValue * GetByteValue () const
- unsigned int GetKey () const

  Set/Get Key.
- const char * GetName () const

  Set/Get Name.
- unsigned int GetNoOfItems () const

  Set/Get NoOfItems.
- unsigned int GetSyngoDT () const

  Set/Get SyngoDT.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- const VM & GetVM () const

  Set/Get VM.
- VR const & GetVR () const

  Set/Get VR.
- bool IsEmpty () const

  Check if CSA Element is empty.
- bool operator< (const CSAElement &de) const

- CSAElement & operator= (const CSAElement &de)=default
- bool operator== (const CSAElement &de) const
- void SetByteValue (const char *array, VL length)

  Set.
- void SetKey (unsigned int key)
- void SetName (const char *name)
- void SetNoOfItems (unsigned int items)
- void SetSyngoDT (unsigned int syngodt)
- void SetValue (Value const &vl)
- void SetVM (const VM &vm)
- void SetVR (VR const &vr)

Protected Types

- typedef SmartPointer< Value > DataPtr

Protected Attributes

- DataPtr DataField
- unsigned int KeyField
- std::string NameField
- unsigned int NoOfItemsField
- unsigned int SyngoDTField
- VM ValueMultiplicityField
- VR VRField

Friends

- std::ostream & operator<< (std::ostream &os, const CSAElement &val)

12.65.1 Detailed Description

Class to represent a CSA Element.

See also

CSAHeader

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

12.65.2 Member Typedef Documentation

12.65.2.1 DataPtr

```
typedef SmartPointer<Value> gdcm::CSAElement::DataPtr  [protected]
```

12.65.3 Constructor & Destructor Documentation

12.65.3.1 CSAElement() [1/2]

```
gdcm::CSAElement::CSAElement (
            unsigned int kf = 0)  [inline]
```

References KeyField.

Referenced by CSAElement(), operator<(), operator<<, operator=(), and operator==().

12.65.3.2 CSAElement() [2/2]

```
gdcm::CSAElement::CSAElement (
            const CSAElement & _val)  [inline]
```

References CSAElement().

12.65.4 Member Function Documentation

12.65.4.1 GetByteValue()

```
const ByteValue * gdcm::CSAElement::GetByteValue () const  [inline]
```

Return the Value of CSAElement as a ByteValue (if possible)

Warning

: You need to check for NULL return value

Examples

DumpSiemensBase64.cxx, and MrProtocol.cxx.

References DataField.

12.65.4.2 GetKey()

```
unsigned int gdcm::CSAElement::GetKey () const  [inline]
```

Set/Get Key.

References KeyField.

Referenced by operator<().

12.65.4.3 GetName()

```
const char * gdcm::CSAElement::GetName () const  [inline]
```

Set/Get Name.

References NameField.

12.65.4.4 GetNoOfItems()

```
unsigned int gdcm::CSAElement::GetNoOfItems () const  [inline]
```

Set/Get NoOfItems.

References NoOfItemsField.

12.65.4.5 GetSyngoDT()

```
unsigned int gdcm::CSAElement::GetSyngoDT () const  [inline]
```

Set/Get SyngoDT.

References SyngoDTField.

12.65.4.6 GetValue() [1/2]

```
Value & gdcm::CSAElement::GetValue ()  [inline]
```

References DataField.

12.65.4.7 GetValue() [2/2]

```
Value const & gdcm::CSAElement::GetValue () const  [inline]
```

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples

csa2img.cxx.

References DataField.

12.65.4.8 GetVM()

```
const VM & gdcm::CSAElement::GetVM () const  [inline]
```

Set/Get VM.

References ValueMultiplicityField.

12.65.4.9 GetVR()

```
VR const & gdcm::CSAElement::GetVR () const  [inline]
```

Set/Get VR.

References VRField.

12.65.4.10 IsEmpty()

```
bool gdcm::CSAElement::IsEmpty () const  [inline]
```

Check if CSA Element is empty.

Examples

csa2img.cxx.

References DataField.

12.65.4.11 operator<()

```
bool gdcm::CSAElement::operator< (
            const CSAElement & de) const  [inline]
```

References CSAElement(), and GetKey().

12.65.4.12 operator=()

```
CSAElement & gdcm::CSAElement::operator= (
            const CSAElement & de)  [default]
```

References CSAElement().

12.65.4.13 operator==()

```
bool gdcm::CSAElement::operator== (
            const CSAElement & de) const  [inline]
```

References CSAElement(), KeyField, NameField, SyngoDTField, ValueMultiplicityField, and VRField.

12.65.4.14 SetByteValue()

```
void gdcm::CSAElement::SetByteValue (
            const char * array,
            VL length)  [inline]
```

Set.

References SetValue().

12.65.4.15 SetKey()

```
void gdcm::CSAElement::SetKey (
            unsigned int key)  [inline]
```

References KeyField.

12.65.4.16 SetName()

```
void gdcm::CSAElement::SetName (
            const char * name)  [inline]
```

References NameField.

12.65.4.17 SetNoOfItems()

```
void gdcm::CSAElement::SetNoOfItems (
            unsigned int items)  [inline]
```

References NoOfItemsField.

12.65.4.18 SetSyngoDT()

```
void gdcm::CSAElement::SetSyngoDT (
            unsigned int syngodt)  [inline]
```

References SyngoDTField.

12.65.4.19 SetValue()

```
void gdcm::CSAElement::SetValue (
            Value const & vl)  [inline]
```

References DataField.

Referenced by SetByteValue().

12.65.4.20 SetVM()

```
void gdcm::CSAElement::SetVM (
            const VM & vm)  [inline]
```

References ValueMultiplicityField.

12.65.4.21 SetVR()

```
void gdcm::CSAElement::SetVR (
            VR const & vr)  [inline]
```

References VRField.

12.65.5 Friends And Related Symbol Documentation

12.65.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const CSAElement & val)  [friend]
```

References CSAElement(), DataField, gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), KeyField, NameField, NoOfItemsField, SyngoDTField, ValueMultiplicityField, gdcm::VM::VM1, and VRField.

12.65.6 Member Data Documentation

12.65.6.1 DataField

```
DataPtr gdcm::CSAElement::DataField [protected]
```

Referenced by GetByteValue(), GetValue(), GetValue(), IsEmpty(), operator<<, and SetValue().

12.65.6.2 KeyField

```
unsigned int gdcm::CSAElement::KeyField [protected]
```

Referenced by CSAElement(), GetKey(), operator<<, operator==(), and SetKey().

12.65.6.3 NameField

```
std::string gdcm::CSAElement::NameField [protected]
```

Referenced by GetName(), operator<<, operator==(), and SetName().

12.65.6.4 NoOfItemsField

```
unsigned int gdcm::CSAElement::NoOfItemsField [protected]
```

Referenced by GetNoOfItems(), operator<<, and SetNoOfItems().

12.65.6.5 SyngoDTField

```
unsigned int gdcm::CSAElement::SyngoDTField [protected]
```

Referenced by GetSyngoDT(), operator<<, operator==(), and SetSyngoDT().

12.65.6.6 ValueMultiplicityField

```
VM gdcm::CSAElement::ValueMultiplicityField [protected]
```

Referenced by GetVM(), operator<<, operator==(), and SetVM().

12.65.6.7 VRField

```
VR gdcm::CSAElement::VRField [protected]
```

Referenced by GetVR(), operator<<, operator==(), and SetVR().

The documentation for this class was generated from the following file:

- gdcmCSAElement.h

12.66 gdcm::CSAHeader Class Reference

Class for CSAHeader.

```
#include <gdcmCSAHeader.h>
```

Public Types

- enum CSAHeaderType {
  UNKNOWN = 0 ,
  SV10 ,
  NOMAGIC ,
  DATASET_FORMAT ,
  INTERFILE ,
  ZEROED_OUT }

  Diverse format of CSAHeader as found 'in the wild'.

Public Member Functions

- CSAHeader ()
- ~CSAHeader ()=default
- bool FindCSAElementByName (const char *name)
- const CSAElement & GetCSAElementByName (const char *name)
- const DataSet & GetDataSet () const

  Return the DataSet output (use only if Format == DATASET_FORMAT).
- CSAHeaderType GetFormat () const
- const char * GetInterfile () const

  Return the string output (use only if Format == Interfile).
- bool GetMrProtocol (const DataSet &ds, MrProtocol &mrProtocol)

  Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:
- bool LoadFromDataElement (DataElement const &de)

  Decode the CSAHeader from element 'de'.
- void Print (std::ostream &os) const

  Print the CSAHeader (use only if Format == SV10 or NOMAGIC).

Static Public Member Functions

- static const PrivateTag & GetCSADataInfo ()
- static const PrivateTag & GetCSAImageHeaderInfoTag ()
- static const PrivateTag & GetCSASeriesHeaderInfoTag ()

Protected Member Functions

- const CSAElement & GetCSAEEnd () const

Friends

- std::ostream & operator<< (std::ostream &_os, const CSAHeader &d)

12.66.1 Detailed Description

Class for CSAHeader.

SIEMENS store private information in tag (0x0029,0x10,"SIEMENS CSA HEADER") this class is meant for user wishing to access values stored within this private attribute. There are basically two main 'format' for this attribute : SV10/↩ NOMAGIC and DATASET_FORMAT SV10 and NOMAGIC are from a user prospective identical, see CSAHeader.xml for possible name / value stored in this format. DATASET_FORMAT is in fact simply just another DICOM dataset (implicit) with -currently unknown- value. This can be only be printed for now.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.

the API of this class might change.

Todo MrEvaProtocol in 29,1020 contains ^M that would be nice to get rid of on UNIX system...

See also

PDBHeader

External references: 5.1.3.2.4.1 MEDCOM History Information and 5.1.4.3 CSA Non-Image Module in `http↩ ://tamsinfo.toshiba.com/docrequest/pdf/E.Soft_v2.0.pdf`

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

12.66.2 Member Enumeration Documentation

12.66.2.1 CSAHeaderType

```
enum gdcm::CSAHeader::CSAHeaderType
```

Diverse format of CSAHeader as found 'in the wild'.

Enumerator

UNKNOWN	
SV10	
NOMAGIC	
DATASET_FORMAT	
INTERFILE	
ZEROED_OUT	

12.66.3 Constructor & Destructor Documentation

12.66.3.1 CSAHeader()

```
gdcm::CSAHeader::CSAHeader () [inline]
```

References UNKNOWN.

Referenced by operator<<.

12.66.3.2 ~CSAHeader()

```
gdcm::CSAHeader::~CSAHeader () [default]
```

12.66.4 Member Function Documentation

12.66.4.1 FindCSAElementByName()

```
bool gdcm::CSAHeader::FindCSAElementByName (
            const char * name)
```

Return true if the CSA element matching 'name' is found or not

Warning

Case Sensitive

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

12.66.4.2 GetCSADataInfo()

```
const PrivateTag & gdcm::CSAHeader::GetCSADataInfo () [static]
```

Return the private tag used by SIEMENS to store the CSA Data Info This is: PrivateTag(0x0029,0x10,"SIEMENS CSA NON-IMAGE");

12.66.4.3 GetCSAEEnd()

```
const CSAElement & gdcm::CSAHeader::GetCSAEEnd () const [protected]
```

12.66.4.4 GetCSAElementByName()

```
const CSAElement & gdcm::CSAHeader::GetCSAElementByName (
            const char * name)
```

Return the CSAElement corresponding to name 'name'

Warning

Case Sensitive

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

12.66.4.5 GetCSAImageHeaderInfoTag()

```
const PrivateTag & gdcm::CSAHeader::GetCSAImageHeaderInfoTag () [static]
```

Return the private tag used by SIEMENS to store the CSA Image Header This is: PrivateTag(0x0029,0x10,"SIEMENS CSA HEADER");

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, PublicDict.cxx, and csa2img.cxx.

12.66.4.6 GetCSASeriesHeaderInfoTag()

```
const PrivateTag & gdcm::CSAHeader::GetCSASeriesHeaderInfoTag () [static]
```

Return the private tag used by SIEMENS to store the CSA Series Header This is: PrivateTag(0x0029,0x20,"SIEMENS CSA HEADER");

Examples

MrProtocol.cxx.

12.66.4.7 GetDataSet()

```
const DataSet & gdcm::CSAHeader::GetDataSet () const  [inline]
```

Return the DataSet output (use only if Format == DATASET_FORMAT).

12.66.4.8 GetFormat()

```
CSAHeaderType gdcm::CSAHeader::GetFormat () const
```

return the format of the CSAHeader SV10 and NOMAGIC are equivalent.

12.66.4.9 GetInterfile()

```
const char * gdcm::CSAHeader::GetInterfile () const  [inline]
```

Return the string output (use only if Format == Interfile).

12.66.4.10 GetMrProtocol()

```
bool gdcm::CSAHeader::GetMrProtocol (
            const DataSet & ds,
            MrProtocol & mrProtocol)
```

Retrieve the ASCII portion stored within the MrProtocol/MrPhoenixProtocol:

Examples

MrProtocol.cxx.

12.66.4.11 LoadFromDataElement()

```
bool gdcm::CSAHeader::LoadFromDataElement (
            DataElement const & de)
```

Decode the CSAHeader from element 'de'.

Examples

DumpCSA.cs, DumpSiemensBase64.cxx, MrProtocol.cxx, and csa2img.cxx.

12.66.4.12 Print()

```
void gdcm::CSAHeader::Print (
            std::ostream & os) const
```

Print the CSAHeader (use only if Format == SV10 or NOMAGIC).

Examples

csa2img.cxx.

Referenced by operator<<.

12.66.5 Friends And Related Symbol Documentation

12.66.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const CSAHeader & d)  [friend]
```

References CSAHeader(), and Print().

The documentation for this class was generated from the following file:

- gdcmCSAHeader.h

12.67 gdcm::CSAHeaderDict Class Reference

Class to represent a map of CSAHeaderDictEntry.

```
#include <gdcmCSAHeaderDict.h>
```

Public Types

- typedef MapCSAHeaderDictEntry::const_iterator ConstIterator
- typedef MapCSAHeaderDictEntry::iterator Iterator
- typedef std::set< CSAHeaderDictEntry > MapCSAHeaderDictEntry

Public Member Functions

- CSAHeaderDict ()
- CSAHeaderDict (const CSAHeaderDict &_val)=delete
- void AddCSAHeaderDictEntry (const CSAHeaderDictEntry &de)
- ConstIterator Begin () const
- ConstIterator End () const
- const CSAHeaderDictEntry & GetCSAHeaderDictEntry (const char *name) const
- bool IsEmpty () const
- CSAHeaderDict & operator= (const CSAHeaderDict &_val)=delete

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const CSAHeaderDict &_val)

12.67.1 Detailed Description

Class to represent a map of CSAHeaderDictEntry.

Examples

MrProtocol.cxx.

12.67.2 Member Typedef Documentation

12.67.2.1 ConstIterator

```
typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator
```

12.67.2.2 Iterator

```
typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator
```

12.67.2.3 MapCSAHeaderDictEntry

```
typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry
```

12.67.3 Constructor & Destructor Documentation

12.67.3.1 CSAHeaderDict() [1/2]

```
gdcm::CSAHeaderDict::CSAHeaderDict () [inline]
```

Referenced by CSAHeaderDict(), operator<<, and operator=().

12.67.3.2 CSAHeaderDict() [2/2]

```
gdcm::CSAHeaderDict::CSAHeaderDict (
            const CSAHeaderDict & _val)  [delete]
```

References CSAHeaderDict(), and operator<<.

12.67.4 Member Function Documentation

12.67.4.1 AddCSAHeaderDictEntry()

```
void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (
            const CSAHeaderDictEntry & de)  [inline]
```

12.67.4.2 Begin()

```
ConstIterator gdcm::CSAHeaderDict::Begin () const  [inline]
```

12.67.4.3 End()

```
ConstIterator gdcm::CSAHeaderDict::End () const  [inline]
```

12.67.4.4 GetCSAHeaderDictEntry()

```
const CSAHeaderDictEntry & gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (
            const char * name) const  [inline]
```

Examples

MrProtocol.cxx.

12.67.4.5 IsEmpty()

```
bool gdcm::CSAHeaderDict::IsEmpty () const  [inline]
```

12.67.4.6 LoadDefault()

```
void gdcm::CSAHeaderDict::LoadDefault ()  [protected]
```

12.67.4.7 operator=()

```
CSAHeaderDict & gdcm::CSAHeaderDict::operator= (
            const CSAHeaderDict & _val)  [delete]
```

References CSAHeaderDict().

12.67.5 Friends And Related Symbol Documentation

12.67.5.1 Dicts

```
friend class Dicts  [friend]
```

References Dicts.

Referenced by Dicts.

12.67.5.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const CSAHeaderDict & _val)  [friend]
```

References CSAHeaderDict().

Referenced by CSAHeaderDict().

The documentation for this class was generated from the following file:

- gdcmCSAHeaderDict.h

12.68 gdcm::CSAHeaderDictEntry Class Reference

Class to represent an Entry in the Dict.

```
#include <gdcmCSAHeaderDictEntry.h>
```

Public Member Functions

- CSAHeaderDictEntry (const char *name="", VR const &vr=VR::INVALID, VM const &vm=VM::VM0, const char *desc="")
- const char * GetDescription () const

  Set/Get Description.
- const char * GetName () const

  Set/Get Name.
- const VM & GetVM () const

  Set/Get VM.
- const VR & GetVR () const

  Set/Get VR.
- bool operator< (const CSAHeaderDictEntry &entry) const
- void SetDescription (const char *desc)
- void SetName (const char *name)
- void SetVM (VM const &vm)
- void SetVR (const VR &vr)

Friends

- std::ostream & operator<< (std::ostream &_os, const CSAHeaderDictEntry &_val)

12.68.1 Detailed Description

Class to represent an Entry in the Dict.

Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information

Note

bla TODO FIXME: Need a PublicCSAHeaderDictEntry...indeed CSAHeaderDictEntry has a notion of retired which does not exist in PrivateCSAHeaderDictEntry...

See also

gdcm::Dict

Examples

MrProtocol.cxx.

12.68.2 Constructor & Destructor Documentation

12.68.2.1 CSAHeaderDictEntry()

```
gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (
            const char * name = "",
            VR const & vr = VR::INVALID,
            VM const & vm = VM::VM0,
            const char * desc = "")  [inline]
```

References gdcm::VR::INVALID, and gdcm::VM::VM0.

Referenced by operator<(), and operator<<.

12.68.3 Member Function Documentation

12.68.3.1 GetDescription()

```
const char * gdcm::CSAHeaderDictEntry::GetDescription () const  [inline]
```

Set/Get Description.

12.68.3.2 GetName()

```
const char * gdcm::CSAHeaderDictEntry::GetName () const  [inline]
```

Set/Get Name.

Referenced by operator<().

12.68.3.3 GetVM()

```
const VM & gdcm::CSAHeaderDictEntry::GetVM () const  [inline]
```

Set/Get VM.

12.68.3.4 GetVR()

```
const VR & gdcm::CSAHeaderDictEntry::GetVR () const  [inline]
```

Set/Get VR.

12.68.3.5 operator<()

```
bool gdcm::CSAHeaderDictEntry::operator< (
            const CSAHeaderDictEntry & entry) const  [inline]
```

References CSAHeaderDictEntry(), and GetName().

12.68.3.6 SetDescription()

```
void gdcm::CSAHeaderDictEntry::SetDescription (
            const char * desc)  [inline]
```

12.68.3.7 SetName()

```
void gdcm::CSAHeaderDictEntry::SetName (
            const char * name)  [inline]
```

12.68.3.8 SetVM()

```
void gdcm::CSAHeaderDictEntry::SetVM (
            VM const & vm)  [inline]
```

12.68.3.9 SetVR()

```
void gdcm::CSAHeaderDictEntry::SetVR (
            const VR & vr)  [inline]
```

12.68.4 Friends And Related Symbol Documentation

12.68.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const CSAHeaderDictEntry & _val)  [friend]
```

References CSAHeaderDictEntry().

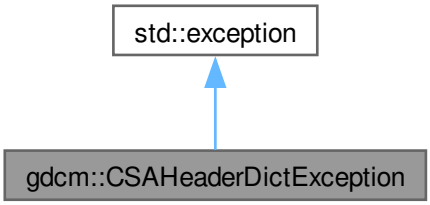
The documentation for this class was generated from the following file:

- gdcmCSAHeaderDictEntry.h

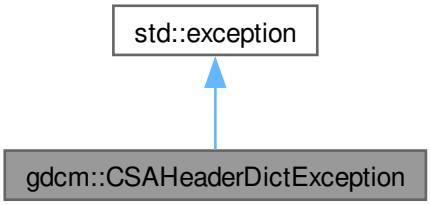
12.69 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



The documentation for this class was generated from the following file:

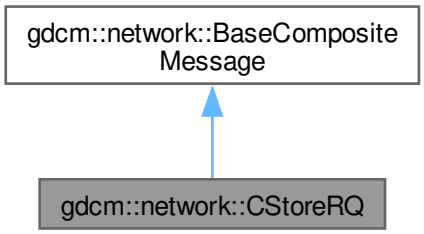
- gdcmCSAHeaderDict.h

12.70 gdcm::network::CStoreRQ Class Reference

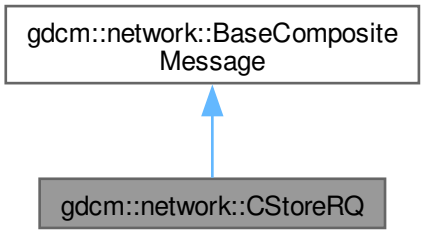
CStoreRQ.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRQ:



Collaboration diagram for gdcm::network::CStoreRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const File &file, bool writeDataSet=true)

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default

12.70.1 Detailed Description

CStoreRQ.

this file defines the messages for the cecho action

12.70.2 Member Function Documentation

12.70.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CStoreRQ::ConstructPDV (
            const ULConnection & inConnection,
            const File & file,
            bool writeDataSet = true)
```

The documentation for this class was generated from the following file:

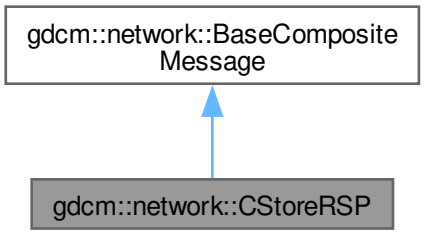
- gdcmCStoreMessages.h

12.71 gdcm::network::CStoreRSP Class Reference

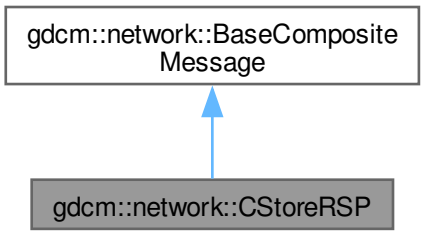
CStoreRSP this file defines the messages for the cecho action.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for gdcm::network::CStoreRSP:



Collaboration diagram for gdcm::network::CStoreRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const DataSet *inDataSet, const BasePDU *inPC)

Public Member Functions inherited from gdcm::network::BaseCompositeMessage

- virtual ~BaseCompositeMessage ()=default

12.71.1 Detailed Description

CStoreRSP this file defines the messages for the cecho action.

12.71.2 Member Function Documentation

12.71.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::CStoreRSP::ConstructPDV (
            const DataSet * inDataSet,
            const BasePDU * inPC)
```

The documentation for this class was generated from the following file:

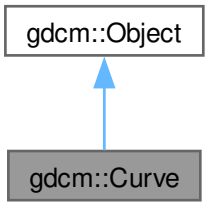
- gdcmCStoreMessages.h

12.72 gdcm::Curve Class Reference

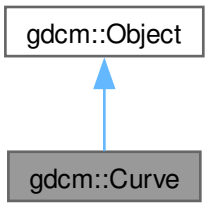
Curve class to handle element 50xx,3000 Curve Data.

```
#include <gdcmCurve.h>
```

Inheritance diagram for gdcm::Curve:



Collaboration diagram for gdcm::Curve:



Public Member Functions

- Curve ()
- Curve (Curve const &ov)
- ~Curve () override
- void Decode (std::istream &is, std::ostream &os)
- void GetAsPoints (float *array) const
- std::vector< unsigned short > const & GetCurveDataDescriptor () const
- unsigned short GetDataValueRepresentation () const
- unsigned short GetDimensions () const
- unsigned short GetGroup () const
- unsigned short GetNumberOfPoints () const
- const char * GetTypeOfData () const
- const char * GetTypeOfDataDescription () const
- bool IsEmpty () const
- void Print (std::ostream &) const override
- void SetCoordinateStartValue (unsigned short v)
- void SetCoordinateStepValue (unsigned short v)
- void SetCurve (const char *array, unsigned int length)
- void SetCurveDataDescriptor (const uint16_t *values, size_t num)
- void SetCurveDescription (const char *curvedescription)
- void SetDataValueRepresentation (unsigned short datavaluerepresentation)
- void SetDimensions (unsigned short dimensions)
- void SetGroup (unsigned short group)
- void SetNumberOfPoints (unsigned short numberofpoints)
- void SetTypeOfData (const char *typeofdata)
- void Update (const DataElement &de)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Static Public Member Functions

- static unsigned int GetNumberOfCurves (DataSet const &ds)

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.72.1 Detailed Description

Curve class to handle element 50xx,3000 Curve Data.

WARNING: This is deprecated and lastly defined in PS 3.3 - 2004

Examples:

- GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
- GE_DLX-8-MONO2-Multiframe.dcm
- gdcmSampleData/Philips_Medical_Images/integris_HV_5000/xa_integris.dcm
- TOSHIBA-CurveData[1-3].dcm

12.72.2 Constructor & Destructor Documentation

12.72.2.1 Curve() [1/2]

```
gdcm::Curve::Curve ()
```

Referenced by Curve().

12.72.2.2 ~Curve()

```
gdcm::Curve::~Curve ()  [override]
```

12.72.2.3 Curve() [2/2]

```
gdcm::Curve::Curve (
            Curve const & ov)
```

References Curve().

12.72.3 Member Function Documentation

12.72.3.1 Decode()

```
void gdcm::Curve::Decode (
            std::istream & is,
            std::ostream & os)
```

12.72.3.2 GetAsPoints()

```
void gdcm::Curve::GetAsPoints (
            float * array) const
```

12.72.3.3 GetCurveDataDescriptor()

```
std::vector< unsigned short > const & gdcm::Curve::GetCurveDataDescriptor () const
```

12.72.3.4 GetDataValueRepresentation()

```
unsigned short gdcm::Curve::GetDataValueRepresentation () const
```

12.72.3.5 GetDimensions()

```
unsigned short gdcm::Curve::GetDimensions () const
```

12.72.3.6 GetGroup()

```
unsigned short gdcm::Curve::GetGroup () const
```

12.72.3.7 GetNumberOfCurves()

```
unsigned int gdcm::Curve::GetNumberOfCurves (
            DataSet const & ds)  [static]
```

12.72.3.8 GetNumberOfPoints()

```
unsigned short gdcm::Curve::GetNumberOfPoints () const
```

12.72.3.9 GetTypeOfData()

```
const char * gdcm::Curve::GetTypeOfData () const
```

12.72.3.10 GetTypeOfDataDescription()

```
const char * gdcm::Curve::GetTypeOfDataDescription () const
```

12.72.3.11 IsEmpty()

```
bool gdcm::Curve::IsEmpty () const
```

12.72.3.12 Print()

```
void gdcm::Curve::Print (
            std::ostream & ) const  [override], [virtual]
```

Reimplemented from gdcm::Object.

12.72.3.13 SetCoordinateStartValue()

```
void gdcm::Curve::SetCoordinateStartValue (
            unsigned short v)
```

12.72.3.14 SetCoordinateStepValue()

```
void gdcm::Curve::SetCoordinateStepValue (
            unsigned short v)
```

12.72.3.15 SetCurve()

```
void gdcm::Curve::SetCurve (
            const char * array,
            unsigned int length)
```

12.72.3.16 SetCurveDataDescriptor()

```
void gdcm::Curve::SetCurveDataDescriptor (
            const uint16_t * values,
            size_t num)
```

12.72.3.17 SetCurveDescription()

```
void gdcm::Curve::SetCurveDescription (
            const char * curvedescription)
```

12.72.3.18 SetDataValueRepresentation()

```
void gdcm::Curve::SetDataValueRepresentation (
            unsigned short datavaluerepresentation)
```

12.72.3.19 SetDimensions()

```
void gdcm::Curve::SetDimensions (
            unsigned short dimensions)
```

12.72.3.20 SetGroup()

```
void gdcm::Curve::SetGroup (
            unsigned short group)
```

12.72.3.21 SetNumberOfPoints()

```
void gdcm::Curve::SetNumberOfPoints (
            unsigned short numberofpoints)
```

12.72.3.22 SetTypeOfData()

```
void gdcm::Curve::SetTypeOfData (
            const char * typeofdata)
```

12.72.3.23 Update()

```
void gdcm::Curve::Update (
            const DataElement & de)
```

The documentation for this class was generated from the following file:

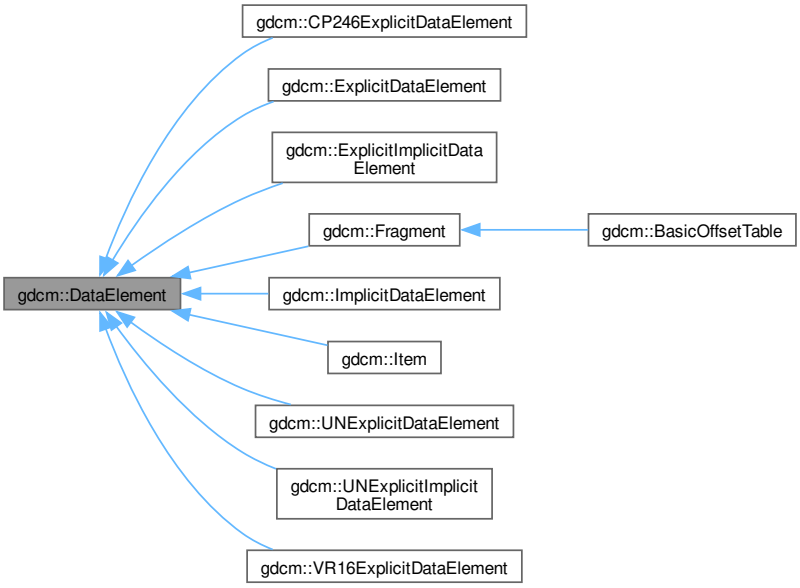
- gdcmCurve.h

12.73 gdcm::DataElement Class Reference

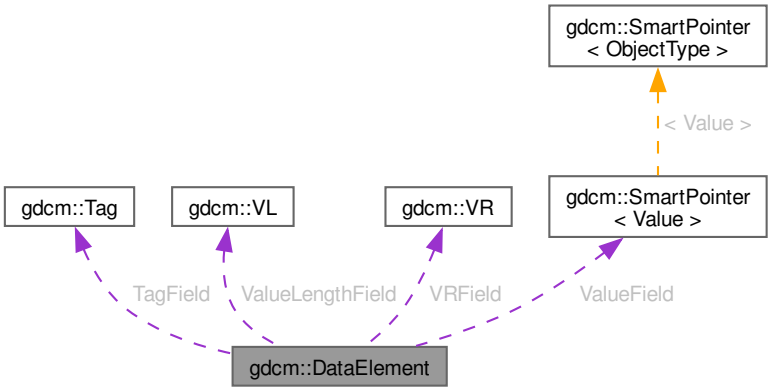
Class to represent a Data Element either Implicit or Explicit.

```
#include <gdcmDataElement.h>
```

Inheritance diagram for gdcm::DataElement:



Collaboration diagram for gdcm::DataElement:



Public Member Functions

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Protected Types

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

Friends

- std::ostream & operator<< (std::ostream &_os, const DataElement &_val)

12.73.1 Detailed Description

Class to represent a Data Element either Implicit or Explicit.

DATA ELEMENT: A unit of information as defined by a single entry in the data dictionary. An encoded Information Object Definition (IOD) Attribute that is composed of, at a minimum, three fields: a Data Element Tag, a Value Length, and a Value Field. For some specific Transfer Syntaxes, a Data Element also contains a VR Field where the Value Representation of that Data Element is specified explicitly.

Design:

- A DataElement in GDCM always store VL (Value Length) on a 32 bits integer even when VL is 16 bits
- A DataElement always store the VR even for Implicit TS, in which case VR is defaulted to VR::INVALID
- For Item start/end (See 0xfffe tags), Value is NULL

See also

ExplicitDataElement ImplicitDataElement

Examples

BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MpegVideoInfo.cs, NewSequence.cs, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, SimplePrint.cs, StreamImageReaderTest.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.73.2 Member Typedef Documentation

12.73.2.1 ValuePtr

```
typedef SmartPointer<Value> gdcm::DataElement::ValuePtr  [protected]
```

12.73.3 Constructor & Destructor Documentation

12.73.3.1 DataElement() [1/2]

```
gdcm::DataElement::DataElement (
            const Tag & t = Tag(0),
            const VL & vl = 0,
            const VR & vr = VR::INVALID)  [inline]
```

References gdcm::VR::INVALID, TagField, ValueField, ValueLengthField, and VRField.

Referenced by DataElement(), gdcm::Fragment::Fragment(), gdcm::Item::Item(), gdcm::Item::Item(), gdcm::Item::GetDataElement(), gdcm::Item::InsertDataElement(), operator<(), operator<<, operator=(), and operator==().

12.73.3.2 DataElement() [2/2]

```
gdcm::DataElement::DataElement (
            const DataElement & _val)  [inline]
```

References DataElement().

12.73.4 Member Function Documentation

12.73.4.1 Clear()

```
void gdcm::DataElement::Clear ()  [inline]
```

Clear Data Element (make Value empty and invalidate Tag & VR).

References gdcm::VR::INVALID, TagField, ValueField, ValueLengthField, and VRField.

Referenced by gdcm::Item::Clear().

12.73.4.2 Empty()

```
void gdcm::DataElement::Empty ()  [inline]
```

Make Data Element empty (no Value).

References ValueField, and ValueLengthField.

12.73.4.3 GetByteValue()

```
const ByteValue * gdcm::DataElement::GetByteValue () const  [inline]
```

Return the Value of DataElement as a ByteValue (if possible)

Warning

: You need to check for NULL return value

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, PatchFile.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References ValueField.

Referenced by IsEmpty(), gdcm::BasicOffsetTable::operator<<, gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement() gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Se gdcm::Element< TVR, TVM >::SetFromDataElement(), gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement(), and gdcm::Fragment::Write().

12.73.4.4 GetLength()

```
template<typename TDE>
VL gdcm::DataElement::GetLength () const  [inline]
```

References GetLength().

Referenced by GetLength().

12.73.4.5 GetSequenceOfFragments() [1/2]

```
SequenceOfFragments * gdcm::DataElement::GetSequenceOfFragments ()
```

12.73.4.6 GetSequenceOfFragments() [2/2]

```
const SequenceOfFragments * gdcm::DataElement::GetSequenceOfFragments () const
```

Return the Value of DataElement as a Sequence Of Fragments (if possible)

Warning

: You need to check for NULL return value

Examples

DecompressImage.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and GetJPEGSamplePrecision.cxx.

12.73.4.7 GetTag() [1/2]

```
Tag & gdcm::DataElement::GetTag () [inline]
```

References TagField.

12.73.4.8 GetTag() [2/2]

```
const Tag & gdcm::DataElement::GetTag () const [inline]
```

Get Tag.

Examples

DumpGEMSMovieGroup.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, SimplePrint.cs, and pmsct_rgb1.cxx.

References TagField.

Referenced by gdcm::CommandDataSet::Insert(), gdcm::DataSet::Insert(), gdcm::FileMetaInformation::Insert(), operator<(), gdcm::SequenceOfItems::Read(), gdcm::SequenceOfFragments::ReadValue(), gdcm::CommandDataSet::Replace(), gdcm::FileMetaInformation::Replace(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >

12.73.4.9 GetValue() [1/2]

```
Value & gdcm::DataElement::GetValue () [inline]
```

References gdcmAssertAlwaysMacro, and ValueField.

12.73.4.10 GetValue() [2/2]

```
Value const & gdcm::DataElement::GetValue () const [inline]
```

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples

ReadAndDumpDICOMDIR.cxx.

References gdcmAssertAlwaysMacro, and ValueField.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Element< TVR, TVM >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

12.73.4.11 GetValueAsSQ()

```
SmartPointer< SequenceOfItems > gdcm::DataElement::GetValueAsSQ () const
```

Interpret the Value stored in the DataElement. This is more robust (but also more expensive) to call this function rather than the simplest form: GetSequenceOfItems() It also return NULL when the Value is NOT of type SequenceOfItems

Warning

in case GetSequenceOfItems() succeed the function return this value, otherwise it creates a new SequenceOfItems, you should handle that in your case, for instance: SmartPointer<SequenceOfItems> sqi = de.GetValueAsSQ();

Examples

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ExtractEncryptedContent.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, ReadAndDumpDICOMDIR.cxx, SimplePrint.cs, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.73.4.12 GetVL() [1/2]

```
VL & gdcm::DataElement::GetVL () [inline]
```

References ValueLengthField.

12.73.4.13 GetVL() [2/2]

```
const VL & gdcm::DataElement::GetVL () const [inline]
```

Get VL.

Examples

SimplePrint.cs.

References ValueLengthField.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::SequenceOfItems::Read(), and gdcm::SequenceOfFragments::ReadValue().

12.73.4.14 GetVR()

```
VR const & gdcm::DataElement::GetVR () const [inline]
```

Get VR do not set VR::SQ on bytevalue data element

Examples

DuplicatePCDE.cxx, and GenFakeIdentifyFile.cxx.

References VRField.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, TVM >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Se gdcm::Element< TVR, TVM >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

12.73.4.15 IsEmpty()

```
bool gdcm::DataElement::IsEmpty () const [inline]
```

Check if Data Element is empty.

Examples

DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, ELSCINT1WaveToText.cxx, FixJAIBugJPEGLS.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References GetByteValue(), and ValueField.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Se and gdcm::Fragment::Write().

12.73.4.16 IsUndefinedLength()

```
bool gdcm::DataElement::IsUndefinedLength () const [inline]
```

return if Value Length if of undefined length

References ValueLengthField.

Referenced by gdcm::Item::InsertDataElement().

12.73.4.17 operator<()

```
bool gdcm::DataElement::operator< (
            const DataElement & de) const  [inline]
```

References DataElement(), and GetTag().

12.73.4.18 operator=()

```
DataElement & gdcm::DataElement::operator= (
            const DataElement & )  [default]
```

References DataElement().

12.73.4.19 operator==()

```
bool gdcm::DataElement::operator== (
            const DataElement & de) const  [inline]
```

References DataElement(), TagField, ValueField, ValueLengthField, and VRField.

12.73.4.20 Read()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataElement::Read (
            std::istream & is)  [inline]
```

Examples

DumpSiemensBase64.cxx.

References Read().

Referenced by Read(), and ReadOrSkip().

12.73.4.21 ReadOrSkip()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataElement::ReadOrSkip (
            std::istream & is,
            std::set< Tag > const & skiptags)  [inline]
```

References Read().

12.73.4.22 ReadPreValue()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataElement::ReadPreValue (
            std::istream & is,
            std::set< Tag > const & skiptags)  [inline]
```

References ReadPreValue().

Referenced by ReadPreValue().

12.73.4.23 ReadValue()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataElement::ReadValue (
            std::istream & is,
            std::set< Tag > const & skiptags)  [inline]
```

References ReadValue().

Referenced by ReadValue().

12.73.4.24 ReadValueWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataElement::ReadValueWithLength (
            std::istream & is,
            VL & length,
            std::set< Tag > const & skiptags)  [inline]
```

References ReadValueWithLength().

Referenced by ReadValueWithLength().

12.73.4.25 ReadWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataElement::ReadWithLength (
            std::istream & is,
            VL & length)  [inline]
```

References ReadWithLength().

Referenced by gdcm::Item::Read(), and ReadWithLength().

12.73.4.26 SetByteValue()

```
void gdcm::DataElement::SetByteValue (
            const char * array,
            VL length)  [inline]
```

Set the byte value

Warning

user need to read DICOM standard for an understanding of:

- even padding
- \0 vs space padding By default even padding is achieved using \0 regardless of the of VR

Examples

BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, Fake_Image_Using_Stream_Image_Writer.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, NewSequence.cs, StreamImageReaderTest.cxx, iU22tomultisc.cxx, and rle2img.cxx.

References SetValue().

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, TVM >::GetAsDataElement(), and gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement().

12.73.4.27 SetTag()

```
void gdcm::DataElement::SetTag (
            const Tag & t)  [inline]
```

Set Tag Use with cautious (need to match Part 6)

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenFakeIdentifyFile.cxx, and GetSubSequenceData.cxx.

References TagField.

12.73.4.28 SetValue()

```
void gdcm::DataElement::SetValue (
            Value const & vl)  [inline]
```

Warning

you need to set the ValueLengthField explicitly

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, DuplicatePCDE.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, MpegVideoInfo.cs, and NewSequence.cs.

References gdcm::Value::GetLength(), ValueField, and ValueLengthField.

Referenced by SetByteValue().

12.73.4.29 SetValueFieldLength()

```
void gdcm::DataElement::SetValueFieldLength (
            VL vl,
            bool readvalues)  [protected]
```

12.73.4.30 SetVL()

```
void gdcm::DataElement::SetVL (
            const VL & vl)  [inline]
```

Set VL Use with cautious (need to match Part 6), advanced user only

See also

SetByteValue

References ValueLengthField.

12.73.4.31 SetVLToUndefined()

```
void gdcm::DataElement::SetVLToUndefined ()
```

Examples

Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, and NewSequence.cs.

12.73.4.32 SetVR()

```
void gdcm::DataElement::SetVR (
            VR const & vr)  [inline]
```

Set VR Use with cautious (need to match Part 6), advanced user only

Precondition

vr is a VR::VRALL (not a dual one such as OB_OW)

Examples

Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, NewSequence.cs, StreamImageReaderTest.cxx, iU22tomultisc.cxx, and rle2img.cxx.

References gdcm::VR::IsVRFile(), and VRField.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< TVR, TVM >::GetAsDataElement(), and gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement().

12.73.4.33 Write()

```
template<typename TDE, typename TSwap>
const std::ostream & gdcm::DataElement::Write (
            std::ostream & os) const  [inline]
```

References Write().

Referenced by Write().

12.73.5 Friends And Related Symbol Documentation

12.73.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const DataElement & _val)  [friend]
```

References DataElement(), operator<<, gdcm::Object::Print(), TagField, ValueField, ValueLengthField, and VRField.

Referenced by operator<<.

12.73.6 Member Data Documentation

12.73.6.1 TagField

```
Tag gdcm::DataElement::TagField [protected]
```

Referenced by DataElement(), Clear(), GetTag(), GetTag(), operator<<, gdcm::Fragment::operator<<, gdcm::Item::operator<<, operator==(), gdcm::BasicOffsetTable::Read(), gdcm::Item::Read(), gdcm::Fragment::ReadBacktrack(), gdcm::Fragment::ReadPreValue(), SetTag(), gdcm::Fragment::Write(), and gdcm::Item::Write().

12.73.6.2 ValueField

```
ValuePtr gdcm::DataElement::ValueField [protected]
```

Referenced by DataElement(), Clear(), Empty(), GetByteValue(), GetValue(), GetValue(), IsEmpty(), gdcm::BasicOffsetTable::operator<<, operator<<, gdcm::Fragment::operator<<, operator==(), gdcm::BasicOffsetTable::Read(), gdcm::Fragment::ReadBacktrack(), gdcm::Fragment::ReadValue(), and SetValue().

12.73.6.3 ValueLengthField

```
VL gdcm::DataElement::ValueLengthField [protected]
```

Referenced by DataElement(), Clear(), Empty(), GetVL(), GetVL(), IsUndefinedLength(), gdcm::BasicOffsetTable::operator<<, operator<<, gdcm::Fragment::operator<<, gdcm::Item::operator<<, operator==(), gdcm::BasicOffsetTable::Read(), gdcm::Item::Read(), gdcm::Fragment::ReadBacktrack(), gdcm::Fragment::ReadPreValue(), gdcm::Fragment::ReadValue(), SetValue(), SetVL(), gdcm::Fragment::Write(), and gdcm::Item::Write().

12.73.6.4 VRField

```
VR gdcm::DataElement::VRField [protected]
```

Referenced by DataElement(), Clear(), GetVR(), operator<<, operator==(), and SetVR().

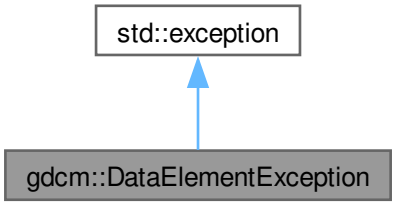
The documentation for this class was generated from the following file:

- gdcmDataElement.h

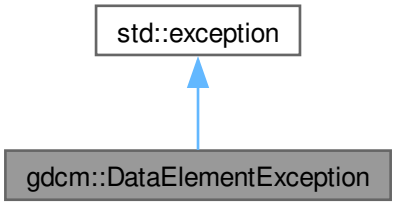
12.74 gdcm::DataElementException Class Reference

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



The documentation for this class was generated from the following file:

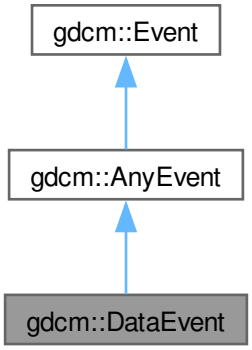
- gdcmDataSet.h

12.75 gdcm::DataEvent Class Reference

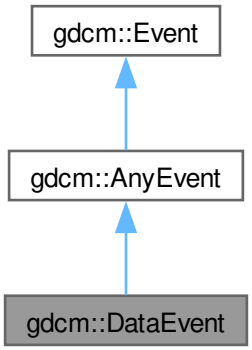
DataEvent.

```
#include <gdcmDataEvent.h>
```

Inheritance diagram for gdcm::DataEvent:



Collaboration diagram for gdcm::DataEvent:



Public Types

- typedef DataEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- DataEvent (const char ∗bytes=nullptr, size_t len=0)

- DataEvent (const Self &s)
- ~DataEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- const char * GetData () const
- size_t GetDataLength () const
- const char * GetEventName () const override
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete
- void SetData (const char *bytes, size_t len)

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

12.75.1 Detailed Description

DataEvent.

12.75.2 Member Typedef Documentation

12.75.2.1 Self

```
typedef DataEvent gdcm::DataEvent::Self
```

12.75.2.2 Superclass

```
typedef AnyEvent gdcm::DataEvent::Superclass
```

12.75.3 Constructor & Destructor Documentation

12.75.3.1 DataEvent() [1/2]

```
gdcm::DataEvent::DataEvent (
            const char * bytes = nullptr,
            size_t len = 0)  [inline]
```

12.75.3.2 ~DataEvent()

```
gdcm::DataEvent::~DataEvent () [override], [default]
```

12.75.3.3 DataEvent() [2/2]

```
gdcm::DataEvent::DataEvent (
            const Self & s) [inline]
```

12.75.4 Member Function Documentation

12.75.4.1 CheckEvent()

```
bool gdcm::DataEvent::CheckEvent (
            const ::gdcm::Event * e) const [inline], [override]
```

12.75.4.2 GetData()

```
const char * gdcm::DataEvent::GetData () const [inline]
```

12.75.4.3 GetDataLength()

```
size_t gdcm::DataEvent::GetDataLength () const [inline]
```

12.75.4.4 GetEventName()

```
const char * gdcm::DataEvent::GetEventName () const [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

12.75.4.5 MakeObject()

```
::gdcm::Event * gdcm::DataEvent::MakeObject () const [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

12.75.4.6 operator=()

```
void gdcm::DataEvent::operator= (
            const Self & )  [delete]
```

12.75.4.7 SetData()

```
void gdcm::DataEvent::SetData (
            const char * bytes,
            size_t len)  [inline]
```

The documentation for this class was generated from the following file:

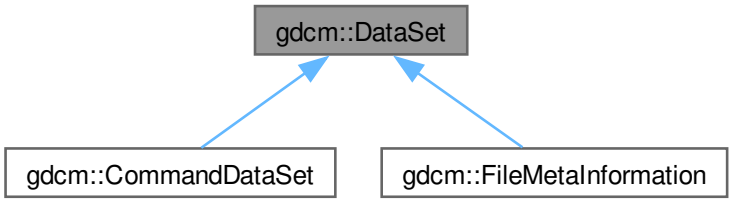
- gdcmDataEvent.h

12.76 gdcm::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements).

```
#include <gdcmDataSet.h>
```

Inheritance diagram for gdcm::DataSet:



Public Types

- typedef DataElementSet::const_iterator ConstIterator
- typedef std::set< DataElement > DataElementSet
- typedef DataElementSet::iterator Iterator
- typedef DataElementSet::size_type SizeType

Public Member Functions

- Iterator Begin ()
- ConstIterator Begin () const
- void Clear ()
- template<typename TDE>
  unsigned int ComputeGroupLength (Tag const &tag) const
- Iterator End ()
- ConstIterator End () const
- bool FindDataElement (const PrivateTag &t) const

  Look up if private tag 't' is present in the dataset:
- bool FindDataElement (const Tag &t) const
- const DataElement & FindNextDataElement (const Tag &t) const
- const DataElement & GetDataElement (const PrivateTag &t) const

  Return the dataelement.
- const DataElement & GetDataElement (const Tag &t) const
- DataElementSet & GetDES ()
- const DataElementSet & GetDES () const
- template<typename TDE>
  VL GetLength () const
- MediaStorage GetMediaStorage () const
- std::string GetPrivateCreator (const Tag &t) const
- PrivateTag GetPrivateTag (const Tag &t) const

  Return the private tag of the private tag 't', private creator will be set to empty if not found.
- void Insert (const DataElement &de)
- bool IsEmpty () const

  Returns if the dataset is empty.
- const DataElement & operator() (uint16_t group, uint16_t element) const
- DataSet & operator= (DataSet const &)=default
- const DataElement & operator[] (const Tag &t) const
- void Print (std::ostream &os, std::string const &indent="") const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadNested (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedPrivateTags (std::istream &is, const std::set< PrivateTag > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedPrivateTagsWithLength (std::istream &is, const std::set< PrivateTag > &tags, VL &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedTags (std::istream &is, const std::set< Tag > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedTagsWithLength (std::istream &is, const std::set< Tag > &tags, VL &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadUpToTag (std::istream &is, const Tag &t, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadUpToTagWithLength (std::istream &is, const Tag &t, std::set< Tag > const &skiptags, VL &length)

- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- SizeType Remove (const Tag &tag)

  Completely remove a dataelement from the dataset.
- void Replace (const DataElement &de)

  Replace a dataelement with another one.
- void ReplaceEmpty (const DataElement &de)

  Only replace a DICOM attribute when it is missing or empty.
- SizeType Size () const
- template<typename TDE, typename TSwap>
  std::ostream const & Write (std::ostream &os) const

Protected Member Functions

- Tag ComputeDataElement (const PrivateTag &t) const
- const DataElement & GetDEEnd () const
- void InsertDataElement (const DataElement &de)

Friends

- class CSAHeader
- std::ostream & operator<< (std::ostream &_os, const DataSet &val)

12.76.1 Detailed Description

Class to represent a Data Set (which contains Data Elements).

A Data Set represents an instance of a real world Information Object

Note

DATA SET: Exchanged information consisting of a structured set of Attribute values directly or indirectly related to Information Objects. The value of each Attribute in a Data Set is expressed as a Data Element. A collection of Data Elements ordered by increasing Data Element Tag number that is an encoding of the values of Attributes of a real world object.

Implementation note. If one do: DataSet ds; ds.SetLength(0); ds.Read(is); setting length to 0 actually means try to read is as if it was a root DataSet. Other value are undefined (nested dataset with undefined length) or defined length (different from 0) means nested dataset with defined length.

Warning

a DataSet does not have a Transfer Syntax type, only a File does.

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Write FileChangeTS.cs, FileChangeTSLossy.cs, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, SimplePrint.cs, SortImage.cxx, SortImage2.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, VolumeSorter.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.76.2 Member Typedef Documentation

12.76.2.1 ConstIterator

```
typedef DataElementSet::const_iterator gdcm::DataSet::ConstIterator
```

Examples

DiffFile.cxx, DumpGEMSMovieGroup.cxx, DumpVisusChange.cxx, and DuplicatePCDE.cxx.

12.76.2.2 DataElementSet

```
typedef std::set<DataElement> gdcm::DataSet::DataElementSet
```

12.76.2.3 Iterator

```
typedef DataElementSet::iterator gdcm::DataSet::Iterator
```

12.76.2.4 SizeType

```
typedef DataElementSet::size_type gdcm::DataSet::SizeType
```

12.76.3 Member Function Documentation

12.76.3.1 Begin() [1/2]

```
Iterator gdcm::DataSet::Begin () [inline]
```

12.76.3.2 Begin() [2/2]

```
ConstIterator gdcm::DataSet::Begin () const [inline]
```

Examples

DiffFile.cxx, DumpGEMSMovieGroup.cxx, DumpVisusChange.cxx, and DuplicatePCDE.cxx.

12.76.3.3 Clear()

```
void gdcm::DataSet::Clear () [inline]
```

Referenced by gdcm::Item::Read().

12.76.3.4 ComputeDataElement()

```
Tag gdcm::DataSet::ComputeDataElement (
            const PrivateTag & t) const [protected]
```

References operator<<.

12.76.3.5 ComputeGroupLength()

```
template<typename TDE>
unsigned int gdcm::DataSet::ComputeGroupLength (
            Tag const & tag) const [inline]
```

References gdcm::Tag::GetElement(), and gdcm::Tag::GetGroup().

12.76.3.6 End() [1/2]

```
Iterator gdcm::DataSet::End () [inline]
```

12.76.3.7 End() [2/2]

```
ConstIterator gdcm::DataSet::End () const  [inline]
```

Examples

DiffFile.cxx, DumpGEMSMovieGroup.cxx, DumpVisusChange.cxx, and DuplicatePCDE.cxx.

12.76.3.8 FindDataElement() [1/2]

```
bool gdcm::DataSet::FindDataElement (
            const PrivateTag & t) const
```

Look up if private tag 't' is present in the dataset:

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >: and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

12.76.3.9 FindDataElement() [2/2]

```
bool gdcm::DataSet::FindDataElement (
            const Tag & t) const  [inline]
```

References GetDataElement(), and GetDEEnd().

12.76.3.10 FindNextDataElement()

```
const DataElement & gdcm::DataSet::FindNextDataElement (
            const Tag & t) const  [inline]
```

Examples

DuplicatePCDE.cxx.

References GetDEEnd().

12.76.3.11 GetDataElement() [1/2]

```
const DataElement & gdcm::DataSet::GetDataElement (
            const PrivateTag & t) const
```

Return the dataelement.

12.76.3.12 GetDataElement() [2/2]

```
const DataElement & gdcm::DataSet::GetDataElement (
            const Tag & t) const  [inline]
```

Return the DataElement with Tag 't'

Warning

: This only search at the 'root level' of the DataSet

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, DecompressImage.cs, DeriveSeries.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cx GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, PatchFile.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References GetDEEnd().

Referenced by FindDataElement(), operator()(), operator[]() , gdcm::Attribute< Group, Element, TVR, TVM >::Set(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataS and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

12.76.3.13 GetDEEnd()

```
const DataElement & gdcm::DataSet::GetDEEnd () const  [protected]
```

Referenced by FindDataElement(), FindNextDataElement(), and GetDataElement().

12.76.3.14 GetDES() [1/2]

```
DataElementSet & gdcm::DataSet::GetDES ()  [inline]
```

12.76.3.15 GetDES() [2/2]

```
const DataElementSet & gdcm::DataSet::GetDES () const  [inline]
```

Examples

ReadAndDumpDICOMDIR.cxx.

12.76.3.16 GetLength()

```
template<typename TDE>
VL gdcm::DataSet::GetLength () const  [inline]
```

Referenced by gdcm::FileMetaInformation::GetFullLength().

12.76.3.17 GetMediaStorage()

```
MediaStorage gdcm::DataSet::GetMediaStorage () const
```

12.76.3.18 GetPrivateCreator()

```
std::string gdcm::DataSet::GetPrivateCreator (
            const Tag & t) const
```

Return the private creator of the private tag 't': or an empty string when not found

Examples

DuplicatePCDE.cxx.

12.76.3.19 GetPrivateTag()

```
PrivateTag gdcm::DataSet::GetPrivateTag (
            const Tag & t) const
```

Return the private tag of the private tag 't', private creator will be set to empty if not found.

12.76.3.20 Insert()

```
void gdcm::DataSet::Insert (
            const DataElement & de)  [inline]
```

Insert a DataElement in the DataSet.

Warning

: Tag need to be >= 0x8 to be considered valid data element

Examples

CreateJPIPDataSet.cxx, DumpSiemensBase64.cxx, DuplicatePCDE.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, NewSequence.cs, StreamImageReaderTest.cxx, and TemplateEmptyImage.cxx.

References gdcmErrorMacro, gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), and InsertDataElement().

12.76.3.21 InsertDataElement()

```
void gdcm::DataSet::InsertDataElement (
            const DataElement & de)  [inline], [protected]
```

References gdcmWarningMacro, gdcm::Value::GetLength(), gdcm::DataElement::GetValue(), gdcm::DataElement::GetVL(), and gdcm::DataElement::IsEmpty().

Referenced by gdcm::CommandDataSet::Insert(), Insert(), and gdcm::FileMetaInformation::Insert().

12.76.3.22 IsEmpty()

```
bool gdcm::DataSet::IsEmpty () const  [inline]
```

Returns if the dataset is empty.

Referenced by gdcm::Item::Read().

12.76.3.23 operator()()

```
const DataElement & gdcm::DataSet::operator() (
            uint16_t group,
            uint16_t element) const  [inline]
```

References GetDataElement().

12.76.3.24 operator=()

```
DataSet & gdcm::DataSet::operator= (
            DataSet const & )  [default]
```

12.76.3.25 operator[]()

```
const DataElement & gdcm::DataSet::operator[] (
            const Tag & t) const  [inline]
```

References GetDataElement().

12.76.3.26 Print()

```
void gdcm::DataSet::Print (
            std::ostream & os,
            std::string const & indent = "") const  [inline]
```

Referenced by gdcm::CommandDataSet::operator<<, operator<<, gdcm::FileMetaInformation::operator<<, and gdcm::Item::operator<<.

12.76.3.27 Read()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::Read (
            std::istream & is)
```

Examples

DumpToshibaDTI.cxx, and DumpToshibaDTI2.cxx.

12.76.3.28 ReadNested()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::ReadNested (
            std::istream & is)
```

12.76.3.29 ReadSelectedPrivateTags()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::ReadSelectedPrivateTags (
            std::istream & is,
            const std::set< PrivateTag > & tags,
            bool readvalues = true)
```

12.76.3.30 ReadSelectedPrivateTagsWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::ReadSelectedPrivateTagsWithLength (
            std::istream & is,
            const std::set< PrivateTag > & tags,
            VL & length,
            bool readvalues = true)
```

12.76.3.31 ReadSelectedTags()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::ReadSelectedTags (
            std::istream & is,
            const std::set< Tag > & tags,
            bool readvalues = true)
```

12.76.3.32 ReadSelectedTagsWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::ReadSelectedTagsWithLength (
            std::istream & is,
            const std::set< Tag > & tags,
            VL & length,
            bool readvalues = true)
```

12.76.3.33 ReadUpToTag()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::ReadUpToTag (
            std::istream & is,
            const Tag & t,
            std::set< Tag > const & skiptags)
```

12.76.3.34 ReadUpToTagWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::ReadUpToTagWithLength (
            std::istream & is,
            const Tag & t,
            std::set< Tag > const & skiptags,
            VL & length)
```

12.76.3.35 ReadWithLength()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::DataSet::ReadWithLength (
            std::istream & is,
            VL & length)
```

12.76.3.36 Remove()

```
SizeType gdcm::DataSet::Remove (
            const Tag & tag)  [inline]
```

Completely remove a dataelement from the dataset.

Examples

ClinicalTrialIdentificationWorkflow.cs, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, ReformatFile.cs, StandardizeFiles.cs, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::CommandDataSet::Replace(), and gdcm::FileMetaInformation::Replace().

12.76.3.37 Replace()

```
void gdcm::DataSet::Replace (
            const DataElement & de)  [inline]
```

Replace a dataelement with another one.

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateFakeRTDOSE.cxx, DeriveSeries.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References gdcmAssertAlwaysMacro.

12.76.3.38 ReplaceEmpty()

```
void gdcm::DataSet::ReplaceEmpty (
            const DataElement & de)  [inline]
```

Only replace a DICOM attribute when it is missing or empty.

Examples

rle2img.cxx.

References gdcmAssertAlwaysMacro.

12.76.3.39 Size()

```
SizeType gdcm::DataSet::Size () const  [inline]
```

Examples

DumpGEMSMovieGroup.cxx.

Referenced by gdcm::SequenceOfItems::Read().

12.76.3.40 Write()

```
template<typename TDE, typename TSwap>
std::ostream const & gdcm::DataSet::Write (
            std::ostream & os) const
```

12.76.4 Friends And Related Symbol Documentation

12.76.4.1 CSAHeader

```
friend class CSAHeader  [friend]
```

References CSAHeader.

Referenced by CSAHeader.

12.76.4.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const DataSet & val)  [friend]
```

References operator<<, and Print().

Referenced by ComputeDataElement(), and operator<<.

The documentation for this class was generated from the following file:

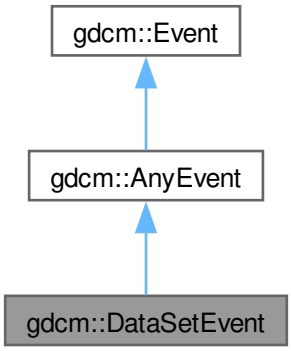
- gdcmDataSet.h

12.77 gdcm::DataSetEvent Class Reference

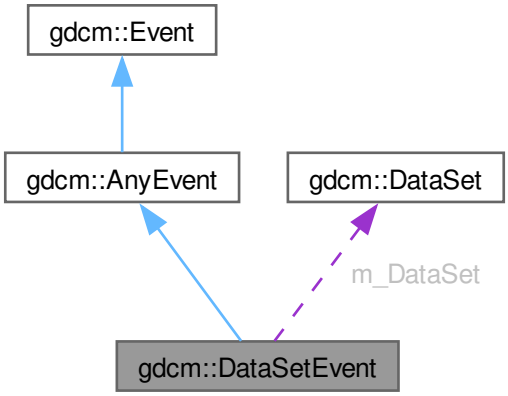
DataSetEvent.

```
#include <gdcmDataSetEvent.h>
```

Inheritance diagram for gdcm::DataSetEvent:



Collaboration diagram for gdcm::DataSetEvent:



Public Types

- typedef DataSetEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- DataSetEvent (const Self &s)
- DataSetEvent (DataSet const *ds=nullptr)
- ~DataSetEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- DataSet const & GetDataSet () const
- const char * GetEventName () const override
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

Public Attributes

- const DataSet * m_DataSet

12.77.1 Detailed Description

DataSetEvent.

Special type of event triggered during the DataSet store/move process

12.77.2 Member Typedef Documentation

12.77.2.1 Self

```
typedef DataSetEvent gdcm::DataSetEvent::Self
```

12.77.2.2 Superclass

```
typedef AnyEvent gdcm::DataSetEvent::Superclass
```

12.77.3 Constructor & Destructor Documentation

12.77.3.1 DataSetEvent() [1/2]

```
gdcm::DataSetEvent::DataSetEvent (
            DataSet const * ds = nullptr)  [inline]
```

References m_DataSet.

12.77.3.2 ~DataSetEvent()

```
gdcm::DataSetEvent::~DataSetEvent ()  [override], [default]
```

12.77.3.3 DataSetEvent() [2/2]

```
gdcm::DataSetEvent::DataSetEvent (
            const Self & s)  [inline]
```

12.77.4 Member Function Documentation

12.77.4.1 CheckEvent()

```
bool gdcm::DataSetEvent::CheckEvent (
            const ::gdcm::Event * e) const  [inline], [override]
```

12.77.4.2 GetDataSet()

```
DataSet const & gdcm::DataSetEvent::GetDataSet () const  [inline]
```

References m_DataSet.

12.77.4.3 GetEventName()

```
const char * gdcm::DataSetEvent::GetEventName () const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

12.77.4.4 MakeObject()

```
::gdcm::Event * gdcm::DataSetEvent::MakeObject () const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

12.77.4.5 operator=()

```
void gdcm::DataSetEvent::operator= (
            const Self & )  [delete]
```

12.77.5 Member Data Documentation

12.77.5.1 m_DataSet

```
const DataSet* gdcm::DataSetEvent::m_DataSet
```

Referenced by DataSetEvent(), and GetDataSet().

The documentation for this class was generated from the following file:

- gdcmDataSetEvent.h

12.78 gdcm::DataSetHelper Class Reference

DataSetHelper (internal class, not intended for user level).

```
#include <gdcmDataSetHelper.h>
```

Static Public Member Functions

- static VR ComputeVR (File const &file, DataSet const &ds, const Tag &tag)

12.78.1 Detailed Description

DataSetHelper (internal class, not intended for user level).

Examples

SimplePrint.cs.

12.78.2 Member Function Documentation

12.78.2.1 ComputeVR()

```
VR gdcm::DataSetHelper::ComputeVR (
            File const & file,
            DataSet const & ds,
            const Tag & tag)  [static]
```

ds -> current dataset, which is not the same as the root dataset return VR::INVALID in case of error

Examples

SimplePrint.cs.

The documentation for this class was generated from the following file:

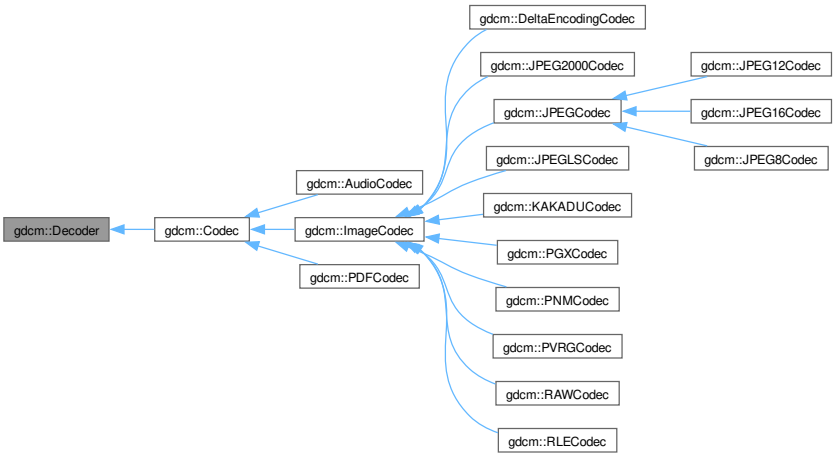
- gdcmDataSetHelper.h

12.79 gdcm::Decoder Class Reference

Decoder.

```
#include <gdcmDecoder.h>
```

Inheritance diagram for gdcm::Decoder:



Public Member Functions

- virtual ~Decoder ()=default
- virtual bool CanDecode (TransferSyntax const &) const =0

  Return whether this decoder support this transfer syntax (can decode it).
- virtual bool Decode (DataElement const &, DataElement &)

  Decode.

Protected Member Functions

- virtual bool DecodeByStreams (std::istream &, std::ostream &)

12.79.1 Detailed Description

Decoder.

12.79.2 Constructor & Destructor Documentation

12.79.2.1 ~Decoder()

```
virtual gdcm::Decoder::~Decoder ()  [virtual], [default]
```

12.79.3 Member Function Documentation

12.79.3.1 CanDecode()

```
virtual bool gdcm::Decoder::CanDecode (
            TransferSyntax const & ) const  [pure virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Implemented in gdcm::AudioCodec, gdcm::ImageCodec, gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PDFCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.79.3.2 Decode()

```
virtual bool gdcm::Decoder::Decode (
            DataElement const & ,
            DataElement & )  [inline], [virtual]
```

Decode.

Reimplemented in gdcm::AudioCodec, gdcm::DeltaEncodingCodec, gdcm::ImageCodec, gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PDFCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.79.3.3 DecodeByStreams()

```
virtual bool gdcm::Decoder::DecodeByStreams (
            std::istream & ,
            std::ostream & )  [inline], [protected], [virtual]
```

Reimplemented in gdcm::ImageCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG2000Codec, gdcm::JPEG8Codec, gdcm::JPEGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

The documentation for this class was generated from the following file:

- gdcmDecoder.h

12.80 gdcm::DefinedTerms Class Reference

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

```
#include <gdcmDefinedTerms.h>
```

Public Member Functions

- DefinedTerms ()=default

12.80.1 Detailed Description

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

12.80.2 Constructor & Destructor Documentation

12.80.2.1 DefinedTerms()

```
gdcm::DefinedTerms::DefinedTerms ()  [default]
```

The documentation for this class was generated from the following file:

- gdcmDefinedTerms.h

12.81 gdcm::Defs Class Reference

FIXME I do not like the name 'Defs'.

```
#include <gdcmDefs.h>
```

Public Member Functions

- Defs ()
- Defs (const Defs &val)=delete
- ~Defs ()
- const IOD & GetIODFromFile (const File &file) const
- IODs & GetIODs ()
- const IODs & GetIODs () const
- Macros & GetMacros ()
- const Macros & GetMacros () const
- Modules & GetModules ()
- const Modules & GetModules () const
- Type GetTypeFromTag (const File &file, const Tag &tag) const
- bool IsEmpty () const
- Defs & operator= (const Defs &val)=delete
- bool Verify (const DataSet &ds) const
- bool Verify (const File &file) const

Static Public Member Functions

- static const char * GetIODNameFromMediaStorage (MediaStorage const &ms)

Protected Member Functions

- void LoadDefaults ()
- void LoadFromFile (const char *filename)

Friends

- class Global

12.81.1 Detailed Description

FIXME I do not like the name 'Defs'.

Note

bla

Examples

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

12.81.2 Constructor & Destructor Documentation

12.81.2.1 Defs() [1/2]

```
gdcm::Defs::Defs ()
```

Referenced by Defs(), and operator=().

12.81.2.2 ~Defs()

```
gdcm::Defs::~Defs ()
```

12.81.2.3 Defs() [2/2]

```
gdcm::Defs::Defs (
            const Defs & val)  [delete]
```

References Defs().

12.81.3 Member Function Documentation

12.81.3.1 GetIODFromFile()

```
const IOD & gdcm::Defs::GetIODFromFile (
            const File & file) const
```

12.81.3.2 GetIODNameFromMediaStorage()

```
const char * gdcm::Defs::GetIODNameFromMediaStorage (
            MediaStorage const & ms)  [static]
```

Examples

GenerateStandardSOPClasses.cxx.

12.81.3.3 GetIODs() [1/2]

```
IODs & gdcm::Defs::GetIODs ()  [inline]
```

12.81.3.4 GetIODs() [2/2]

```
const IODs & gdcm::Defs::GetIODs () const  [inline]
```

Examples

TraverseModules.cxx.

12.81.3.5 GetMacros() [1/2]

```
Macros & gdcm::Defs::GetMacros ()  [inline]
```

12.81.3.6 GetMacros() [2/2]

```
const Macros & gdcm::Defs::GetMacros () const  [inline]
```

Users should not directly use Macro. Macro are simply a way for DICOM WG to re-use Tables. Macros are conveniently wrapped within Modules. See gdcm::Module API directly

Examples

TraverseModules.cxx.

12.81.3.7 GetModules() [1/2]

```
Modules & gdcm::Defs::GetModules ()  [inline]
```

12.81.3.8 GetModules() [2/2]

```
const Modules & gdcm::Defs::GetModules () const  [inline]
```

Examples

TraverseModules.cxx.

Referenced by IsEmpty().

12.81.3.9 GetTypeFromTag()

```
Type gdcm::Defs::GetTypeFromTag (
            const File & file,
            const Tag & tag) const
```

12.81.3.10 IsEmpty()

```
bool gdcm::Defs::IsEmpty () const  [inline]
```

References GetModules().

12.81.3.11 LoadDefaults()

```
void gdcm::Defs::LoadDefaults ()  [protected]
```

12.81.3.12 LoadFromFile()

```
void gdcm::Defs::LoadFromFile (
            const char * filename)  [protected]
```

12.81.3.13 operator=()

```
Defs & gdcm::Defs::operator= (
            const Defs & val)  [delete]
```

References Defs().

12.81.3.14 Verify() [1/2]

```
bool gdcm::Defs::Verify (
            const DataSet & ds) const
```

12.81.3.15 Verify() [2/2]

```
bool gdcm::Defs::Verify (
            const File & file) const
```

12.81.4 Friends And Related Symbol Documentation

12.81.4.1 Global

```
friend class Global  [friend]
```

References Global.

Referenced by Global.

The documentation for this class was generated from the following file:

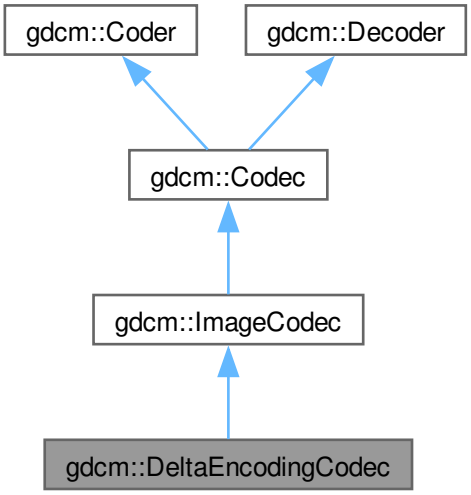
- gdcmDefs.h

12.82 gdcm::DeltaEncodingCodec Class Reference

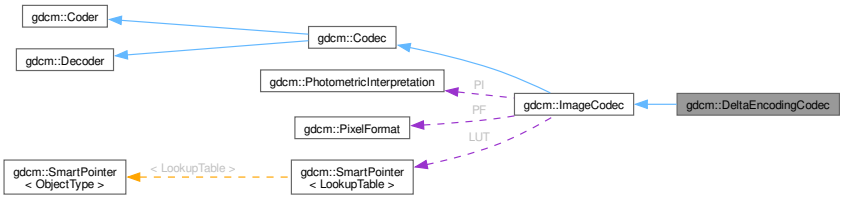
DeltaEncodingCodec compression used by some private vendor.

```
#include <gdcmDeltaEncodingCodec.h>
```

Inheritance diagram for gdcm::DeltaEncodingCodec:



Collaboration diagram for gdcm::DeltaEncodingCodec:



Public Member Functions

- DeltaEncodingCodec ()
- ~DeltaEncodingCodec ()
- bool CanDecode (TransferSyntax const &ts)
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CanCode (TransferSyntax const &) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &) const override

  Return whether this decoder support this transfer syntax (can decode it).
- bool CleanupUnusedBits (char *data, size_t datalen)
- virtual ImageCodec * Clone () const =0
- bool Decode (DataElement const &is_, DataElement &os) override

  Decode.
- const unsigned int * GetDimensions () const
- virtual bool GetHeaderInfo (std::istream &is_, TransferSyntax &ts)
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool Decode (std::istream &is, std::ostream &os)

Protected Member Functions inherited from gdcm::ImageCodec

- virtual bool AppendFrameEncode (std::ostream &out, const char ∗data, size_t datalen)
- virtual bool AppendRowEncode (std::ostream &out, const char ∗data, size_t datalen)
- bool DecodeByStreams (std::istream &is_, std::ostream &os) override
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsFrameEncoder ()
- virtual bool IsRowEncoder ()
- virtual bool IsValid (PhotometricInterpretation const &pi)
- virtual bool StartEncode (std::ostream &os)
- virtual bool StopEncode (std::ostream &os)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char ∗bv, unsigned long len, std::ostream &os)

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.82.1 Detailed Description

DeltaEncodingCodec compression used by some private vendor.

12.82.2 Constructor & Destructor Documentation

12.82.2.1 DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::DeltaEncodingCodec ()
```

12.82.2.2 ~DeltaEncodingCodec()

```
gdcm::DeltaEncodingCodec::~DeltaEncodingCodec ()
```

12.82.3 Member Function Documentation

12.82.3.1 CanDecode()

```
bool gdcm::DeltaEncodingCodec::CanDecode (
            TransferSyntax const & ts)
```

12.82.3.2 Decode() [1/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
            DataElement const & ,
            DataElement & )  [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

12.82.3.3 Decode() [2/2]

```
bool gdcm::DeltaEncodingCodec::Decode (
            std::istream & is,
            std::ostream & os)  [protected]
```

The documentation for this class was generated from the following file:

- gdcmDeltaEncodingCodec.h

12.83 gdcm::DICOMDIR Class Reference

DICOMDIR class.

```
#include <gdcmDICOMDIR.h>
```

Public Member Functions

- DICOMDIR ()=default
- DICOMDIR (FileSet fs)

12.83.1 Detailed Description

DICOMDIR class.

Structured for handling DICOMDIR

12.83.2 Constructor & Destructor Documentation

12.83.2.1 DICOMDIR() [1/2]

```
gdcm::DICOMDIR::DICOMDIR ()  [default]
```

12.83.2.2 DICOMDIR() [2/2]

```
gdcm::DICOMDIR::DICOMDIR (
            FileSet fs)  [inline]
```

The documentation for this class was generated from the following file:

- gdcmDICOMDIR.h

12.84 gdcm::DICOMDIRGenerator Class Reference

DICOMDIRGenerator class.

```
#include <gdcmDICOMDIRGenerator.h>
```

Public Types

- typedef Directory::FilenamesType FilenamesType
- typedef Directory::FilenameType FilenameType

Public Member Functions

- DICOMDIRGenerator ()
- ~DICOMDIRGenerator ()
- bool Generate ()

  Main function to generate the DICOMDIR.
- File & GetFile ()
- void SetDescriptor (const char *d)
- void SetFile (const File &f)

  Set/Get file. The DICOMDIR file will be valid once a call to Generate has been done.
- void SetFilenames (FilenamesType const &fns)

  Set the list of filenames from which the DICOMDIR should be generated from.
- void SetRootDirectory (FilenameType const &root)

  Set the root directory from which the filenames should be considered.

Protected Member Functions

- bool AddImageDirectoryRecord ()
- bool AddPatientDirectoryRecord ()
- bool AddSeriesDirectoryRecord ()
- bool AddStudyDirectoryRecord ()
- Scanner & GetScanner ()

12.84.1 Detailed Description

DICOMDIRGenerator class.

This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles

Note

PS 3.11 - 2008 / D.3.2 Physical Medium And Medium Format The STD-GEN-CD and STD-GEN-SEC-CD application profiles require the 120 mm CD-R physical medium with the ISO/IEC 9660 Media Format, as defined in PS3.12. See also PS 3.12 - 2008 / Annex F 120mm CD-R Medium (Normative) and PS 3.10 - 2008 / 8 DICOM File Service / 8.1 FILE-SET

Warning

: PS 3.11 - 2008 / D.3.1 SOP Classes and Transfer Syntaxes Composite Image & Stand-alone Storage are required to be stored as Explicit VR Little Endian Uncompressed (1.2.840.10008.1.2.1). When a DICOM file is found using another Transfer Syntax the generator will simply stops.

- Input files should be Explicit VR Little Endian
- filenames should be valid VR::CS value (16 bytes, upper case ...)

Bug : There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the Scanner does not allow us See PS 3.11 / Table D.3-2 STD-GEN Additional DICOMDIR Keys

Examples

GenerateDICOMDIR.cs.

12.84.2 Member Typedef Documentation

12.84.2.1 FilenamesType

```
typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType
```

12.84.2.2 FilenameType

```
typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType
```

12.84.3 Constructor & Destructor Documentation

12.84.3.1 DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::DICOMDIRGenerator ()
```

12.84.3.2 ∼DICOMDIRGenerator()

```
gdcm::DICOMDIRGenerator::~DICOMDIRGenerator ()
```

12.84.4 Member Function Documentation

12.84.4.1 AddImageDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ()  [protected]
```

12.84.4.2 AddPatientDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ()  [protected]
```

12.84.4.3 AddSeriesDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ()  [protected]
```

12.84.4.4 AddStudyDirectoryRecord()

```
bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ()  [protected]
```

12.84.4.5 Generate()

```
bool gdcm::DICOMDIRGenerator::Generate ()
```

Main function to generate the DICOMDIR.

Examples

GenerateDICOMDIR.cs.

12.84.4.6 GetFile()

```
File & gdcm::DICOMDIRGenerator::GetFile ()
```

Examples

GenerateDICOMDIR.cs.

12.84.4.7 GetScanner()

```
Scanner & gdcm::DICOMDIRGenerator::GetScanner ()  [protected]
```

12.84.4.8 SetDescriptor()

```
void gdcm::DICOMDIRGenerator::SetDescriptor (
            const char * d)
```

Set the File Set ID.

Warning

this need to be a valid VR::CS value

Examples

GenerateDICOMDIR.cs.

12.84.4.9 SetFile()

```
void gdcm::DICOMDIRGenerator::SetFile (
            const File & f)
```

Set/Get file. The DICOMDIR file will be valid once a call to Generate has been done.

12.84.4.10 SetFilenames()

```
void gdcm::DICOMDIRGenerator::SetFilenames (
            FilenamesType const & fns)
```

Set the list of filenames from which the DICOMDIR should be generated from.

Examples

GenerateDICOMDIR.cs.

12.84.4.11 SetRootDirectory()

```
void gdcm::DICOMDIRGenerator::SetRootDirectory (
            FilenameType const & root)
```

Set the root directory from which the filenames should be considered.

The documentation for this class was generated from the following file:

- gdcmDICOMDIRGenerator.h

12.85 gdcm::Dict Class Reference

Class to represent a map of DictEntry.

```
#include <gdcmDict.h>
```

Public Types

- typedef MapDictEntry::const_iterator ConstIterator
- typedef MapDictEntry::iterator Iterator
- typedef std::map< Tag, DictEntry > MapDictEntry

Public Member Functions

- Dict ()
- Dict (const Dict &_val)=delete
- void AddDictEntry (const Tag &tag, const DictEntry &de)
- ConstIterator Begin () const
- ConstIterator End () const
- const DictEntry & GetDictEntry (const Tag &tag) const
- const DictEntry & GetDictEntryByKeyword (const char *keyword, Tag &tag) const
- const DictEntry & GetDictEntryByName (const char *name, Tag &tag) const
- const char * GetKeywordFromTag (Tag const &tag) const

  Function to return the Keyword from a Tag.
- bool IsEmpty () const
- Dict & operator= (const Dict &_val)=delete

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const Dict &_val)

12.85.1 Detailed Description

Class to represent a map of DictEntry.

Note

bla TODO FIXME: For Element == 0x0 need to return Name = Group Length ValueRepresentation = UL Value↩ Multiplicity = 1

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

12.85.2 Member Typedef Documentation

12.85.2.1 ConstIterator

```
typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator
```

12.85.2.2 Iterator

```
typedef MapDictEntry::iterator gdcm::Dict::Iterator
```

12.85.2.3 MapDictEntry

```
typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry
```

12.85.3 Constructor & Destructor Documentation

12.85.3.1 Dict() [1/2]

```
gdcm::Dict::Dict () [inline]
```

Referenced by Dict(), operator<<, and operator=().

12.85.3.2 Dict() [2/2]

```
gdcm::Dict::Dict (
            const Dict & _val)  [delete]
```

References Dict(), and operator<<.

12.85.4 Member Function Documentation

12.85.4.1 AddDictEntry()

```
void gdcm::Dict::AddDictEntry (
            const Tag & tag,
            const DictEntry & de)  [inline]
```

12.85.4.2 Begin()

```
ConstIterator gdcm::Dict::Begin () const  [inline]
```

Examples

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

12.85.4.3 End()

```
ConstIterator gdcm::Dict::End () const  [inline]
```

Examples

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

12.85.4.4 GetDictEntry()

```
const DictEntry & gdcm::Dict::GetDictEntry (
            const Tag & tag) const  [inline]
```

Examples

GenFakeIdentifyFile.cxx, and PublicDict.cxx.

12.85.4.5 GetDictEntryByKeyword()

```
const DictEntry & gdcm::Dict::GetDictEntryByKeyword (
            const char * keyword,
            Tag & tag) const  [inline]
```

Lookup DictEntry by keyword. Even if DICOM standard defines keyword as being unique. The lookup table is built on Tag. Therefore looking up a DictEntry by Keyword is more inefficient than looking up by Tag.

12.85.4.6 GetDictEntryByName()

```
const DictEntry & gdcm::Dict::GetDictEntryByName (
            const char * name,
            Tag & tag) const  [inline]
```

Inefficient way of looking up tag by name. Technically DICOM does not guarantee uniqueness (and Curve / Overlay are there to prove it). But most of the time name is in fact uniq and can be uniquely link to a tag

Examples

ReadAndPrintAttributes.cxx.

12.85.4.7 GetKeywordFromTag()

```
const char * gdcm::Dict::GetKeywordFromTag (
            Tag const & tag) const  [inline]
```

Function to return the Keyword from a Tag.

12.85.4.8 IsEmpty()

```
bool gdcm::Dict::IsEmpty () const  [inline]
```

12.85.4.9 LoadDefault()

```
void gdcm::Dict::LoadDefault ()  [protected]
```

12.85.4.10 operator=()

```
Dict & gdcm::Dict::operator= (
            const Dict & _val)  [delete]
```

References Dict().

12.85.5 Friends And Related Symbol Documentation

12.85.5.1 Dicts

```
friend class Dicts  [friend]
```

References Dicts.

Referenced by Dicts.

12.85.5.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Dict & _val)  [friend]
```

References Dict().

Referenced by Dict().

The documentation for this class was generated from the following file:

- gdcmDict.h

12.86 gdcm::DictConverter Class Reference

Class to convert a .dic file into something else:

```
#include <gdcmDictConverter.h>
```

Public Types

- enum OutputTypes {
  DICT_DEFAULT = 0 ,
  DICT_DEBUG ,
  DICT_XML }

Public Member Functions

- DictConverter ()
- ~DictConverter ()
- void Convert ()
- const std::string & GetDictName () const
- const std::string & GetInputFilename () const
- const std::string & GetOutputFilename () const
- int GetOutputType () const
- void SetDictName (const char *name)
- void SetInputFileName (const char *filename)
- void SetOutputFileName (const char *filename)
- void SetOutputType (int type)

Static Public Member Functions

- static bool Readuint16 (const char ∗raw, uint16_t &ov)
- static bool ReadVM (const char ∗raw, VM::VMType &type)
- static bool ReadVR (const char ∗raw, VR::VRType &type)

Protected Member Functions

- void AddGroupLength ()
- bool ConvertToCXX (const char ∗raw, std::string &cxx)
- bool ConvertToXML (const char ∗raw, std::string &cxx)
- void WriteFooter ()
- void WriteHeader ()

12.86.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embed dict into shared lib (DICT_DEFAULT)
- Debug mode (DICT_DEBUG)
- XML dict (DICT_XML)

  Note

12.86.2 Member Enumeration Documentation

12.86.2.1 OutputTypes

```
enum gdcm::DictConverter::OutputTypes
```

Enumerator

DICT_DEFAULT	
DICT_DEBUG	
DICT_XML	

12.86.3 Constructor & Destructor Documentation

12.86.3.1 DictConverter()

```
gdcm::DictConverter::DictConverter ()
```

12.86.3.2 ~DictConverter()

```
gdcm::DictConverter::~DictConverter ()
```

12.86.4 Member Function Documentation

12.86.4.1 AddGroupLength()

```
void gdcm::DictConverter::AddGroupLength ()  [protected]
```

12.86.4.2 Convert()

```
void gdcm::DictConverter::Convert ()
```

12.86.4.3 ConvertToCXX()

```
bool gdcm::DictConverter::ConvertToCXX (
            const char * raw,
            std::string & cxx)  [protected]
```

12.86.4.4 ConvertToXML()

```
bool gdcm::DictConverter::ConvertToXML (
            const char * raw,
            std::string & cxx)  [protected]
```

12.86.4.5 GetDictName()

```
const std::string & gdcm::DictConverter::GetDictName () const
```

12.86.4.6 GetInputFilename()

```
const std::string & gdcm::DictConverter::GetInputFilename () const
```

12.86.4.7 GetOutputFilename()

```
const std::string & gdcm::DictConverter::GetOutputFilename () const
```

12.86.4.8 GetOutputType()

```
int gdcm::DictConverter::GetOutputType () const  [inline]
```

12.86.4.9 Readuint16()

```
bool gdcm::DictConverter::Readuint16 (
            const char * raw,
            uint16_t & ov)  [static]
```

12.86.4.10 ReadVM()

```
bool gdcm::DictConverter::ReadVM (
            const char * raw,
            VM::VMType & type)  [static]
```

12.86.4.11 ReadVR()

```
bool gdcm::DictConverter::ReadVR (
            const char * raw,
            VR::VRType & type)  [static]
```

12.86.4.12 SetDictName()

```
void gdcm::DictConverter::SetDictName (
            const char * name)
```

12.86.4.13 SetInputFileName()

```
void gdcm::DictConverter::SetInputFileName (
            const char * filename)
```

12.86.4.14 SetOutputFileName()

```
void gdcm::DictConverter::SetOutputFileName (
            const char * filename)
```

12.86.4.15 SetOutputType()

```
void gdcm::DictConverter::SetOutputType (
            int type)  [inline]
```

12.86.4.16 WriteFooter()

```
void gdcm::DictConverter::WriteFooter () [protected]
```

12.86.4.17 WriteHeader()

```
void gdcm::DictConverter::WriteHeader () [protected]
```

The documentation for this class was generated from the following file:

- gdcmDictConverter.h

12.87 gdcm::DictEntry Class Reference

Class to represent an Entry in the Dict.

```
#include <gdcmDictEntry.h>
```

Public Member Functions

- DictEntry (const char *name="", const char *keyword="", VR const &vr=VR::INVALID, VM const &vm=VM::VM0, bool ret=false)
- const char * GetKeyword () const

  same as GetName but without spaces...
- const char * GetName () const

  Set/Get Name.
- bool GetRetired () const

  Set/Get Retired flag.
- const VM & GetVM () const

  Set/Get VM.
- const VR & GetVR () const

  Set/Get VR.
- bool IsUnique () const
- void SetElementXX (bool v)

  Set whether element is shared in multiple elements (Source Image IDs typically).
- void SetGroupXX (bool v)

  Set whether element is shared in multiple groups (Curve/Overlay typically).
- void SetKeyword (const char *keyword)
- void SetName (const char *name)
- void SetRetired (bool retired)
- void SetVM (VM const &vm)
- void SetVR (const VR &vr)

Friends

- class Dict
- std::ostream & operator<< (std::ostream &_os, const DictEntry &_val)

12.87.1 Detailed Description

Class to represent an Entry in the Dict.

Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from gdcm::Tag to the needed information

Note

bla TODO FIXME: Need a PublicDictEntry...indeed DictEntry has a notion of retired which does not exist in PrivateDictEntry...

See also

gdcm::Dict

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and TraverseModules.cxx.

12.87.2 Constructor & Destructor Documentation

12.87.2.1 DictEntry()

```
gdcm::DictEntry::DictEntry (
            const char * name = "",
            const char * keyword = "",
            VR const & vr = VR::INVALID,
            VM const & vm = VM::VM0,
            bool ret = false)  [inline]
```

References gdcm::VR::INVALID, and gdcm::VM::VM0.

Referenced by operator<<.

12.87.3 Member Function Documentation

12.87.3.1 GetKeyword()

```
const char * gdcm::DictEntry::GetKeyword () const  [inline]
```

same as GetName but without spaces...

12.87.3.2 GetName()

```
const char * gdcm::DictEntry::GetName () const  [inline]
```

Set/Get Name.

Referenced by gdcm::PrivateDict::PrintXML().

12.87.3.3 GetRetired()

```
bool gdcm::DictEntry::GetRetired () const  [inline]
```

Set/Get Retired flag.

Examples

GenAllVR.cxx.

12.87.3.4 GetVM()

```
const VM & gdcm::DictEntry::GetVM () const  [inline]
```

Set/Get VM.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

12.87.3.5 GetVR()

```
const VR & gdcm::DictEntry::GetVR () const  [inline]
```

Set/Get VR.

Examples

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

12.87.3.6 IsUnique()

```
bool gdcm::DictEntry::IsUnique () const  [inline]
```

Return whether the name of the DataElement can be considered to be unique. As of 2008 all elements name were unique (except the expclitely 'XX' ones)

12.87.3.7 SetElementXX()

```
void gdcm::DictEntry::SetElementXX (
            bool v)  [inline]
```

Set whether element is shared in multiple elements (Source Image IDs typically).

12.87.3.8 SetGroupXX()

```
void gdcm::DictEntry::SetGroupXX (
            bool v)  [inline]
```

Set whether element is shared in multiple groups (Curve/Overlay typically).

12.87.3.9 SetKeyword()

```
void gdcm::DictEntry::SetKeyword (
            const char * keyword)  [inline]
```

12.87.3.10 SetName()

```
void gdcm::DictEntry::SetName (
            const char * name)  [inline]
```

12.87.3.11 SetRetired()

```
void gdcm::DictEntry::SetRetired (
            bool retired)  [inline]
```

12.87.3.12 SetVM()

```
void gdcm::DictEntry::SetVM (
            VM const & vm)  [inline]
```

Referenced by gdcm::PrivateDict::AddDictEntry().

12.87.3.13 SetVR()

```
void gdcm::DictEntry::SetVR (
            const VR & vr)  [inline]
```

Referenced by gdcm::PrivateDict::AddDictEntry().

12.87.4 Friends And Related Symbol Documentation

12.87.4.1 Dict

```
friend class Dict  [friend]
```

References Dict.

Referenced by Dict.

12.87.4.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const DictEntry & _val)  [friend]
```

References DictEntry().

The documentation for this class was generated from the following file:

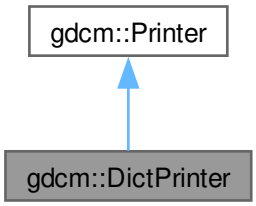
- gdcmDictEntry.h

12.88 gdcm::DictPrinter Class Reference

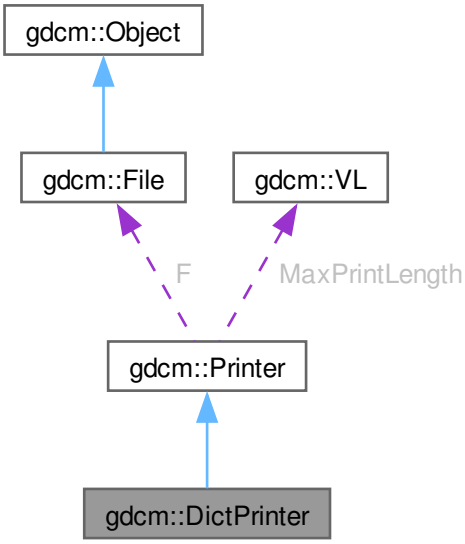
DictPrinter class.

```
#include <gdcmDictPrinter.h>
```

Inheritance diagram for gdcm::DictPrinter:



Collaboration diagram for gdcm::DictPrinter:



Public Member Functions

- DictPrinter ()
- ~DictPrinter ()=default
- void Print (std::ostream &os)

Public Member Functions inherited from gdcm::Printer

- Printer ()
- ~Printer ()=default
- PrintStyles GetPrintStyle () const

  Get PrintStyle value.
- void Print (std::ostream &os)

  Print.
- void PrintDataSet (const DataSet &ds, std::ostream &os, const std::string &s="")

  Print an individual dataset.
- void SetColor (bool c)

  Set color mode or not.
- void SetFile (File const &f)

  Set file.
- void SetStyle (PrintStyles ps)

  Set PrintStyle value.

Protected Member Functions

- void PrintDataElement2 (std::ostream &os, const DataSet &ds, const DataElement &ide)
- void PrintDataSet2 (std::ostream &os, const DataSet &ds)

Protected Member Functions inherited from gdcm::Printer

- VR PrintDataElement (std::ostringstream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream &out, std::string const &indent)
- void PrintSQ (const SequenceOfItems *sqi, std::ostream &os, std::string const &indent)

Additional Inherited Members

Public Types inherited from gdcm::Printer

- enum PrintStyles {
  VERBOSE_STYLE = 0 ,
  CONDENSED_STYLE ,
  XML ,
  CXX }

Protected Attributes inherited from gdcm::Printer

- const File * F
- VL MaxPrintLength
- PrintStyles PrintStyle

12.88.1 Detailed Description

DictPrinter class.

12.88.2 Constructor & Destructor Documentation

12.88.2.1 DictPrinter()

```
gdcm::DictPrinter::DictPrinter ()
```

12.88.2.2 ~DictPrinter()

```
gdcm::DictPrinter::~DictPrinter ()  [default]
```

12.88.3 Member Function Documentation

12.88.3.1 Print()

```
void gdcm::DictPrinter::Print (
            std::ostream & os)
```

12.88.3.2 PrintDataElement2()

```
void gdcm::DictPrinter::PrintDataElement2 (
            std::ostream & os,
            const DataSet & ds,
            const DataElement & ide)  [protected]
```

12.88.3.3 PrintDataSet2()

```
void gdcm::DictPrinter::PrintDataSet2 (
            std::ostream & os,
            const DataSet & ds)  [protected]
```

The documentation for this class was generated from the following file:

- gdcmDictPrinter.h

12.89 gdcm::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load).

```
#include <gdcmDicts.h>
```

Public Member Functions

- Dicts ()
- Dicts (const Dicts &_val)=delete
- ~Dicts ()
- const CSAHeaderDict & GetCSAHeaderDict () const
- const DictEntry & GetDictEntry (const PrivateTag &tag) const
- const DictEntry & GetDictEntry (const Tag &tag, const char *owner=nullptr) const

  THREAD SAFE.
- PrivateDict & GetPrivateDict ()
- const PrivateDict & GetPrivateDict () const
- const Dict & GetPublicDict () const
- bool IsEmpty () const
- Dicts & operator= (const Dicts &_val)=delete

Protected Types

- enum ConstructorType {
  PHILIPS ,
  GEMS ,
  SIEMENS }

Protected Member Functions

- void LoadDefaults ()

Static Protected Member Functions

- static const char ∗ GetConstructorString (ConstructorType type)

Friends

- class Global
- std::ostream & operator<< (std::ostream &_os, const Dicts &d)

12.89.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load).

Note

bla

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

12.89.2 Member Enumeration Documentation

12.89.2.1 ConstructorType

`enum gdcm::Dicts::ConstructorType [protected]`

Enumerator

PHILIPS	
GEMS	

SIEMENS

12.89.3 Constructor & Destructor Documentation

12.89.3.1 Dicts() [1/2]

```
gdcm::Dicts::Dicts ()
```

Referenced by Dicts(), operator<<, and operator=().

12.89.3.2 ~Dicts()

```
gdcm::Dicts::~Dicts ()
```

12.89.3.3 Dicts() [2/2]

```
gdcm::Dicts::Dicts (
            const Dicts & _val)  [delete]
```

References Dicts().

12.89.4 Member Function Documentation

12.89.4.1 GetConstructorString()

```
const char * gdcm::Dicts::GetConstructorString (
            ConstructorType type)  [static], [protected]
```

12.89.4.2 GetCSAHeaderDict()

```
const CSAHeaderDict & gdcm::Dicts::GetCSAHeaderDict () const
```

Examples

MrProtocol.cxx.

12.89.4.3 GetDictEntry() [1/2]

```
const DictEntry & gdcm::Dicts::GetDictEntry (
            const PrivateTag & tag) const
```

12.89.4.4 GetDictEntry() [2/2]

```
const DictEntry & gdcm::Dicts::GetDictEntry (
            const Tag & tag,
            const char * owner = nullptr) const
```

THREAD SAFE.

works for both public and private dicts: owner is null for public dict

Warning

owner need to be set to appropriate owner for call to work. see

Examples

PublicDict.cxx, and TraverseModules.cxx.

12.89.4.5 GetPrivateDict() [1/2]

```
PrivateDict & gdcm::Dicts::GetPrivateDict ()
```

12.89.4.6 GetPrivateDict() [2/2]

```
const PrivateDict & gdcm::Dicts::GetPrivateDict () const
```

12.89.4.7 GetPublicDict()

```
const Dict & gdcm::Dicts::GetPublicDict () const
```

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

Referenced by IsEmpty().

12.89.4.8 IsEmpty()

```
bool gdcm::Dicts::IsEmpty () const  [inline]
```

References GetPublicDict().

12.89.4.9 LoadDefaults()

```
void gdcm::Dicts::LoadDefaults () [protected]
```

12.89.4.10 operator=()

```
Dicts & gdcm::Dicts::operator= (
            const Dicts & _val) [delete]
```

References Dicts().

12.89.5 Friends And Related Symbol Documentation

12.89.5.1 Global

```
friend class Global [friend]
```

References Global.

Referenced by Global.

12.89.5.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Dicts & d) [friend]
```

References Dicts().

The documentation for this class was generated from the following file:

- gdcmDicts.h

12.90 gdcm::network::DIMSE Class Reference

DIMSE.

```
#include <gdcmDIMSE.h>
```

Public Types

- enum CommandTypes {
  C_STORE_RQ = 0x0001 ,
  C_STORE_RSP = 0x8001 ,
  C_GET_RQ = 0x0010 ,
  C_GET_RSP = 0x8010 ,
  C_FIND_RQ = 0x0020 ,
  C_FIND_RSP = 0x8020 ,
  C_MOVE_RQ = 0x0021 ,
  C_MOVE_RSP = 0x8021 ,
  C_ECHO_RQ = 0x0030 ,
  C_ECHO_RSP = 0x8030 ,
  N_EVENT_REPORT_RQ = 0x0100 ,
  N_EVENT_REPORT_RSP = 0x8100 ,
  N_GET_RQ = 0x0110 ,
  N_GET_RSP = 0x8110 ,
  N_SET_RQ = 0x0120 ,
  N_SET_RSP = 0x8120 ,
  N_ACTION_RQ = 0x0130 ,
  N_ACTION_RSP = 0x8130 ,
  N_CREATE_RQ = 0x0140 ,
  N_CREATE_RSP = 0x8140 ,
  N_DELETE_RQ = 0x0150 ,
  N_DELETE_RSP = 0x8150 ,
  C_CANCEL_RQ = 0x0FFF }

12.90.1 Detailed Description

DIMSE.

PS 3.7 - 2009 Annex E Command Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS Table E.1-1 COMMAND FIELDS (PART 1)

12.90.2 Member Enumeration Documentation

12.90.2.1 CommandTypes

```
enum gdcm::network::DIMSE::CommandTypes
```

Enumerator

C_STORE_RQ	
C_STORE_RSP	
C_GET_RQ	
C_GET_RSP	
C_FIND_RQ	

C_FIND_RSP	
C_MOVE_RQ	
C_MOVE_RSP	
C_ECHO_RQ	
C_ECHO_RSP	
N_EVENT_REPORT_RQ	
N_EVENT_REPORT_RSP	
N_GET_RQ	
N_GET_RSP	
N_SET_RQ	
N_SET_RSP	
N_ACTION_RQ	
N_ACTION_RSP	
N_CREATE_RQ	
N_CREATE_RSP	
N_DELETE_RQ	
N_DELETE_RSP	
C_CANCEL_RQ	

The documentation for this class was generated from the following file:

- gdcmDIMSE.h

12.91 gdcm::DirectionCosines Class Reference

class to handle DirectionCosines

```
#include <gdcmDirectionCosines.h>
```

Public Member Functions

- DirectionCosines ()
- DirectionCosines (const double dircos[6])
- ~DirectionCosines ()
- double ComputeDistAlongNormal (const double ipp[3]) const

  Compute the distance along the normal.
- void Cross (double z[3]) const

  Compute Cross product.
- double CrossDot (DirectionCosines const &dc) const

  Compute the Dot product of the two cross vector of both DirectionCosines object.
- double Dot () const

Compute Dot.

- bool IsValid () const

  Return whether or not this is a valid direction cosines.
- void Normalize ()

  Normalize in-place.
- operator const double ∗ () const

  Make the class behave like a const double ∗.
- void Print (std::ostream &) const

  Print.
- bool SetFromString (const char ∗str)

Static Public Member Functions

- static double Dot (const double x[3], const double y[3])

  Compute Dot.
- static double Norm (const double v[3])

  Return norm of the vector.
- static void Normalize (double v[3])

  Normalize in-place.

12.91.1 Detailed Description

class to handle DirectionCosines

Examples

DiscriminateVolume.cxx.

12.91.2 Constructor & Destructor Documentation

12.91.2.1 DirectionCosines() [1/2]

```
gdcm::DirectionCosines::DirectionCosines ()
```

Referenced by CrossDot().

12.91.2.2 DirectionCosines() [2/2]

```
gdcm::DirectionCosines::DirectionCosines (
            const double dircos[6])
```

12.91.2.3 ~DirectionCosines()

```
gdcm::DirectionCosines::~DirectionCosines ()
```

12.91.3 Member Function Documentation

12.91.3.1 ComputeDistAlongNormal()

```
double gdcm::DirectionCosines::ComputeDistAlongNormal (
            const double ipp[3]) const
```

Compute the distance along the normal.

12.91.3.2 Cross()

```
void gdcm::DirectionCosines::Cross (
            double z[3]) const
```

Compute Cross product.

12.91.3.3 CrossDot()

```
double gdcm::DirectionCosines::CrossDot (
            DirectionCosines const & dc) const
```

Compute the Dot product of the two cross vector of both DirectionCosines object.

Examples

DiscriminateVolume.cxx.

References DirectionCosines().

12.91.3.4 Dot() [1/2]

```
double gdcm::DirectionCosines::Dot () const
```

Compute Dot.

12.91.3.5 Dot() [2/2]

```
double gdcm::DirectionCosines::Dot (
            const double x[3],
            const double y[3])  [static]
```

Compute Dot.

12.91.3.6 IsValid()

```
bool gdcm::DirectionCosines::IsValid () const
```

Return whether or not this is a valid direction cosines.

12.91.3.7 Norm()

```
double gdcm::DirectionCosines::Norm (
            const double v[3])  [static]
```

Return norm of the vector.

12.91.3.8 Normalize() [1/2]

```
void gdcm::DirectionCosines::Normalize ()
```

Normalize in-place.

12.91.3.9 Normalize() [2/2]

```
void gdcm::DirectionCosines::Normalize (
            double v[3])  [static]
```

Normalize in-place.

12.91.3.10 operator const double *()

```
gdcm::DirectionCosines::operator const double * () const  [inline]
```

Make the class behave like a const double *.

12.91.3.11 Print()

```
void gdcm::DirectionCosines::Print (
            std::ostream & ) const
```

Print.

12.91.3.12 SetFromString()

```
bool gdcm::DirectionCosines::SetFromString (
            const char * str)
```

Initialize from string str. It requires 6 floating point separated by a backslash character.

Examples

DiscriminateVolume.cxx.

The documentation for this class was generated from the following file:

- gdcmDirectionCosines.h

12.92 gdcm::Directory Class Reference

Class for manipulation directories.

```
#include <gdcmDirectory.h>
```

Public Types

- typedef std::vector< FilenameType > FilenamesType
- typedef std::string FilenameType

Public Member Functions

- Directory ()=default
- ~Directory ()=default
- FilenamesType const & GetDirectories () const

  Return the Directories traversed.
- FilenamesType const & GetFilenames () const

  Set/Get the file names within the directory.
- FilenameType const & GetToplevel () const

  Get the name of the toplevel directory.
- unsigned int Load (FilenameType const &name, bool recursive=false)
- void Print (std::ostream &os=std::cout) const

  Print.

Protected Member Functions

- unsigned int Explore (FilenameType const &name, bool recursive)

  Return number of file found when 'recursive'ly exploring directory `name`.

Friends

- std::ostream & operator<< (std::ostream &_os, const Directory &d)

12.92.1 Detailed Description

Class for manipulation directories.

Note

This implementation provide a cross platform implementation for manipulating directories: basically traversing directories and harvesting files

will not take into account unix type hidden file recursive option will not look into UNIX type hidden directory (those starting with a '.')

Since python or C# provide there own equivalent implementation, in which case gdcm::Directory does not make much sense.

Examples

DecompressImageMultiframe.cs, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, DumpVisusChange.cxx, GenerateDICOMDIR.cs, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, ScanDirectory.cs, SortImage.cxx, StandardizeFiles.cs, VolumeSorter.cxx, gdcmorthoplanes.cxx, and threadgdcm.cxx.

12.92.2 Member Typedef Documentation

12.92.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::Directory::FilenamesType
```

Examples

CStoreQtProgress.cxx, DumpVisusChange.cxx, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, SimpleScanner.cxx, VolumeSorter.cxx, gdcmorthoplanes.cxx, reslicesphere.cxx, and threadgdcm.cxx.

12.92.2.2 FilenameType

```
typedef std::string gdcm::Directory::FilenameType
```

12.92.3 Constructor & Destructor Documentation

12.92.3.1 Directory()

```
gdcm::Directory::Directory () [default]
```

Referenced by operator<<.

12.92.3.2 ~Directory()

```
gdcm::Directory::~Directory () [default]
```

12.92.4 Member Function Documentation

12.92.4.1 Explore()

```
unsigned int gdcm::Directory::Explore (
            FilenameType const & name,
            bool recursive) [protected]
```

Return number of file found when 'recursive'ly exploring directory `name`.

12.92.4.2 GetDirectories()

```
FilenamesType const & gdcm::Directory::GetDirectories () const [inline]
```

Return the Directories traversed.

12.92.4.3 GetFilenames()

```
FilenamesType const & gdcm::Directory::GetFilenames () const [inline]
```

Set/Get the file names within the directory.

Examples

ClinicalTrialIdentificationWorkflow.cs, DecompressImageMultiframe.cs, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, DumpVisusChange.cxx, GenerateDICOMDIR.cs, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, ScanDirectory.cs, SortImage.cxx, StandardizeFiles.cs, VolumeSorter.cxx, gdcmorthoplanes.cxx, reslicesphere.cxx, and threadgdcm.cxx.

12.92.4.4 GetToplevel()

```
FilenameType const & gdcm::Directory::GetToplevel () const [inline]
```

Get the name of the toplevel directory.

12.92.4.5 Load()

```
unsigned int gdcm::Directory::Load (
            FilenameType const & name,
            bool recursive = false)
```

construct a list of filenames and subdirectory beneath directory: name

Warning

: hidden file and hidden directory are not loaded.

Examples

ClinicalTrialIdentificationWorkflow.cs, DecompressImageMultiframe.cs, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, DumpVisusChange.cxx, GenerateDICOMDIR.cs, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, ScanDirectory.cs, SortImage.cxx, StandardizeFiles.cs, VolumeSorter.cxx, gdcmorthoplanes.cxx, reslicesphere.cxx, and threadgdcm.cxx.

12.92.4.6 Print()

```
void gdcm::Directory::Print (
            std::ostream & os = std::cout) const
```

Print.

Examples

SortImage.cxx.

Referenced by operator<<.

12.92.5 Friends And Related Symbol Documentation

12.92.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Directory & d)  [friend]
```

References Directory(), and Print().

The documentation for this class was generated from the following file:

- gdcmDirectory.h

12.93 gdcm::DirectoryHelper Class Reference

DirectoryHelper.

```
#include <gdcmDirectoryHelper.h>
```

Static Public Member Functions

- static Directory::FilenamesType GetCTImageSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetFilenamesFromSeriesUIDs (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string GetFrameOfReference (const std::vector< DataSet > &inDS)
- static Directory::FilenamesType GetMRImageSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetRTStructSeriesUIDs (const std::string &inDirectory)
- static Directory::FilenamesType GetSeriesUIDsBySOPClassUID (const std::string &inDirectory, const std::string &inSOPClassUID)
- static std::string GetSOPClassUID (const std::vector< DataSet > &inDS)
- static std::string GetStringValueFromTag (const Tag &t, const DataSet &ds)
- static std::vector< DataSet > LoadImageFromFiles (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string RetrieveSOPInstanceUIDFromIndex (int inIndex, const std::vector< DataSet > &inDS)
- static std::string RetrieveSOPInstanceUIDFromZPosition (double inZPos, const std::vector< DataSet > &inDS)

12.93.1 Detailed Description

DirectoryHelper.

this class is designed to help mitigate some of the commonly performed operations on directories. namely: 1) the ability to determine the number of series in a directory by what type of series is present 2) the ability to find all ct series in a directory 3) the ability to find all mr series in a directory 4) to load a set of DataSets from a series that's already been sorted by the IPP sorter 5) For rtstruct stuff, you need to know the sopinstanceuid of each z plane, so there's a retrieval function for that 6) then a few other functions for rtstruct writeouts

12.93.2 Member Function Documentation

12.93.2.1 GetCTImageSeriesUIDs()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetCTImageSeriesUIDs (
            const std::string & inDirectory)  [static]
```

12.93.2.2 GetFilenamesFromSeriesUIDs()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (
            const std::string & inDirectory,
            const std::string & inSeriesUID)  [static]
```

Examples

GenerateRTSTRUCT.cxx.

12.93.2.3 GetFrameOfReference()

```
std::string gdcm::DirectoryHelper::GetFrameOfReference (
            const std::vector< DataSet > & inDS)  [static]
```

12.93.2.4 GetMRImageSeriesUIDs()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (
            const std::string & inDirectory)  [static]
```

12.93.2.5 GetRTStructSeriesUIDs()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (
            const std::string & inDirectory)  [static]
```

Examples

GenerateRTSTRUCT.cxx.

12.93.2.6 GetSeriesUIDsBySOPClassUID()

```
Directory::FilenamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (
            const std::string & inDirectory,
            const std::string & inSOPClassUID)  [static]
```

12.93.2.7 GetSOPClassUID()

```
std::string gdcm::DirectoryHelper::GetSOPClassUID (
            const std::vector< DataSet > & inDS)  [static]
```

12.93.2.8 GetStringValueFromTag()

```
std::string gdcm::DirectoryHelper::GetStringValueFromTag (
            const Tag & t,
            const DataSet & ds)  [static]
```

12.93.2.9 LoadImageFromFiles()

```
std::vector< DataSet > gdcm::DirectoryHelper::LoadImageFromFiles (
            const std::string & inDirectory,
            const std::string & inSeriesUID)  [static]
```

12.93.2.10 RetrieveSOPInstanceUIDFromIndex()

```
std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (
            int inIndex,
            const std::vector< DataSet > & inDS)  [static]
```

12.93.2.11 RetrieveSOPInstanceUIDFromZPosition()

```
std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromZPosition (
            double inZPos,
            const std::vector< DataSet > & inDS)  [static]
```

The documentation for this class was generated from the following file:

- gdcmDirectoryHelper.h

12.94 gdcm::DPath Class Reference

class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.google.com/g/comp.↩protocols.dicom/c/IyIH0IOBMPA`

```
#include <gdcmDPath.h>
```

Public Member Functions

- DPath ()
- ~DPath ()
- bool ConstructFromString (const char *path)
- bool Match (DPath const &other) const

  Return whether or not 'other' match the template DPath.
- bool operator< (const DPath &rhs) const
- void Print (std::ostream &) const

Static Public Member Functions

- static bool IsValid (const char *path)

  Return if path is valid or not.

Friends

- std::ostream & operator<< (std::ostream &_os, const DPath &_val)

12.94.1 Detailed Description

class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.google.com/g/comp.↩protocols.dicom/c/IyIH0IOBMPA`

12.94.2 Constructor & Destructor Documentation

12.94.2.1 DPath()

```
gdcm::DPath::DPath ()
```

Referenced by Match(), operator<(), and operator<<.

12.94.2.2 ~DPath()

```
gdcm::DPath::~DPath ()
```

12.94.3 Member Function Documentation

12.94.3.1 ConstructFromString()

```
bool gdcm::DPath::ConstructFromString (
            const char * path)
```

Examples

Cleaner.cs.

12.94.3.2 IsValid()

```
bool gdcm::DPath::IsValid (
            const char * path)  [static]
```

Return if path is valid or not.

12.94.3.3 Match()

```
bool gdcm::DPath::Match (
            DPath const & other) const
```

Return whether or not 'other' match the template DPath.

References DPath().

12.94.3.4 operator<()

```
bool gdcm::DPath::operator< (
            const DPath & rhs) const
```

References DPath().

12.94.3.5 Print()

```
void gdcm::DPath::Print (
            std::ostream & ) const
```

12.94.4 Friends And Related Symbol Documentation

12.94.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const DPath & _val)  [friend]
```

References DPath().

The documentation for this class was generated from the following file:

- gdcmDPath.h

12.95 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcmDummyValueGenerator.h>
```

Static Public Member Functions

- static const char * Generate (const char *input)

12.95.1 Detailed Description

Class for generating dummy value.

See also

Anonymizer

12.95.2 Member Function Documentation

12.95.2.1 Generate()

```
const char * gdcm::DummyValueGenerator::Generate (
            const char * input)  [static]
```

Generate a dummy value from an input value. This is guarantee to always return the same output value when input is identical. Return an array of bytes that can be used for anonymization purpose, return NULL on error NOT THREAD SAFE

The documentation for this class was generated from the following file:

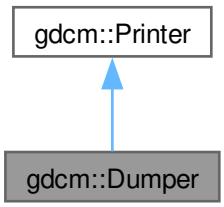
- gdcmDummyValueGenerator.h

12.96 gdcm::Dumper Class Reference

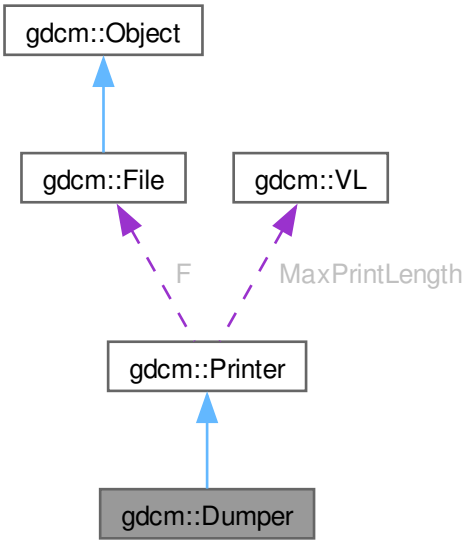
Codec class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for gdcm::Dumper:



Public Member Functions

- Dumper ()
- ~Dumper ()=default

Public Member Functions inherited from gdcm::Printer

- Printer ()
- ~Printer ()=default
- PrintStyles GetPrintStyle () const

  Get PrintStyle value.
- void Print (std::ostream &os)

  Print.
- void PrintDataSet (const DataSet &ds, std::ostream &os, const std::string &s="")

  Print an individual dataset.
- void SetColor (bool c)

  Set color mode or not.
- void SetFile (File const &f)

  Set file.
- void SetStyle (PrintStyles ps)

  Set PrintStyle value.

Additional Inherited Members

Public Types inherited from gdcm::Printer

- enum PrintStyles {
  VERBOSE_STYLE = 0 ,
  CONDENSED_STYLE ,
  XML ,
  CXX }

Protected Member Functions inherited from gdcm::Printer

- VR PrintDataElement (std::ostringstream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream &out, std::string const &indent)
- void PrintSQ (const SequenceOfItems *sqi, std::ostream &os, std::string const &indent)

Protected Attributes inherited from gdcm::Printer

- const File * F
- VL MaxPrintLength
- PrintStyles PrintStyle

12.96.1 Detailed Description

Codec class.

Note

Use it to simply dump value read from the file. No interpretation is done. But it is real fast ! Almost no overhead

12.96.2 Constructor & Destructor Documentation

12.96.2.1 Dumper()

```
gdcm::Dumper::Dumper () [inline]
```

References gdcm::Printer::CONDENSED_STYLE, and gdcm::Printer::PrintStyle.

12.96.2.2 ~Dumper()

```
gdcm::Dumper::~Dumper () [default]
```

The documentation for this class was generated from the following file:

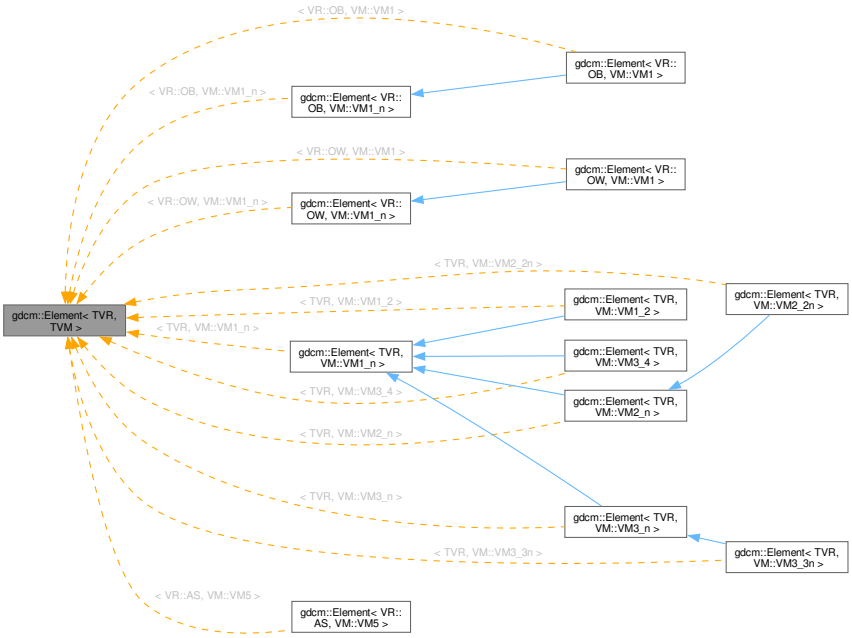
- gdcmDumper.h

12.97 gdcm::Element< TVR, TVM > Class Template Reference

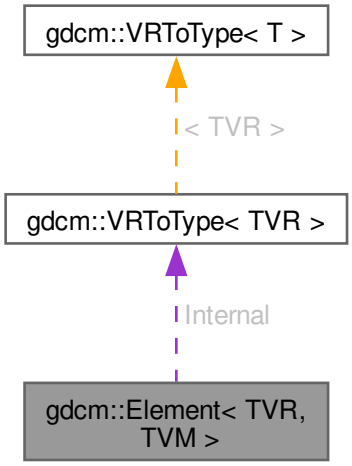
Element class.

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, TVM >:



Collaboration diagram for gdcm::Element< TVR, TVM >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

12.97.1 Detailed Description

template<long long TVR, int TVM>
class gdcm::Element< TVR, TVM >

Element class.

Note

TODO

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GetSubSequenceData.cxx, csa2img.cxx, and iU22tomultisc.cxx.

12.97.2 Member Typedef Documentation

12.97.2.1 Type

```
template<long long TVR, int TVM>
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.97.3 Member Function Documentation

12.97.3.1 GetAsDataElement()

```
template<long long TVR, int TVM>
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

Examples

Extracting_All_Resolution.cxx, and Fake_Image_Using_Stream_Image_Writer.cxx.

References GetLength(), gdcm::DataElement::GetVR(), GetVR(), Internal, gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, gdcm::VR::VRASCII, and Write().

12.97.3.2 GetLength()

```
template<long long TVR, int TVM>
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

Examples

DumpGEMSMovieGroup.cxx.

Referenced by GetAsDataElement(), Read(), Set(), SetNoSwap(), and Write().

12.97.3.3 GetValue() [1/2]

```
template<long long TVR, int TVM>
VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0)  [inline]
```

References Internal.

12.97.3.4 GetValue() [2/2]

```
template<long long TVR, int TVM>
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, GetSubSequenceData.cxx, and csa2img.cxx.

References Internal.

Referenced by operator[ ]().

12.97.3.5 GetValues()

```
template<long long TVR, int TVM>
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

References Internal.

12.97.3.6 GetVM()

```
template<long long TVR, int TVM>
VM gdcm::Element< TVR, TVM >::GetVM ()  [inline], [static]
```

12.97.3.7 GetVR()

```
template<long long TVR, int TVM>
VR gdcm::Element< TVR, TVM >::GetVR () [inline], [static]
```

Referenced by GetAsDataElement().

12.97.3.8 operator[]()

```
template<long long TVR, int TVM>
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const [inline]
```

References GetValue().

12.97.3.9 Print()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const [inline]
```

Examples

DumpGEMSMovieGroup.cxx.

References Internal.

12.97.3.10 Read()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is) [inline]
```

References GetLength(), Internal, and Read().

Referenced by Read(), and Set().

12.97.3.11 Set()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Set (
            Value const & v) [inline]
```

Examples

csa2img.cxx.

References gdcm::ByteValue::GetLength(), GetLength(), gdcm::ByteValue::GetPointer(), Internal, and Read().

Referenced by SetFromDataElement().

12.97.3.12 SetFromDataElement()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< TVR, TVM > const & de)  [inline]
```

Examples

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetValue(), gdcm::DataElement::GetVR(), gdcm::VR::INVALID, Set(), SetNoSwap(), and gdcm::VR::UN.

12.97.3.13 SetNoSwap()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), GetLength(), gdcm::ByteValue::GetPointer(), and Internal.

Referenced by SetFromDataElement().

12.97.3.14 SetValue()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

Examples

Extracting_All_Resolution.cxx, and Fake_Image_Using_Stream_Image_Writer.cxx.

References Internal.

12.97.3.15 Write()

```
template<long long TVR, int TVM>
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

References GetLength(), Internal, and Write().

Referenced by GetAsDataElement(), and Write().

12.97.4 Member Data Documentation

12.97.4.1 Internal

```
template<long long TVR, int TVM>
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

Referenced by GetAsDataElement(), GetValue(), GetValue(), GetValues(), Print(), Read(), Set(), SetNoSwap(), SetValue(), and Write().

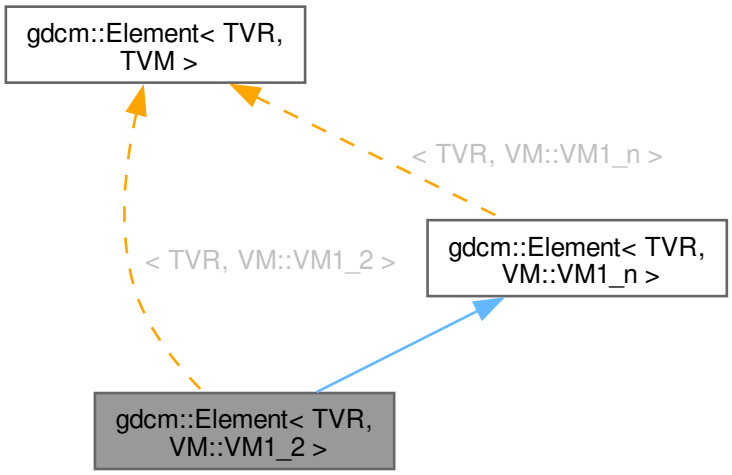
The documentation for this class was generated from the following file:

- gdcmElement.h

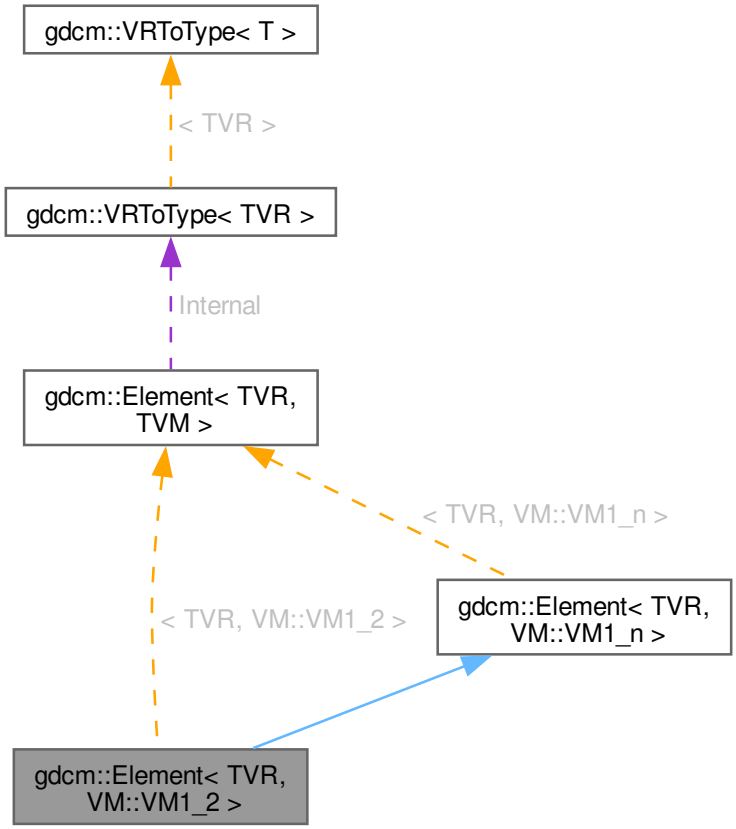
12.98 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM1_2 >:



Collaboration diagram for gdcm::Element< TVR, VM::VM1_2 >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent
- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< TVR, VM::VM1_n >

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const

- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetLength (int len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- void SetNoSwap (Value const &v)

12.98.1 Member Typedef Documentation

12.98.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM1_2 >::Parent
```

12.98.1.2 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.98.2 Member Function Documentation

12.98.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.98.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

12.98.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.98.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.98.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.98.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.98.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const [inline]
```

12.98.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const [inline]
```

12.98.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is) [inline]
```

12.98.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v) [inline]
```

12.98.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< TVR, VM::VM1_2 > const & de) [inline]
```

12.98.2.12 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_2 >::SetLength (
            int len) [inline]
```

References gdcm::Element< TVR, VM::VM1_n >::SetLength().

12.98.2.13 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

12.98.2.14 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.98.2.15 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.98.3 Member Data Documentation

12.98.3.1 Internal

```
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

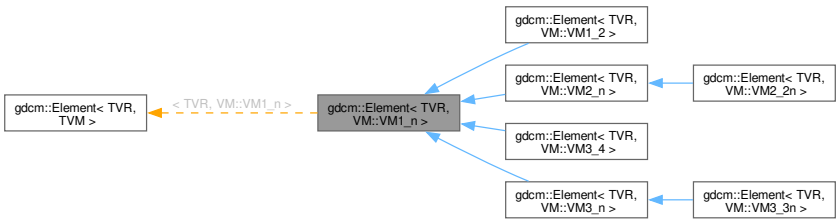
The documentation for this class was generated from the following file:

- gdcmElement.h

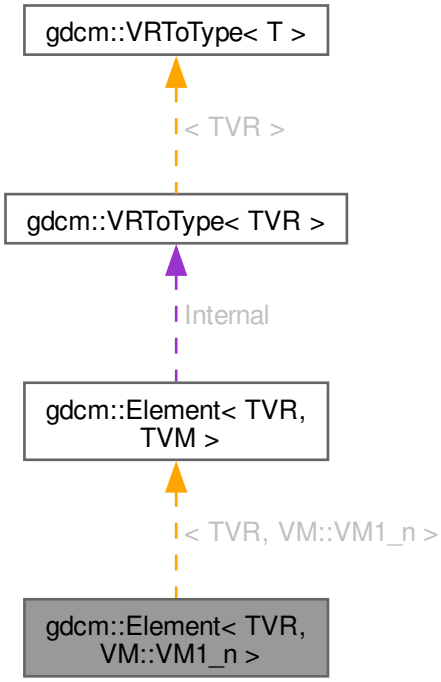
12.99 gdcm::Element< TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM1_n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM1_n >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Member Functions

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)

- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Protected Member Functions

- void SetNoSwap (Value const &v)

12.99.1 Member Typedef Documentation

12.99.1.1 Type

```
template<long long TVR>
typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type
```

12.99.2 Constructor & Destructor Documentation

12.99.2.1 Element() [1/2]

```
template<long long TVR>
gdcm::Element< TVR, VM::VM1_n >::Element ()  [inline], [explicit]
```

Referenced by Element(), and operator=().

12.99.2.2 ~Element()

```
template<long long TVR>
gdcm::Element< TVR, VM::VM1_n >::~Element ()  [inline]
```

12.99.2.3 Element() [2/2]

```
template<long long TVR>
gdcm::Element< TVR, VM::VM1_n >::Element (
            const Element< TVR, VM::VM1_n > & _val)  [inline]
```

References Element().

12.99.3 Member Function Documentation

12.99.3.1 GetAsDataElement()

```
template<long long TVR>
DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement () const  [inline]
```

References GetLength(), gdcm::DataElement::GetVR(), GetVR(), gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, gdcm::VR::VRASCII, and Write().

12.99.3.2 GetLength()

```
template<long long TVR>
unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength () const  [inline]
```

Referenced by GetAsDataElement(), Print(), Read(), Set(), SetNoSwap(), Write(), and WriteASCII().

12.99.3.3 GetValue() [1/2]

```
template<long long TVR>
VRToType< TVR >::Type & gdcm::Element< TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0)  [inline]
```

12.99.3.4 GetValue() [2/2]

```
template<long long TVR>
const VRToType< TVR >::Type & gdcm::Element< TVR, VM::VM1_n >::GetValue (
            unsigned int idx = 0) const  [inline]
```

Referenced by operator[ ]().

12.99.3.5 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.99.3.6 GetVM()

```
template<long long TVR>
VM gdcm::Element< TVR, VM::VM1_n >::GetVM ()  [inline], [static]
```

References gdcm::VM::VM1_n.

12.99.3.7 GetVR()

```
template<long long TVR>
VR gdcm::Element< TVR, VM::VM1_n >::GetVR ()  [inline], [static]
```

Referenced by GetAsDataElement().

12.99.3.8 operator=()

```
template<long long TVR>
Element & gdcm::Element< TVR, VM::VM1_n >::operator= (
            const Element< TVR, VM::VM1_n > & _val)  [inline]
```

References Element(), and SetArray().

12.99.3.9 operator[]()

```
template<long long TVR>
VRToType< TVR >::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (
            unsigned int idx) const  [inline]
```

References GetValue().

12.99.3.10 Print()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Print (
            std::ostream & _os) const  [inline]
```

References GetLength().

12.99.3.11 Read()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Read (
            std::istream & _is)  [inline]
```

References GetLength(), and Read().

Referenced by Read(), and Set().

12.99.3.12 Set()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Set (
            Value const & v)  [inline]
```

References gdcm::ByteValue::GetLength(), GetLength(), gdcm::ByteValue::GetPointer(), gdcm::ByteValue::GetVoidPointer(), Read(), SetArray(), and gdcm::VR::VRBINARY.

Referenced by SetFromDataElement().

12.99.3.13 SetArray()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetArray (
            const Type * array,
            unsigned long len,
            bool save = false)  [inline]
```

References SetLength().

Referenced by operator=(), Set(), and SetNoSwap().

12.99.3.14 SetFromDataElement()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement (
            DataElement< TVR, VM::VM1_n > const & de)  [inline]
```

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::GetValue(), gdcm::DataElement::GetVR(), gdcm::VR::INVALID, Set(), SetNoSwap(), and gdcm::VR::UN.

12.99.3.15 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetLength (
            unsigned long len)  [inline]
```

Referenced by SetArray(), gdcm::Element< TVR, VM::VM1_2 >::SetLength(), gdcm::Element< TVR, VM::VM2_n >::SetLength(), gdcm::Element< TVR, VM::VM3_4 >::SetLength(), and gdcm::Element< TVR, VM::VM3_n >::SetLength().

12.99.3.16 SetNoSwap()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

References gdcm::ByteValue::GetLength(), GetLength(), gdcm::ByteValue::GetPointer(), SetArray(), and gdcm::VR::VRBINARY.

Referenced by SetFromDataElement().

12.99.3.17 SetValue()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.99.3.18 Write()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::Write (
            std::ostream & _os) const  [inline]
```

References GetLength(), and Write().

Referenced by GetAsDataElement(), and Write().

12.99.3.19 WriteASCII()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (
            std::ostream & os) const  [inline]
```

References GetLength().

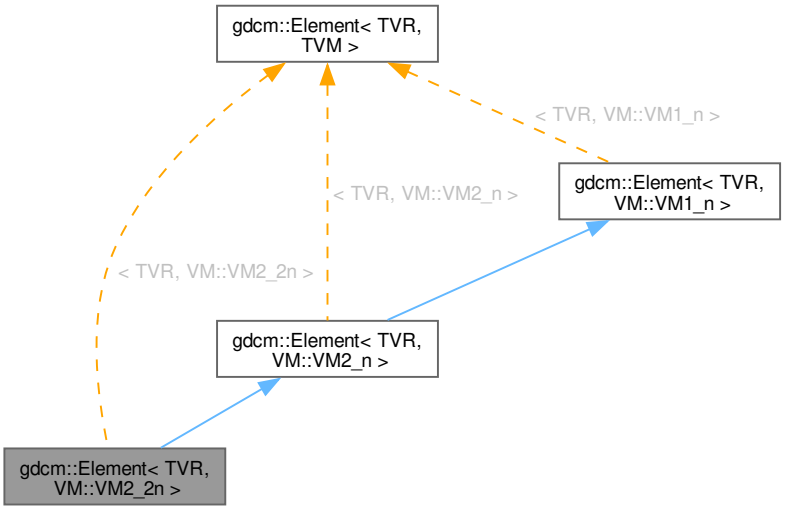
The documentation for this class was generated from the following file:

- gdcmElement.h

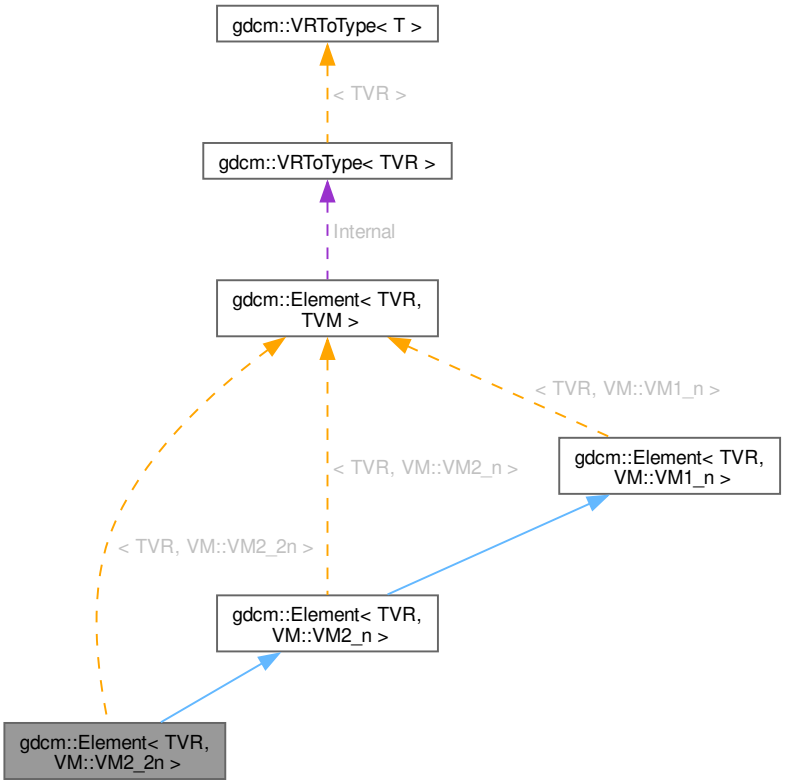
12.100 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM2_2n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM2_2n >:



Public Types

- typedef Element< TVR, VM::VM2_n > Parent
- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< TVR, VM::VM2_n >

- typedef Element< TVR, VM::VM1_n > Parent
- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< TVR, VM::VM1_n >

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetLength (int len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< TVR, VM::VM2_n >

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetLength (int len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< TVR, VM::VM2_n >

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Public Attributes inherited from gdcm::Element< TVR, VM::VM2_n >

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< TVR, VM::VM2_n >

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- void SetNoSwap (Value const &v)

12.100.1 Member Typedef Documentation

12.100.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent
```

12.100.1.2 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.100.2 Member Function Documentation

12.100.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.100.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

12.100.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.100.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.100.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM ()  [inline], [static]
```

12.100.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR ()  [inline], [static]
```

12.100.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const  [inline]
```

12.100.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const  [inline]
```

12.100.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is)  [inline]
```

12.100.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v)  [inline]
```

12.100.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< TVR, VM::VM2_2n > const & de)  [inline]
```

12.100.2.12 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM2_2n >::SetLength (
            int len)  [inline]
```

References gdcm::Element< TVR, VM::VM2_n >::SetLength().

12.100.2.13 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

12.100.2.14 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.100.2.15 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.100.3 Member Data Documentation

12.100.3.1 Internal

```
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

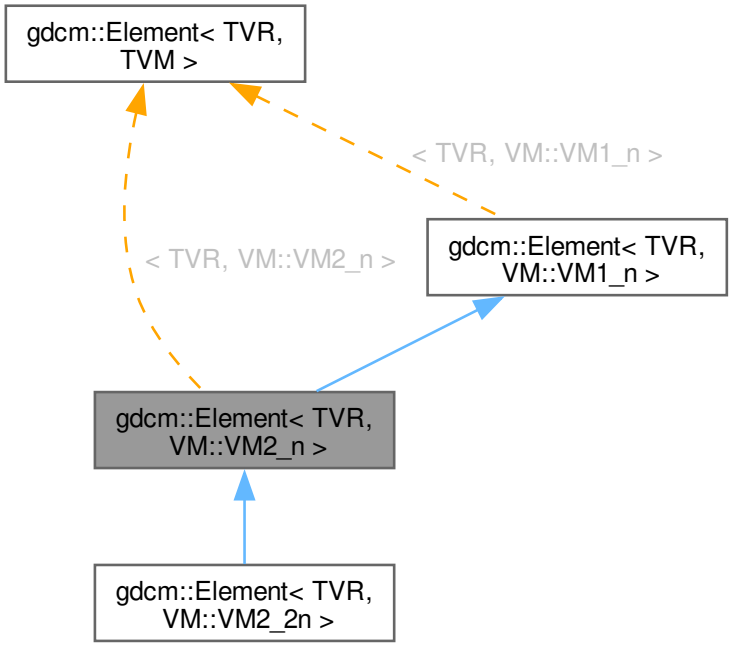
The documentation for this class was generated from the following file:

- gdcmElement.h

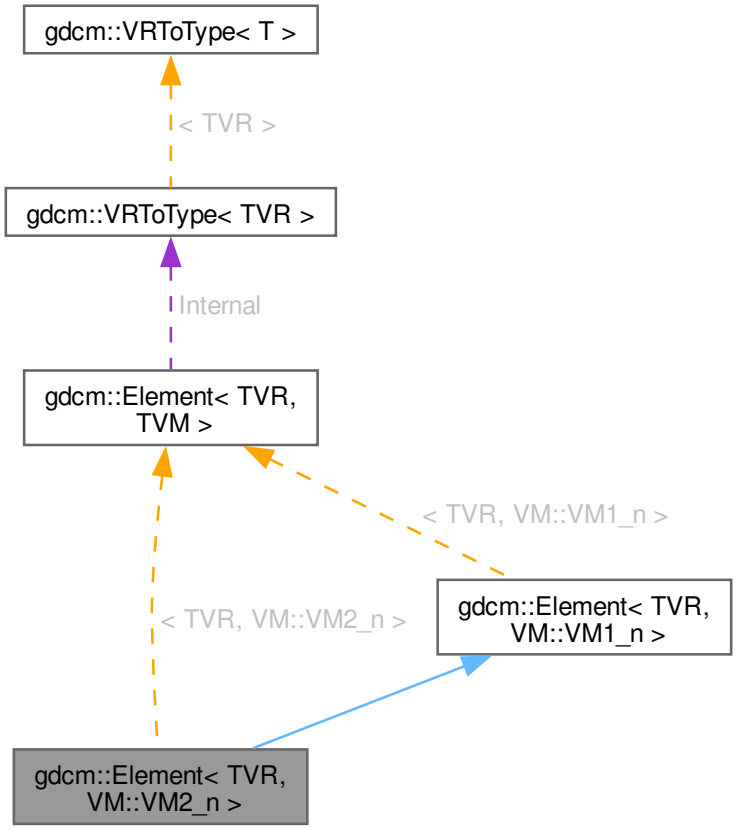
12.101 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM2_n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM2_n >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent
- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< TVR, VM::VM1_n >

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const

- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetLength (int len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- void SetNoSwap (Value const &v)

12.101.1 Member Typedef Documentation

12.101.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM2_n >::Parent
```

12.101.1.2 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.101.2 Member Function Documentation

12.101.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.101.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

12.101.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.101.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.101.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.101.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.101.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const [inline]
```

12.101.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const [inline]
```

12.101.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is) [inline]
```

12.101.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v) [inline]
```

12.101.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< TVR, VM::VM2_n > const & de) [inline]
```

12.101.2.12 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM2_n >::SetLength (
            int len) [inline]
```

References gdcm::Element< TVR, VM::VM1_n >::SetLength().

Referenced by gdcm::Element< TVR, VM::VM2_2n >::SetLength().

12.101.2.13 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

12.101.2.14 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.101.2.15 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.101.3 Member Data Documentation

12.101.3.1 Internal

```
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

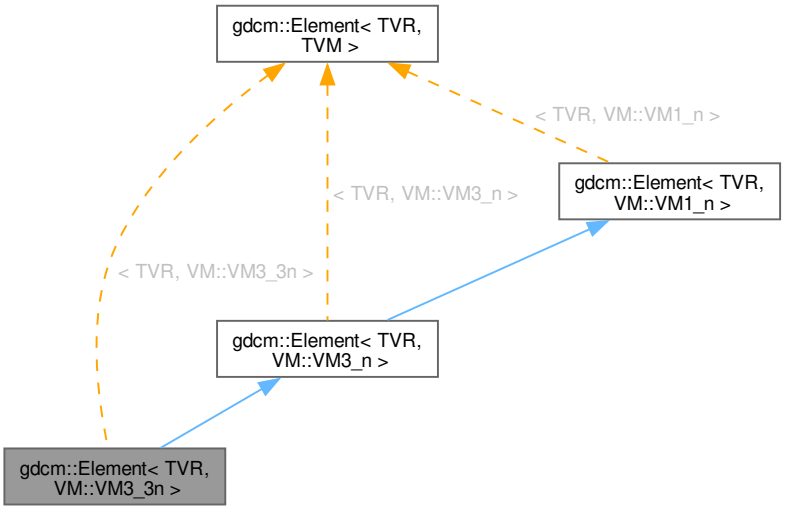
The documentation for this class was generated from the following file:

- gdcmElement.h

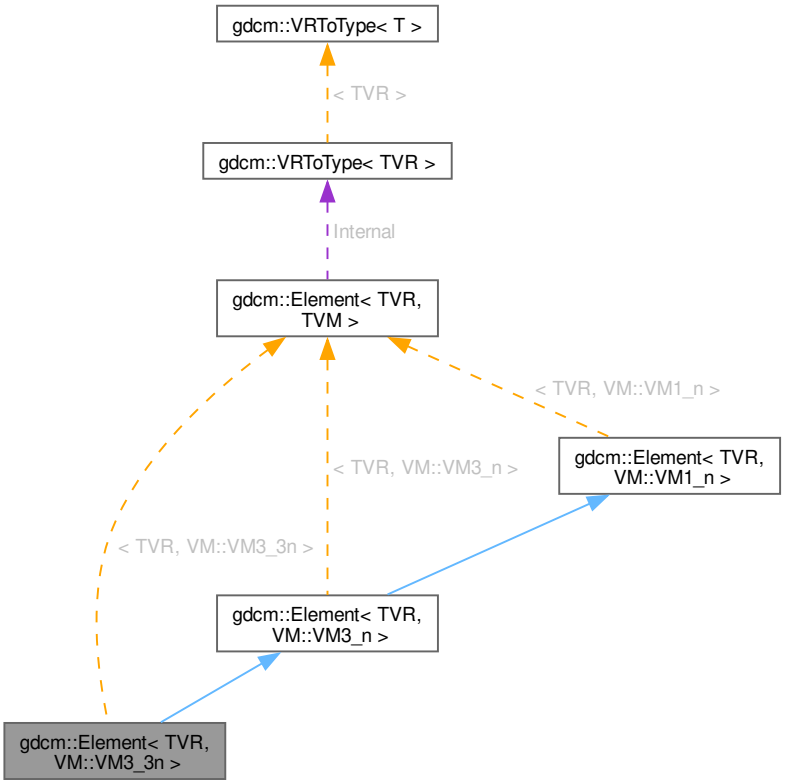
12.102 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3_3n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_3n >:



Public Types

- typedef Element< TVR, VM::VM3_n > Parent
- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< TVR, VM::VM3_n >

- typedef Element< TVR, VM::VM1_n > Parent
- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< TVR, VM::VM1_n >

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetLength (int len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< TVR, VM::VM3_n >

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetLength (int len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< TVR, VM::VM3_n >

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Public Attributes inherited from gdcm::Element< TVR, VM::VM3_n >

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< TVR, VM::VM3_n >

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- void SetNoSwap (Value const &v)

12.102.1 Member Typedef Documentation

12.102.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM3_n> gdcm::Element< TVR, VM::VM3_3n >::Parent
```

12.102.1.2 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.102.2 Member Function Documentation

12.102.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.102.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

12.102.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.102.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.102.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM ()  [inline], [static]
```

12.102.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR ()  [inline], [static]
```

12.102.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const  [inline]
```

12.102.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const  [inline]
```

12.102.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is)  [inline]
```

12.102.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v)  [inline]
```

12.102.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< TVR, VM::VM3_3n > const & de)  [inline]
```

12.102.2.12 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_3n >::SetLength (
            int len)  [inline]
```

References gdcm::Element< TVR, VM::VM3_n >::SetLength().

12.102.2.13 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

12.102.2.14 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.102.2.15 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.102.3 Member Data Documentation

12.102.3.1 Internal

```
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

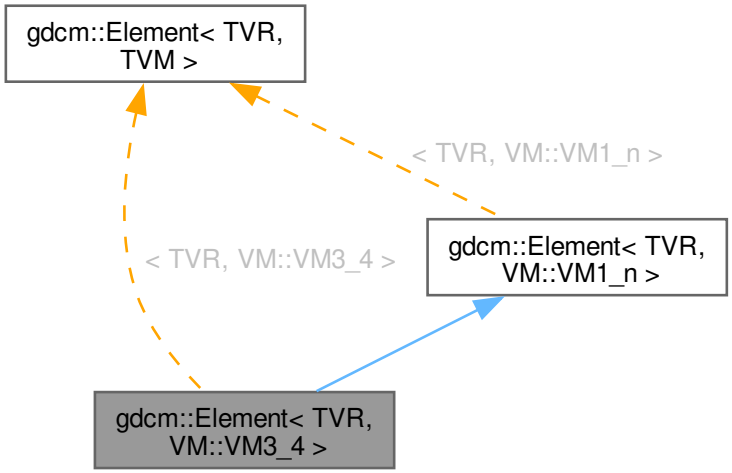
The documentation for this class was generated from the following file:

- gdcmElement.h

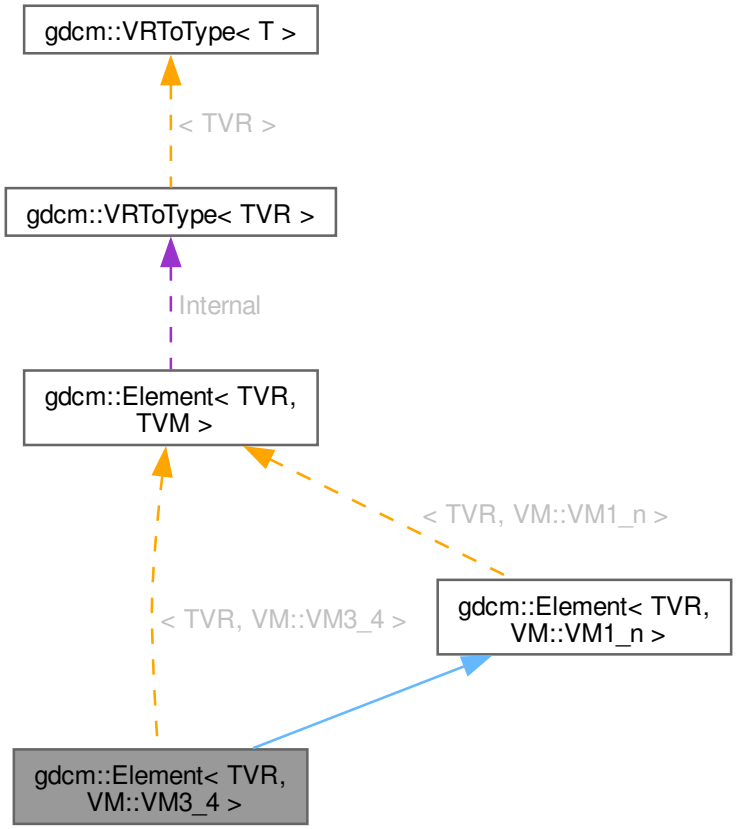
12.103 gdcm::Element< TVR, VM::VM3_4 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3_4 >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_4 >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent
- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< TVR, VM::VM1_n >

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const

- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetLength (int len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- void SetNoSwap (Value const &v)

12.103.1 Member Typedef Documentation

12.103.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_4 >::Parent
```

12.103.1.2 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.103.2 Member Function Documentation

12.103.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.103.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

12.103.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.103.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.103.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.103.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.103.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const [inline]
```

12.103.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const [inline]
```

12.103.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is) [inline]
```

12.103.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v) [inline]
```

12.103.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< TVR, VM::VM3_4 > const & de) [inline]
```

12.103.2.12 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_4 >::SetLength (
            int len) [inline]
```

References gdcm::Element< TVR, VM::VM1_n >::SetLength().

12.103.2.13 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

12.103.2.14 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.103.2.15 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.103.3 Member Data Documentation

12.103.3.1 Internal

```
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

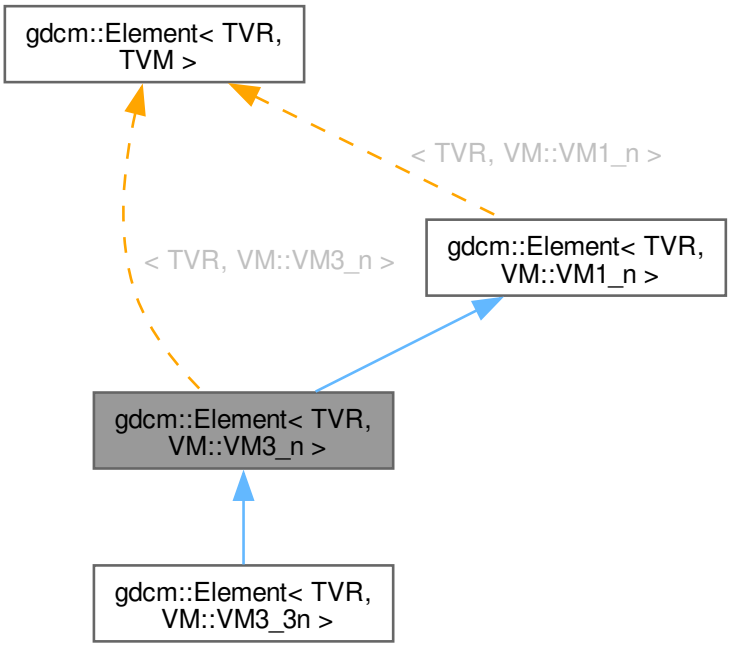
The documentation for this class was generated from the following file:

- gdcmElement.h

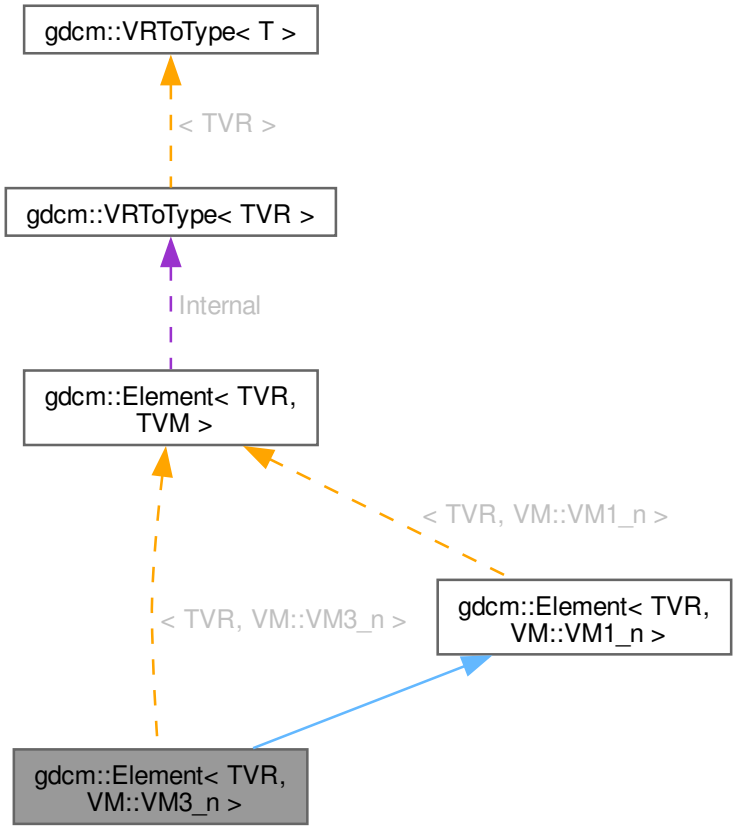
12.104 gdcm::Element< TVR, VM::VM3_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM3_n >:



Collaboration diagram for gdcm::Element< TVR, VM::VM3_n >:



Public Types

- typedef Element< TVR, VM::VM1_n > Parent
- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< TVR, VM::VM1_n >

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const

- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetLength (int len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- Element ()
- Element (const Element &_val)
- ~Element ()
- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- Element & operator= (const Element &_val)
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetArray (const Type *array, unsigned long len, bool save=false)
- void SetFromDataElement (DataElement const &de)
- void SetLength (unsigned long len)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const
- void WriteASCII (std::ostream &os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< TVR, VM::VM1_n >

- void SetNoSwap (Value const &v)

12.104.1 Member Typedef Documentation

12.104.1.1 Parent

```
template<long long TVR>
typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_n >::Parent
```

12.104.1.2 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.104.2 Member Function Documentation

12.104.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.104.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

12.104.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.104.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.104.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.104.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.104.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const [inline]
```

12.104.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const [inline]
```

12.104.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is) [inline]
```

12.104.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v) [inline]
```

12.104.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< TVR, VM::VM3_n > const & de) [inline]
```

12.104.2.12 SetLength()

```
template<long long TVR>
void gdcm::Element< TVR, VM::VM3_n >::SetLength (
            int len) [inline]
```

References gdcm::Element< TVR, VM::VM1_n >::SetLength().

Referenced by gdcm::Element< TVR, VM::VM3_3n >::SetLength().

12.104.2.13 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

12.104.2.14 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.104.2.15 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.104.3 Member Data Documentation

12.104.3.1 Internal

```
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

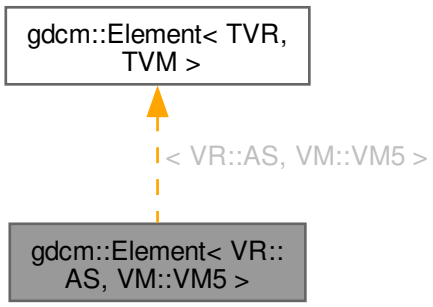
The documentation for this class was generated from the following file:

- gdcmElement.h

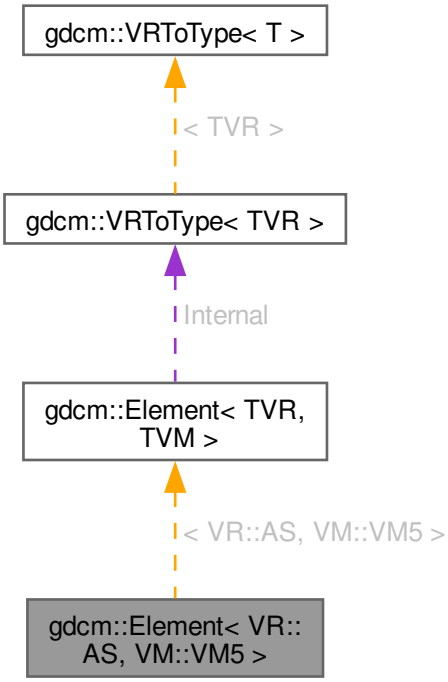
12.105 gdcm::Element< VR::AS, VM::VM5 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::AS, VM::VM5 >:



Collaboration diagram for gdcm::Element< VR::AS, VM::VM5 >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- char Internal [VMToLength< VM::VM5 >::Length ∗sizeof(VRToType< VR::AS >::Type)]

Protected Member Functions

- void SetNoSwap (Value const &v)

12.105.1 Member Typedef Documentation

12.105.1.1 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.105.2 Member Function Documentation

12.105.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.105.2.2 GetLength()

```
unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength () const  [inline]
```

12.105.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.105.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.105.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM () [inline], [static]
```

12.105.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR () [inline], [static]
```

12.105.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const [inline]
```

12.105.2.8 Print()

```
void gdcm::Element< VR::AS, VM::VM5 >::Print (
            std::ostream & _os) const [inline]
```

References Internal.

12.105.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is) [inline]
```

12.105.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v) [inline]
```

12.105.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< VR::AS, VM::VM5 > const & de) [inline]
```

12.105.2.12 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v) [inline], [protected]
```

12.105.2.13 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.105.2.14 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.105.3 Member Data Documentation

12.105.3.1 Internal

```
char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType<
VR::AS >::Type)]
```

Referenced by Print().

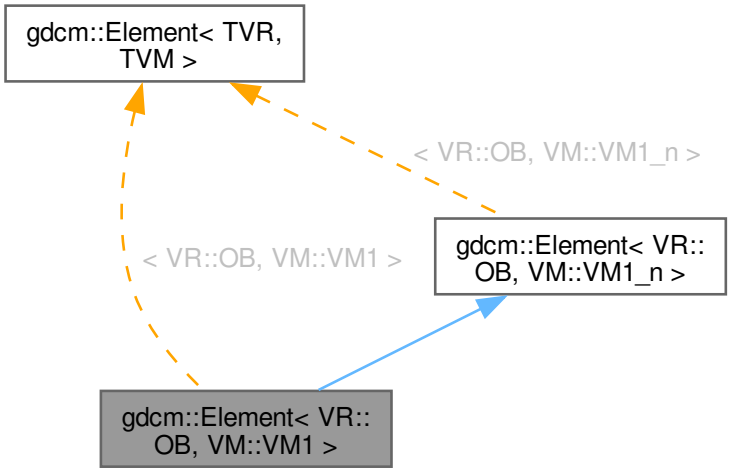
The documentation for this class was generated from the following file:

- gdcmElement.h

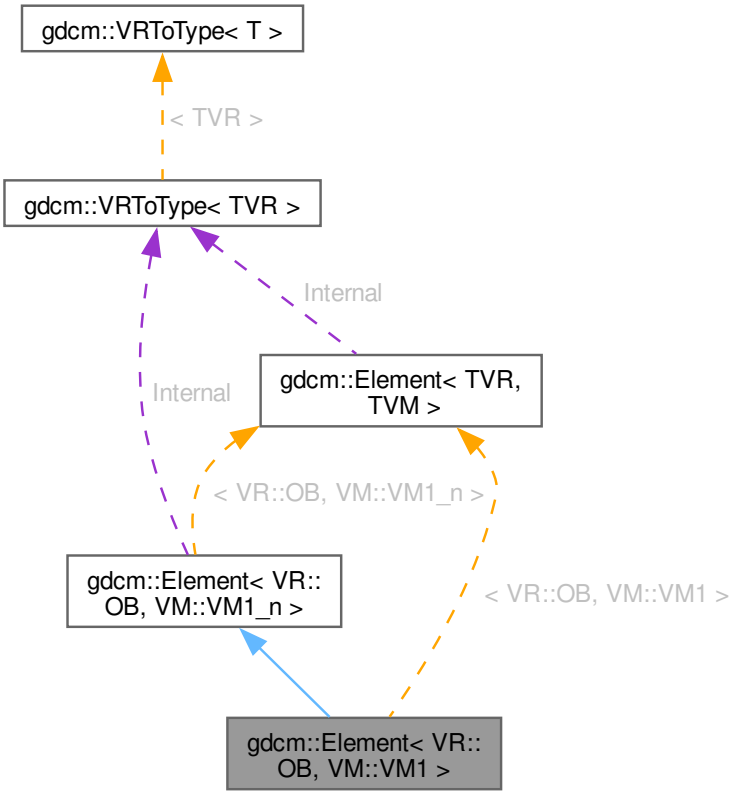
12.106 gdcm::Element< VR::OB, VM::VM1 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OB, VM::VM1 >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< VR::OB, VM::VM1_n >

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const

- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< VR::OB, VM::VM1_n >

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< VR::OB, VM::VM1_n >

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Public Attributes inherited from gdcm::Element< VR::OB, VM::VM1_n >

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< VR::OB, VM::VM1_n >

- void SetNoSwap (Value const &v)

12.106.1 Member Typedef Documentation

12.106.1.1 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.106.2 Member Function Documentation

12.106.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.106.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

12.106.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.106.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.106.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM ()  [inline], [static]
```

12.106.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR ()  [inline], [static]
```

12.106.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const  [inline]
```

12.106.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const  [inline]
```

12.106.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is)  [inline]
```

12.106.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v)  [inline]
```

12.106.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< VR::OB, VM::VM1 > const & de)  [inline]
```

12.106.2.12 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

12.106.2.13 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.106.2.14 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.106.3 Member Data Documentation

12.106.3.1 Internal

```
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

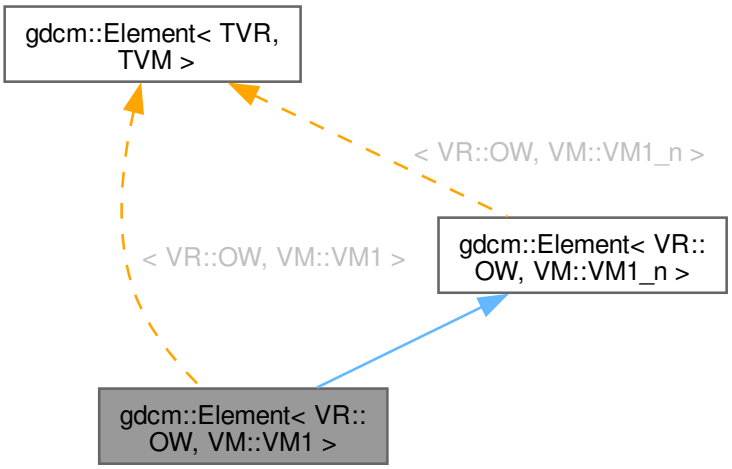
The documentation for this class was generated from the following file:

- gdcmElement.h

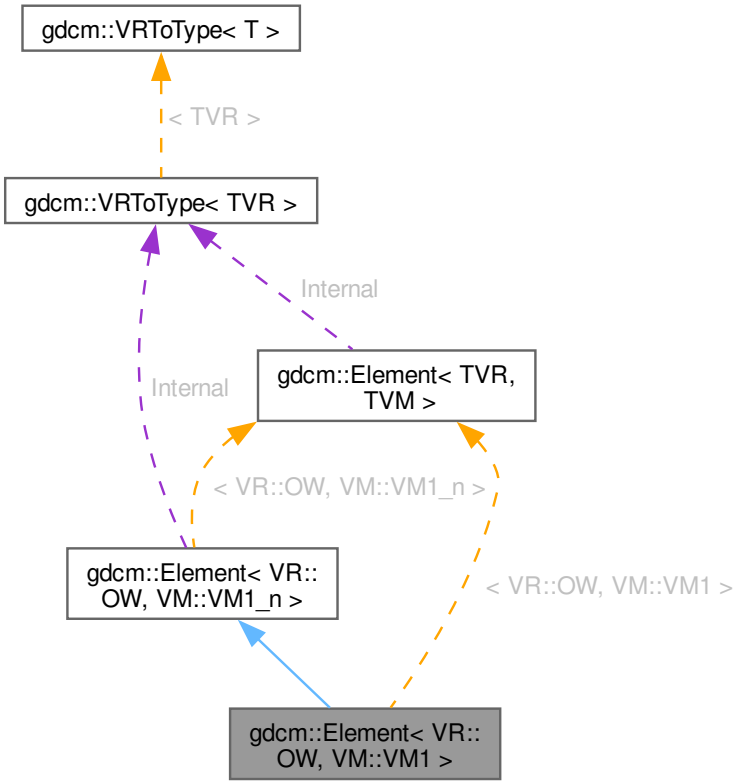
12.107 gdcm::Element< VR::OW, VM::VM1 > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< VR::OW, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OW, VM::VM1 >:



Public Types

- typedef VRToType< TVR >::Type Type

Public Types inherited from gdcm::Element< VR::OW, VM::VM1_n >

- typedef VRToType< TVR >::Type Type

Public Member Functions

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const

- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Public Member Functions inherited from gdcm::Element< VR::OW, VM::VM1_n >

- DataElement GetAsDataElement () const
- unsigned long GetLength () const
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- const VRToType< TVR >::Type * GetValues () const
- VRToType< TVR >::Type operator[ ] (unsigned int idx) const
- void Print (std::ostream &_os) const
- void Read (std::istream &_is)
- void Set (Value const &v)
- void SetFromDataElement (DataElement const &de)
- void SetValue (typename VRToType< TVR >::Type v, unsigned int idx=0)
- void Write (std::ostream &_os) const

Static Public Member Functions

- static VM GetVM ()
- static VR GetVR ()

Static Public Member Functions inherited from gdcm::Element< VR::OW, VM::VM1_n >

- static VM GetVM ()
- static VR GetVR ()

Public Attributes

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Public Attributes inherited from gdcm::Element< VR::OW, VM::VM1_n >

- VRToType< TVR >::Type Internal [VMToLength< TVM >::Length]

Protected Member Functions

- void SetNoSwap (Value const &v)

Protected Member Functions inherited from gdcm::Element< VR::OW, VM::VM1_n >

- void SetNoSwap (Value const &v)

12.107.1 Member Typedef Documentation

12.107.1.1 Type

```
typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type
```

12.107.2 Member Function Documentation

12.107.2.1 GetAsDataElement()

```
DataElement gdcm::Element< TVR, TVM >::GetAsDataElement () const  [inline]
```

12.107.2.2 GetLength()

```
unsigned long gdcm::Element< TVR, TVM >::GetLength () const  [inline]
```

12.107.2.3 GetValue()

```
const VRToType< TVR >::Type & gdcm::Element< TVR, TVM >::GetValue (
            unsigned int idx = 0) const  [inline]
```

12.107.2.4 GetValues()

```
const VRToType< TVR >::Type * gdcm::Element< TVR, TVM >::GetValues () const  [inline]
```

12.107.2.5 GetVM()

```
VM gdcm::Element< TVR, TVM >::GetVM ()  [inline], [static]
```

12.107.2.6 GetVR()

```
VR gdcm::Element< TVR, TVM >::GetVR ()  [inline], [static]
```

12.107.2.7 operator[]()

```
VRToType< TVR >::Type gdcm::Element< TVR, TVM >::operator[] (
            unsigned int idx) const  [inline]
```

12.107.2.8 Print()

```
void gdcm::Element< TVR, TVM >::Print (
            std::ostream & _os) const  [inline]
```

12.107.2.9 Read()

```
void gdcm::Element< TVR, TVM >::Read (
            std::istream & _is)  [inline]
```

12.107.2.10 Set()

```
void gdcm::Element< TVR, TVM >::Set (
            Value const & v)  [inline]
```

12.107.2.11 SetFromDataElement()

```
void gdcm::Element< TVR, TVM >::SetFromDataElement (
            DataElement< VR::OW, VM::VM1 > const & de)  [inline]
```

12.107.2.12 SetNoSwap()

```
void gdcm::Element< TVR, TVM >::SetNoSwap (
            Value const & v)  [inline], [protected]
```

12.107.2.13 SetValue()

```
void gdcm::Element< TVR, TVM >::SetValue (
            typename VRToType< TVR >::Type v,
            unsigned int idx = 0)  [inline]
```

12.107.2.14 Write()

```
void gdcm::Element< TVR, TVM >::Write (
            std::ostream & _os) const  [inline]
```

12.107.3 Member Data Documentation

12.107.3.1 Internal

```
VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]
```

The documentation for this class was generated from the following file:

- gdcmElement.h

12.108 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference

A class which is used to produce compile errors for an invalid combination of template parameters.

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::ElementDisableCombinations< TVR, TVM >:

12.108.1 Detailed Description

template<long long TVR, int TVM>
class gdcm::ElementDisableCombinations< TVR, TVM >

A class which is used to produce compile errors for an invalid combination of template parameters.

Invalid combinations have specialized declarations with no definition.

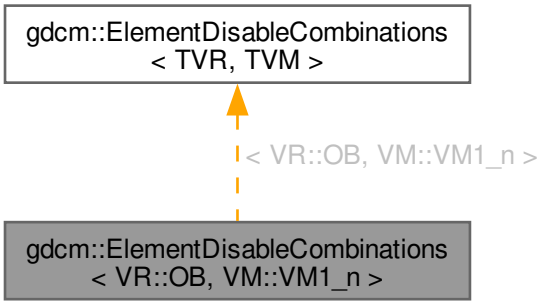
The documentation for this class was generated from the following file:

- gdcmElement.h

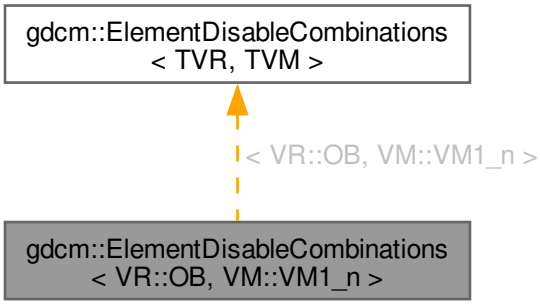
12.109 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >:



Collaboration diagram for gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >:



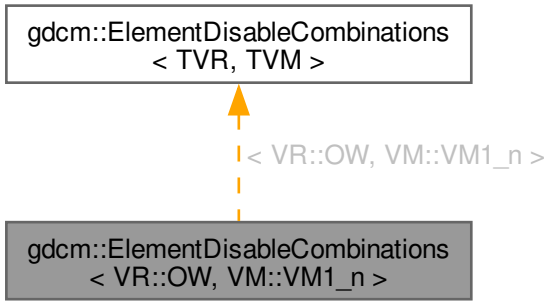
The documentation for this class was generated from the following file:

- gdcmElement.h

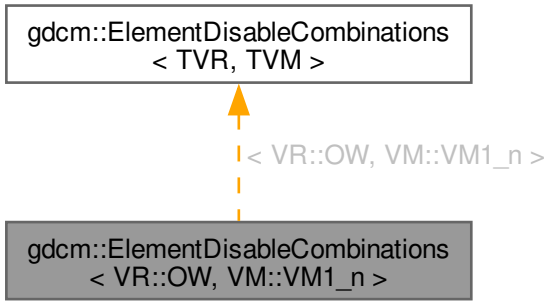
12.110 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >:



Collaboration diagram for gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >:



The documentation for this class was generated from the following file:

- gdcmElement.h

12.111 gdcm::EmptyMaskGenerator Class Reference

EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory.

```
#include <gdcmEmptyMaskGenerator.h>
```

Public Types

- enum SOPClassUIDMode {
  UseOriginalSOPClassUID = 0 ,
  UseGrayscaleSecondaryImageStorage }

Public Member Functions

- EmptyMaskGenerator ()
- ~EmptyMaskGenerator ()
- bool Execute ()

  Main loop.
- void SetInputDirectory (const char *dirname)

  Specify input directory.
- void SetOutputDirectory (const char *dirname)

  Specify output directory.
- void SetSOPClassUIDMode (SOPClassUIDMode mode)

12.111.1 Detailed Description

EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory.

The class allow two mode of operations:

- UseOriginalSOPClassUID
- UseGrayscaleSecondaryImageStorage

UseOriginalSOPClassUID is the mode where original attributes are copied from the original DICOM instance.

UseGrayscaleSecondaryImageStorage is the mode where attributes are generated so as to create a Multiframe↩ GrayscaleByteSecondaryCaptureImageStorage (MultiframeGrayscaleWordSecondaryCaptureImageStorage) instance.

In both mode:

- the Study references (StudyInstanceUID and StudyID) are preserved.
- the PatientID reference is preserved.
- the Image Type attribute will be setup so that the fourth element is set to 'MASK'.
- a new Series Instance UID is generated. It is thus required to run the process over all files using the same input Series Instance UID so that a proper mapping from the old Series UID is done to the new one. Since a new Series Instance UID is generated, there is no sense to preserve the original Frame of Reference UID, although it would have made sense here.

Examples

EmptyMask.cxx.

12.111.2 Member Enumeration Documentation

12.111.2.1 SOPClassUIDMode

```
enum gdcm::EmptyMaskGenerator::SOPClassUIDMode
```

Enumerator

UseOriginalSOPClassUID	
UseGrayscaleSecondaryImageStorage	

12.111.3 Constructor & Destructor Documentation

12.111.3.1 EmptyMaskGenerator()

```
gdcm::EmptyMaskGenerator::EmptyMaskGenerator ()
```

12.111.3.2 ~EmptyMaskGenerator()

```
gdcm::EmptyMaskGenerator::~EmptyMaskGenerator ()
```

12.111.4 Member Function Documentation

12.111.4.1 Execute()

```
bool gdcm::EmptyMaskGenerator::Execute ()
```

Main loop.

Examples

EmptyMask.cxx.

12.111.4.2 SetInputDirectory()

```
void gdcm::EmptyMaskGenerator::SetInputDirectory (
            const char * dirname)
```

Specify input directory.

Examples

EmptyMask.cxx.

12.111.4.3 SetOutputDirectory()

```
void gdcm::EmptyMaskGenerator::SetOutputDirectory (
            const char * dirname)
```

Specify output directory.

Examples

EmptyMask.cxx.

12.111.4.4 SetSOPClassUIDMode()

```
void gdcm::EmptyMaskGenerator::SetSOPClassUIDMode (
            SOPClassUIDMode mode)
```

Select generation of SOP Class UID method: Default is UseOriginalSOPClassUID

Examples

EmptyMask.cxx.

The documentation for this class was generated from the following file:

- gdcmEmptyMaskGenerator.h

12.112 gdcm::EncapsulatedDocument Class Reference

EncapsulatedDocument.

```
#include <gdcmEncapsulatedDocument.h>
```

Public Member Functions

- EncapsulatedDocument ()=default

12.112.1 Detailed Description

EncapsulatedDocument.

12.112.2 Constructor & Destructor Documentation

12.112.2.1 EncapsulatedDocument()

```
gdcm::EncapsulatedDocument::EncapsulatedDocument () [default]
```

The documentation for this class was generated from the following file:

- gdcmEncapsulatedDocument.h

12.113 gdcm::EncodingImplementation< T > Class Template Reference

EncodingImplementation.

Inheritance diagram for gdcm::EncodingImplementation< T >:

12.113.1 Detailed Description

template< long long T >
class gdcm::EncodingImplementation< T >

EncodingImplementation.

Note

TODO

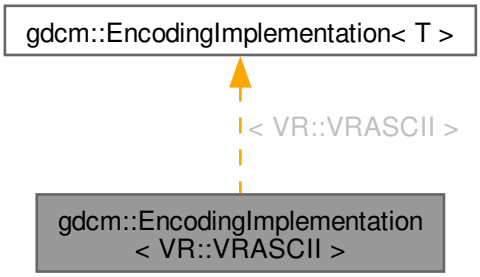
The documentation for this class was generated from the following file:

- gdcmElement.h

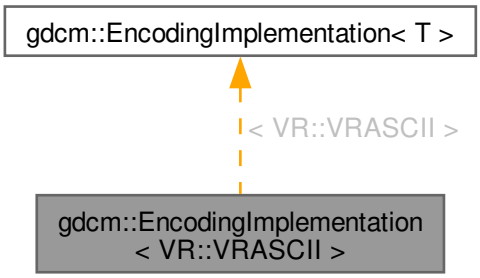
12.114 gdcm::EncodingImplementation< VR::VRASCII > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::EncodingImplementation< VR::VRASCII >:



Collaboration diagram for gdcm::EncodingImplementation< VR::VRASCII >:



Public Member Functions

- template<> void Write (const double *data, unsigned long length, std::ostream &_os)

Static Public Member Functions

- template<typename T>
  static void Read (T *data, unsigned long length, std::istream &_is)
- template<typename T>
  static void ReadComputeLength (T *data, unsigned int &length, std::istream &_is)
- template<typename T>
  static void ReadNoSwap (T *data, unsigned long length, std::istream &_is)
- template<typename T>
  static void Write (const T *data, unsigned long length, std::ostream &_os)

12.114.1 Member Function Documentation

12.114.1.1 Read()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRASCII >::Read (
            T * data,
            unsigned long length,
            std::istream & _is)  [inline], [static]
```

Referenced by ReadNoSwap().

12.114.1.2 ReadComputeLength()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (
            T * data,
            unsigned int & length,
            std::istream & _is)  [inline], [static]
```

References gdcm::backslash().

12.114.1.3 ReadNoSwap()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (
            T * data,
            unsigned long length,
            std::istream & _is)  [inline], [static]
```

References Read().

12.114.1.4 Write() [1/2]

```
template<>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
            const double * data,
            unsigned long length,
            std::ostream & _os)  [inline]
```

References gdcm::x16printf().

12.114.1.5 Write() [2/2]

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRASCII >::Write (
            const T * data,
            unsigned long length,
            std::ostream & _os)  [inline], [static]
```

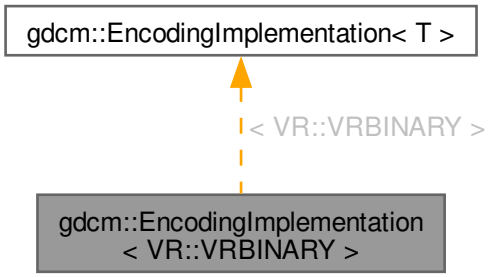
The documentation for this class was generated from the following file:

- gdcmElement.h

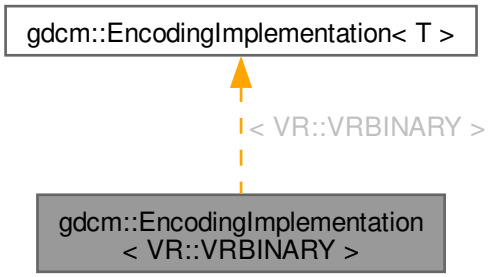
12.115 gdcm::EncodingImplementation< VR::VRBINARY > Class Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::EncodingImplementation< VR::VRBINARY >:



Collaboration diagram for gdcm::EncodingImplementation< VR::VRBINARY >:



Static Public Member Functions

- template<typename T>
  static void Read (T *data, unsigned long length, std::istream &_is)
- template<typename T>
  static void ReadComputeLength (T *data, unsigned int &length, std::istream &_is)
- template<typename T>
  static void ReadNoSwap (T *data, unsigned long length, std::istream &_is)
- template<typename T>
  static void Write (const T *data, unsigned long length, std::ostream &_os)

12.115.1 Member Function Documentation

12.115.1.1 Read()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRBINARY >::Read (
            T * data,
            unsigned long length,
            std::istream & _is)  [inline], [static]
```

References gdcm::SwapperNoOp::SwapArray().

12.115.1.2 ReadComputeLength()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (
            T * data,
            unsigned int & length,
            std::istream & _is)  [inline], [static]
```

12.115.1.3 ReadNoSwap()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (
            T * data,
            unsigned long length,
            std::istream & _is)  [inline], [static]
```

12.115.1.4 Write()

```
template<typename T>
void gdcm::EncodingImplementation< VR::VRBINARY >::Write (
            const T * data,
            unsigned long length,
            std::ostream & _os)  [inline], [static]
```

References gdcm::SwapperNoOp::Swap().

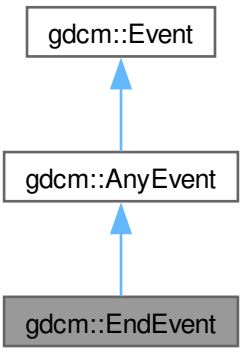
The documentation for this class was generated from the following file:

- gdcmElement.h

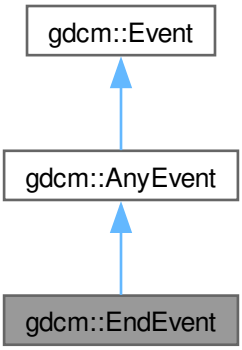
12.116 gdcm::EndEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::EndEvent:



Collaboration diagram for gdcm::EndEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

- gdcmEvent.h

12.117 gdcm::EnumeratedValues Class Reference

Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

```
#include <gdcmEnumeratedValues.h>
```

Public Member Functions

- EnumeratedValues ()=default

12.117.1 Detailed Description

Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

1. Patient Sex (0010, 0040) is an example of a Data Element having Enumerated Values. It is defined to have a Value that is either "M", "F", or "O" (see PS 3.3). No other Value shall be given to this Data Element.
2. Future modifications of this standard may add to the set of allowed values for Data Elements with Enumerated Values. Such additions by themselves may or may not require a change in SOP Class UIDs, depending on the semantics of the Data Element.

12.117.2 Constructor & Destructor Documentation

12.117.2.1 EnumeratedValues()

```
gdcm::EnumeratedValues::EnumeratedValues () [default]
```

The documentation for this class was generated from the following file:

- gdcmEnumeratedValues.h

12.118 gdcm::EquipmentManufacturer Class Reference

```
#include <gdcmEquipmentManufacturer.h>
```

Public Types

- enum Type {
  UNKNOWN = 0 ,
  FUJI ,
  GEMS ,
  HITACHI ,
  KODAK ,
  MARCONI ,
  PMS ,
  SIEMENS ,
  TOSHIBA ,
  AGFA ,
  SAMSUNG ,
  UIH }

Static Public Member Functions

- static Type Compute (DataSet const &ds)
- static const char * TypeToString (Type type)

12.118.1 Detailed Description

The intent is for private tags handling. This class is not meant to handle all possible vendors in the world, simply those well known where we intend to read private tags afterwards (typically SIEMENS+CSA, GEMS+PDB ...)

12.118.2 Member Enumeration Documentation

12.118.2.1 Type

```
enum gdcm::EquipmentManufacturer::Type
```

Enumerator

UNKNOWN	
FUJI	
GEMS	
HITACHI	
KODAK	
MARCONI	
PMS	
SIEMENS	
TOSHIBA	
AGFA	
SAMSUNG	
UIH	

12.118.3 Member Function Documentation

12.118.3.1 Compute()

```
Type gdcm::EquipmentManufacturer::Compute (
            DataSet const & ds)  [static]
```

12.118.3.2 TypeToString()

```
const char * gdcm::EquipmentManufacturer::TypeToString (
            Type type)  [static]
```

The documentation for this class was generated from the following file:

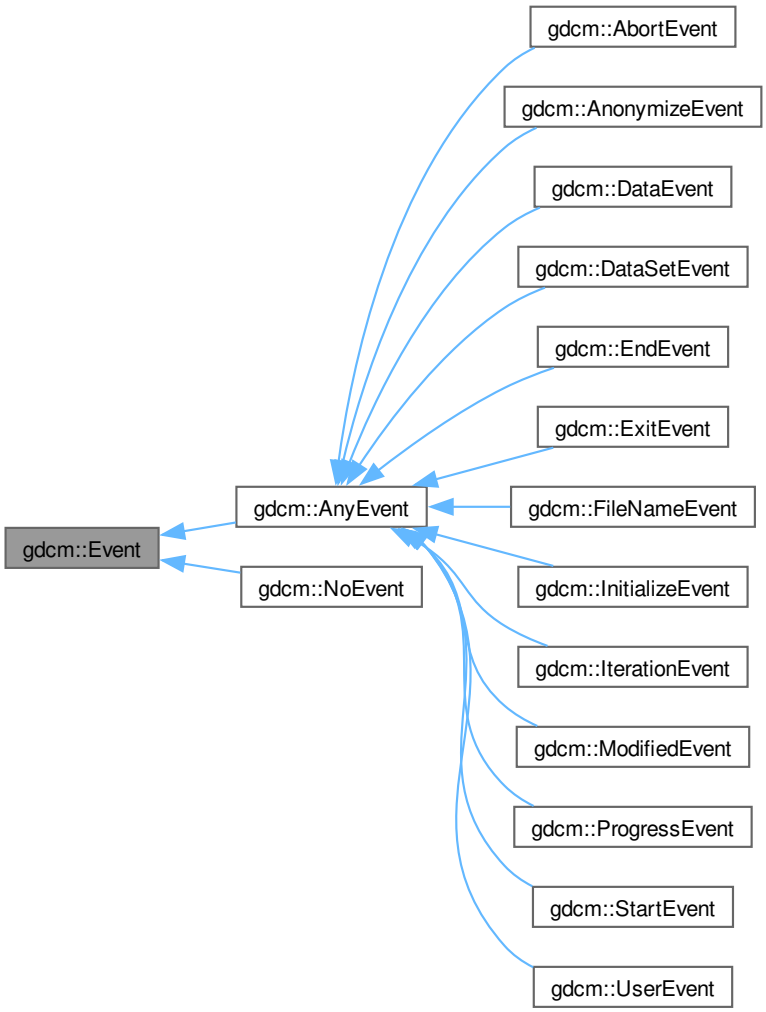
- gdcmEquipmentManufacturer.h

12.119 gdcm::Event Class Reference

superclass for callback/observer methods

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::Event:



Public Member Functions

- Event ()
- Event (const Event &)
- virtual ~Event ()

- virtual bool CheckEvent (const Event ∗) const =0
- virtual const char ∗ GetEventName () const =0
- virtual Event ∗ MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

12.119.1 Detailed Description

superclass for callback/observer methods

See also

Command Subject

Examples

BasicAnonymizer.cs, Cleaner.cs, ClinicalTrialIdentificationWorkflow.cs, ScanDirectory.cs, and SimpleScanner.cxx.

12.119.2 Constructor & Destructor Documentation

12.119.2.1 Event() [1/2]

```
gdcm::Event::Event ()
```

Referenced by Event(), CheckEvent(), MakeObject(), and operator=().

12.119.2.2 ∼Event()

```
virtual gdcm::Event::~Event ()  [virtual]
```

12.119.2.3 Event() [2/2]

```
gdcm::Event::Event (
            const Event & )
```

References Event().

12.119.3 Member Function Documentation

12.119.3.1 CheckEvent()

```
virtual bool gdcm::Event::CheckEvent (
            const Event * ) const  [pure virtual]
```

Check if given event matches or derives from this event.

References Event().

12.119.3.2 GetEventName()

```
virtual const char * gdcm::Event::GetEventName () const  [pure virtual]
```

Return the StringName associated with the event.

Implemented in gdcm::AnonymizeEvent, gdcm::DataEvent, gdcm::DataSetEvent, gdcm::FileNameEvent, and gdcm::ProgressEvent.

Examples

BasicAnonymizer.cs, Cleaner.cs, ClinicalTrialIdentificationWorkflow.cs, and ScanDirectory.cs.

12.119.3.3 MakeObject()

```
virtual Event * gdcm::Event::MakeObject () const  [pure virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implemented in gdcm::AnonymizeEvent, gdcm::DataEvent, gdcm::DataSetEvent, gdcm::FileNameEvent, and gdcm::ProgressEvent.

References Event().

12.119.3.4 operator=()

```
void gdcm::Event::operator= (
            const Event & )  [delete]
```

References Event().

12.119.3.5 Print()

```
virtual void gdcm::Event::Print (
            std::ostream & os) const  [virtual]
```

Print Event information. This method can be overridden by specific Event subtypes. The default is to print out the type of the event.

Referenced by gdcm::operator<<().

The documentation for this class was generated from the following file:

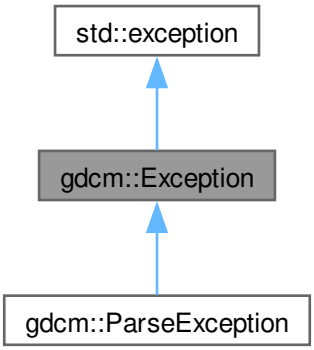
- gdcmEvent.h

12.120 gdcm::Exception Class Reference

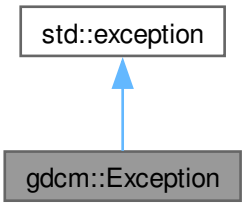
Exception.

```
#include <gdcmException.h>
```

Inheritance diagram for gdcm::Exception:



Collaboration diagram for gdcm::Exception:



Public Member Functions

- Exception (const char ∗desc="None", const char ∗file=__FILE__, unsigned int lineNumber=__LINE__, const char ∗func="")
- ∼Exception () override throw ()
- const char ∗ GetDescription () const

  Return the Description.
- const char ∗ what () const override throw ()

  what implementation

12.120.1 Detailed Description

Exception.

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, FileChangeTS.cs, and FileChangeTSLossy.cs.

12.120.2 Constructor & Destructor Documentation

12.120.2.1 Exception()

```
gdcm::Exception::Exception (
            const char * desc = "None",
            const char * file = __FILE__,
            unsigned int lineNumber = __LINE__,
            const char * func = "")  [inline], [explicit]
```

Explicit constructor, initializing the description and the text returned by what().

Note

The last parameter is ignored for the time being. It may be used to specify the function where the exception was thrown.

Referenced by gdcm::ParseException::ParseException().

12.120.2.2 ~Exception()

```
gdcm::Exception::~Exception () throw ( )   [inline], [override]
```

12.120.3 Member Function Documentation

12.120.3.1 GetDescription()

```
const char * gdcm::Exception::GetDescription () const  [inline]
```

Return the Description.

Referenced by gdcm::SequenceOfItems::Read().

12.120.3.2 what()

```
const char * gdcm::Exception::what () const throw ( )   [inline], [override]
```

what implementation

Referenced by gdcm::SequenceOfFragments::ReadValue().

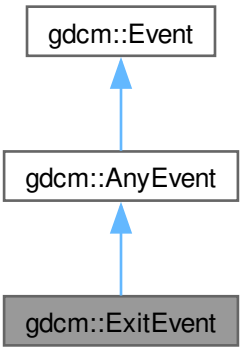
The documentation for this class was generated from the following file:

- gdcmException.h

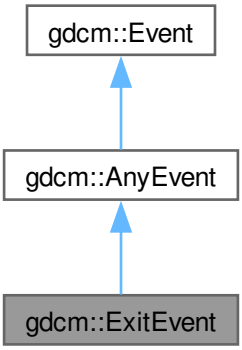
12.121 gdcm::ExitEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ExitEvent:



Collaboration diagram for gdcm::ExitEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

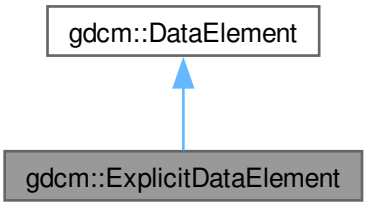
- gdcmEvent.h

12.122 gdcm::ExplicitDataElement Class Reference

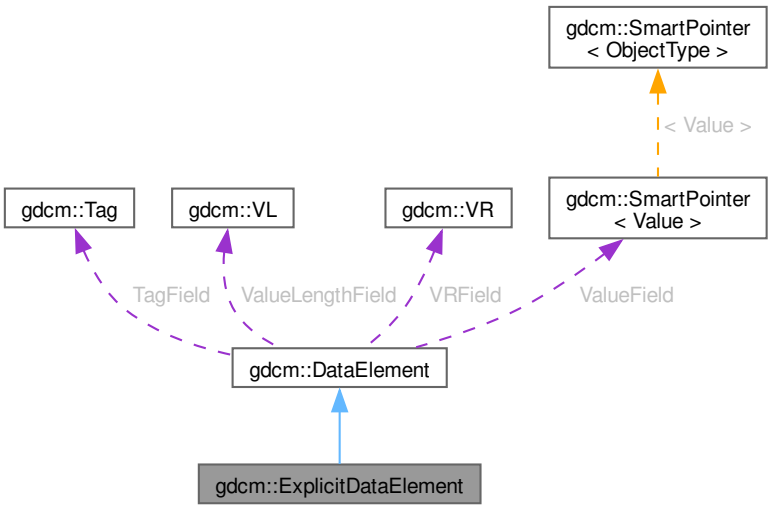
Class to read/write a DataElement as Explicit Data Element.

```
#include <gdcmExplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitDataElement:



Collaboration diagram for gdcm::ExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- template<typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.122.1 Detailed Description

Class to read/write a DataElement as Explicit Data Element.

Note

bla

Examples

DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, and ReadAndDumpDICOMDIR2.cxx.

12.122.2 Member Function Documentation

12.122.2.1 GetLength()

```
VL gdcm::ExplicitDataElement::GetLength () const
```

12.122.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::ExplicitDataElement::Read (
            std::istream & is)
```

12.122.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::ExplicitDataElement::ReadPreValue (
            std::istream & is)
```

12.122.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::ExplicitDataElement::ReadValue (
            std::istream & is,
            bool readvalues = true)
```

12.122.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::ExplicitDataElement::ReadWithLength (
            std::istream & is,
            VL & length)
```

12.122.2.6 Write()

```
template<typename TSwap>
const std::ostream & gdcm::ExplicitDataElement::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

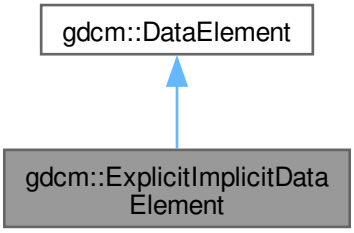
- gdcmExplicitDataElement.h

12.123 gdcm::ExplicitImplicitDataElement Class Reference

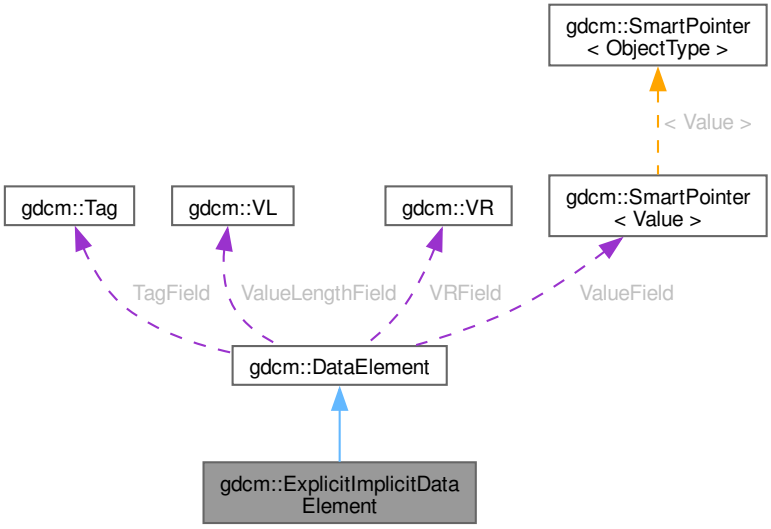
Class to read/write a DataElement as ExplicitImplicit Data Element.

```
#include <gdcmExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::ExplicitImplicitDataElement:



Collaboration diagram for gdcm::ExplicitImplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.123.1 Detailed Description

Class to read/write a DataElement as ExplicitImplicit Data Element.

Note

This only happen for some Philips images Should I derive from ExplicitDataElement instead ? This is the class that is the closest the GDCM1.x parser. At each element we try first to read it as explicit, if this fails, then we try again as an implicit element.

12.123.2 Member Function Documentation

12.123.2.1 GetLength()

```
VL gdcm::ExplicitImplicitDataElement::GetLength () const
```

12.123.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::ExplicitImplicitDataElement::Read (
            std::istream & is)
```

Referenced by ReadWithLength().

12.123.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::ExplicitImplicitDataElement::ReadPreValue (
            std::istream & is)
```

12.123.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::ExplicitImplicitDataElement::ReadValue (
            std::istream & is,
            bool readvalues = true)
```

12.123.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::ExplicitImplicitDataElement::ReadWithLength (
            std::istream & is,
            VL & length)  [inline]
```

References Read().

The documentation for this class was generated from the following file:

- gdcmExplicitImplicitDataElement.h

12.124 gdcm::Fiducials Class Reference

Fiducials.

```
#include <gdcmFiducials.h>
```

Public Member Functions

- Fiducials ()=default

12.124.1 Detailed Description

Fiducials.

12.124.2 Constructor & Destructor Documentation

12.124.2.1 Fiducials()

```
gdcm::Fiducials::Fiducials () [default]
```

The documentation for this class was generated from the following file:

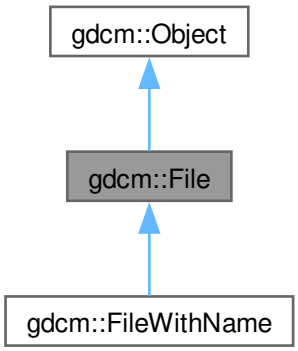
- gdcmFiducials.h

12.125 gdcm::File Class Reference

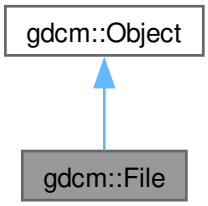
a DICOM File

```
#include <gdcmFile.h>
```

Inheritance diagram for gdcm::File:



Collaboration diagram for gdcm::File:



Public Member Functions

- File ()
- ~File () override
- DataSet & GetDataSet ()

  Get Data Set.
- const DataSet & GetDataSet () const

  Get Data Set.
- FileMetaInformation & GetHeader ()

  Get File Meta Information.
- const FileMetaInformation & GetHeader () const

  Get File Meta Information.
- std::istream & Read (std::istream &is)

  Read.
- void SetDataSet (const DataSet &ds)

  Set Data Set.
- void SetHeader (const FileMetaInformation &fmi)

  Set File Meta Information.
- std::ostream const & Write (std::ostream &os) const

  Write.

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Friends

- std::ostream & operator<< (std::ostream &os, const File &val)

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.125.1 Detailed Description

a DICOM File

See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.

See also

Reader Writer

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, DiffFile.cxx, DumpCSA.cs, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cx FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MpegVideoInfo.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, SimplePrint.cs, SimplePrintPatientName.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, and iU22tomultisc.cxx.

12.125.2 Constructor & Destructor Documentation

12.125.2.1 File()

```
gdcm::File::File ()
```

Referenced by gdcm::FileWithName::FileWithName(), ~File(), and operator<<.

12.125.2.2 ~File()

```
gdcm::File::~File () [override]
```

References File(), and operator<<.

12.125.3 Member Function Documentation

12.125.3.1 GetDataSet() [1/2]

```
DataSet & gdcm::File::GetDataSet () [inline]
```

Get Data Set.

12.125.3.2 GetDataSet() [2/2]

```
const DataSet & gdcm::File::GetDataSet () const [inline]
```

Get Data Set.

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DecompressImage.cs, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, SimplePrint.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.125.3.3 GetHeader() [1/2]

```
FileMetaInformation & gdcm::File::GetHeader () [inline]
```

Get File Meta Information.

12.125.3.4 GetHeader() [2/2]

```
const FileMetaInformation & gdcm::File::GetHeader () const [inline]
```

Get File Meta Information.

Examples

CreateJPIPDataSet.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetJPEGSamplePrecision.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReformatFile.cs, StandardizeFiles.cs, StreamImageReaderTest.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by operator<<.

12.125.3.5 Read()

```
std::istream & gdcm::File::Read (
            std::istream & is)
```

Read.

12.125.3.6 SetDataSet()

```
void gdcm::File::SetDataSet (
            const DataSet & ds)  [inline]
```

Set Data Set.

12.125.3.7 SetHeader()

```
void gdcm::File::SetHeader (
            const FileMetaInformation & fmi)  [inline]
```

Set File Meta Information.

12.125.3.8 Write()

```
std::ostream const & gdcm::File::Write (
            std::ostream & os) const
```

Write.

12.125.4 Friends And Related Symbol Documentation

12.125.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const File & val)  [friend]
```

References File(), GetHeader(), and operator<<.

Referenced by ~File(), and operator<<.

The documentation for this class was generated from the following file:

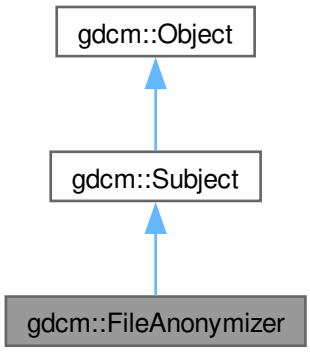
- gdcmFile.h

12.126 gdcm::FileAnonymizer Class Reference

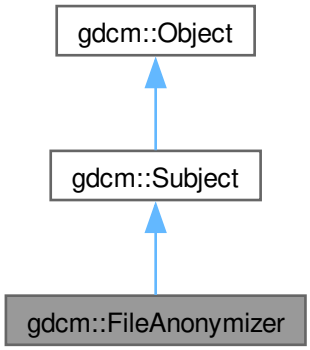
FileAnonymizer.

```
#include <gdcmFileAnonymizer.h>
```

Inheritance diagram for gdcm::FileAnonymizer:



Collaboration diagram for gdcm::FileAnonymizer:



Public Member Functions

- FileAnonymizer ()
- ~FileAnonymizer () override
- void Empty (Tag const &t)
- void Remove (Tag const &t)

  remove a tag (even a SQ can be removed)
- void Replace (Tag const &t, const char *value_data, VL const &vl)
- void Replace (Tag const &t, const char *value_str)
- void SetInputFileName (const char *filename_native)

  Set input filename.
- void SetOutputFileName (const char *filename_native)

  Set output filename.
- bool Write ()

  Write the output file.

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.126.1 Detailed Description

FileAnonymizer.

This Anonymizer is a file-based Anonymizer. It requires a valid DICOM file and will use the Value Length to skip over any information.

It will not load the DICOM dataset taken from SetInputFileName() into memory and should consume much less memory than Anonymizer.

Warning

: Each time you call Replace() with a value. This value will copied, and stored in memory. The behavior is not ideal for extremely large data (larger than memory size). This class is really meant to take a large DICOM input file and then only changed some small attribute.

caveats:

- This class will NOT work with unordered attributes in a DICOM File,
- This class does neither recompute nor update the Group Length element,
- This class currently does not update the File Meta Information header.
- Only strict inplace Replace operation is supported when input and output file are the same.

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

12.126.2 Constructor & Destructor Documentation

12.126.2.1 FileAnonymizer()

```
gdcm::FileAnonymizer::FileAnonymizer ()
```

12.126.2.2 ~FileAnonymizer()

```
gdcm::FileAnonymizer::~FileAnonymizer ()  [override]
```

12.126.3 Member Function Documentation

12.126.3.1 Empty()

```
void gdcm::FileAnonymizer::Empty (
            Tag const & t)
```

Make Tag t empty Warning: does not handle SQ element

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

12.126.3.2 Remove()

```
void gdcm::FileAnonymizer::Remove (
            Tag const & t)
```

remove a tag (even a SQ can be removed)

Examples

FileAnonymize.cs.

12.126.3.3 Replace() [1/2]

```
void gdcm::FileAnonymizer::Replace (
            Tag const & t,
            const char * value_data,
            VL const & vl)
```

when the value contains \0, it is a good idea to specify the length. This function is required when dealing with VRBINARY tag

12.126.3.4 Replace() [2/2]

```
void gdcm::FileAnonymizer::Replace (
            Tag const & t,
            const char * value_str)
```

Replace tag with another value, if tag is not found it will be created: WARNING: this function can only execute if tag is a VRASCII WARNING: Do not ever try to write a value in a SQ Data Element !

Examples

FileAnonymize.cs.

12.126.3.5 SetInputFileName()

```
void gdcm::FileAnonymizer::SetInputFileName (
            const char * filename_native)
```

Set input filename.

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

12.126.3.6 SetOutputFileName()

```
void gdcm::FileAnonymizer::SetOutputFileName (
            const char * filename_native)
```

Set output filename.

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

12.126.3.7 Write()

```
bool gdcm::FileAnonymizer::Write ()
```

Write the output file.

Examples

FileAnonymize.cs, and MakeTemplate.cxx.

The documentation for this class was generated from the following file:

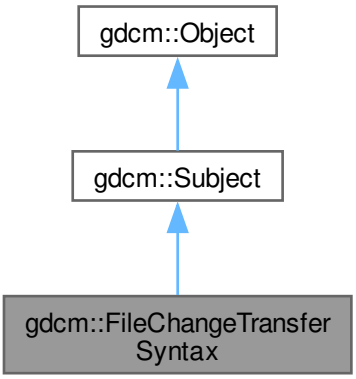
- gdcmFileAnonymizer.h

12.127 gdcm::FileChangeTransferSyntax Class Reference

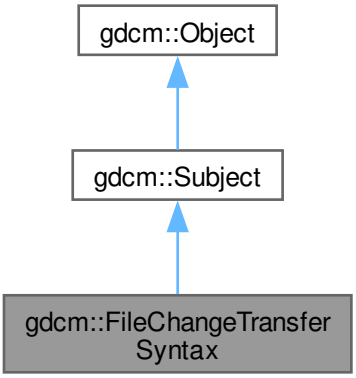
FileChangeTransferSyntax.

```
#include <gdcmFileChangeTransferSyntax.h>
```

Inheritance diagram for gdcm::FileChangeTransferSyntax:



Collaboration diagram for gdcm::FileChangeTransferSyntax:



Public Member Functions

- FileChangeTransferSyntax ()
- ~FileChangeTransferSyntax () override
- bool Change ()

  Change the transfer syntax.
- ImageCodec * GetCodec ()
- void SetInputFileName (const char *filename_native)

  Set input filename (raw DICOM).
- void SetOutputFileName (const char *filename_native)

  Set output filename (target compressed DICOM).
- void SetTransferSyntax (TransferSyntax const &ts)

  Specify the Target Transfer Syntax.

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static SmartPointer< FileChangeTransferSyntax > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.127.1 Detailed Description

FileChangeTransferSyntax.

This class is a file-based (limited) replacement of the in-memory ImageChangeTransferSyntax.

This class provide a file-based compression-only mechanism. It will take in an uncompressed DICOM image file (Pixel Data element). Then produced as output a compressed DICOM file (Transfer Syntax will be updated).

Currently it supports the following transfer syntax:

- JPEGLosslessProcess14_1

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

12.127.2 Constructor & Destructor Documentation

12.127.2.1 FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ()
```

Referenced by New().

12.127.2.2 ~FileChangeTransferSyntax()

```
gdcm::FileChangeTransferSyntax::~FileChangeTransferSyntax () [override]
```

12.127.3 Member Function Documentation

12.127.3.1 Change()

```
bool gdcm::FileChangeTransferSyntax::Change ()
```

Change the transfer syntax.

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

12.127.3.2 GetCodec()

```
ImageCodec * gdcm::FileChangeTransferSyntax::GetCodec ()
```

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

Examples

FileChangeTSLossy.cs.

12.127.3.3 New()

```
SmartPointer< FileChangeTransferSyntax > gdcm::FileChangeTransferSyntax::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

References FileChangeTransferSyntax().

12.127.3.4 SetInputFileName()

```
void gdcm::FileChangeTransferSyntax::SetInputFileName (
            const char * filename_native)
```

Set input filename (raw DICOM).

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

12.127.3.5 SetOutputFileName()

```
void gdcm::FileChangeTransferSyntax::SetOutputFileName (
            const char * filename_native)
```

Set output filename (target compressed DICOM).

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

12.127.3.6 SetTransferSyntax()

```
void gdcm::FileChangeTransferSyntax::SetTransferSyntax (
            TransferSyntax const & ts)
```

Specify the Target Transfer Syntax.

Examples

FileChangeTS.cs, and FileChangeTSLossy.cs.

The documentation for this class was generated from the following file:

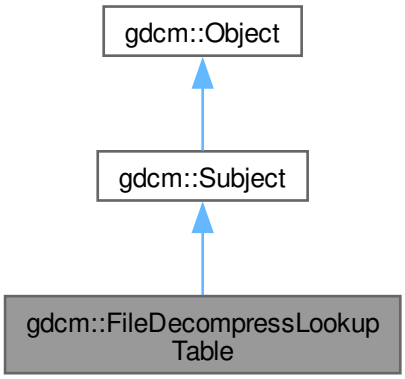
- gdcmFileChangeTransferSyntax.h

12.128 gdcm::FileDecompressLookupTable Class Reference

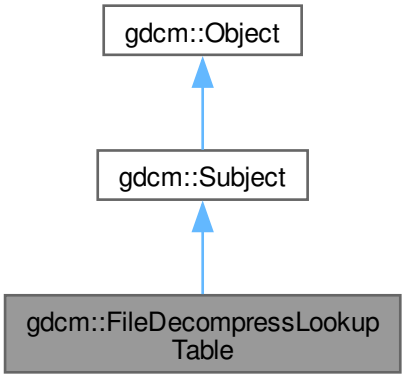
FileDecompressLookupTable class.

```
#include <gdcmFileDecompressLookupTable.h>
```

Inheritance diagram for gdcm::FileDecompressLookupTable:



Collaboration diagram for gdcm::FileDecompressLookupTable:



Public Member Functions

- FileDecompressLookupTable ()=default
- ~FileDecompressLookupTable () override=default
- bool Change ()

  Decompress.
- File & GetFile ()
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const
- void SetFile (const File &f)

  Set/Get File.
- void SetPixmap (Pixmap const &img)

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.128.1 Detailed Description

FileDecompressLookupTable class.

It decompress the segmented LUT into linearized one (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image

12.128.2 Constructor & Destructor Documentation

12.128.2.1 FileDecompressLookupTable()

```
gdcm::FileDecompressLookupTable::FileDecompressLookupTable () [default]
```

12.128.2.2 ~FileDecompressLookupTable()

```
gdcm::FileDecompressLookupTable::~FileDecompressLookupTable () [override], [default]
```

12.128.3 Member Function Documentation

12.128.3.1 Change()

```
bool gdcm::FileDecompressLookupTable::Change ()
```

Decompress.

12.128.3.2 GetFile()

```
File & gdcm::FileDecompressLookupTable::GetFile () [inline]
```

12.128.3.3 GetPixmap() [1/2]

```
Pixmap & gdcm::FileDecompressLookupTable::GetPixmap () [inline]
```

12.128.3.4 GetPixmap() [2/2]

```
const Pixmap & gdcm::FileDecompressLookupTable::GetPixmap () const [inline]
```

12.128.3.5 SetFile()

```
void gdcm::FileDecompressLookupTable::SetFile (
            const File & f) [inline]
```

Set/Get File.

12.128.3.6 SetPixmap()

```
void gdcm::FileDecompressLookupTable::SetPixmap (
            Pixmap const & img) [inline]
```

The documentation for this class was generated from the following file:

- gdcmFileDecompressLookupTable.h

12.129 gdcm::FileDerivation Class Reference

FileDerivation class.

```
#include <gdcmFileDerivation.h>
```

Public Member Functions

- FileDerivation ()
- ~FileDerivation ()
- bool AddReference (const char *referencedsopclassuid, const char *referencedsopinstanceuid)
- bool Derive ()

  Change.
- File & GetFile ()
- const File & GetFile () const
- void SetAppendDerivationHistory (bool b)
- void SetDerivationCodeSequenceCodeValue (unsigned int codevalue)

  Specify the Derivation Code Sequence Code Value. Eg 113040.
- void SetDerivationDescription (const char *dd)

  Specify the Derivation Description. Eg "lossy conversion".
- void SetFile (const File &f)

  Set/Get File.
- void SetPurposeOfReferenceCodeSequenceCodeValue (unsigned int codevalue)

  Specify the Purpose Of Reference Code Value. Eg. 121320.

Protected Member Functions

- bool AddDerivationDescription ()
- bool AddPurposeOfReferenceCodeSequence (DataSet &ds)
- bool AddSourceImageSequence ()

12.129.1 Detailed Description

FileDerivation class.

See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence

URL: `http://medical.nema.org/medical/dicom/2008/08_16pu.pdf`

DICOM Part 16 has two Context Groups CID 7202 and CID 7203 which contain a set of codes defining reason for a source image reference (ie. reason code for referenced image sequence) and a coded description of the deriation applied to the new image data from the original. Both these context groups are extensible.

File Derivation is compulsory when creating a lossy derived image.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

12.129.2 Constructor & Destructor Documentation

12.129.2.1 FileDerivation()

```
gdcm::FileDerivation::FileDerivation ()
```

12.129.2.2 ~FileDerivation()

```
gdcm::FileDerivation::~FileDerivation ()
```

12.129.3 Member Function Documentation

12.129.3.1 AddDerivationDescription()

```
bool gdcm::FileDerivation::AddDerivationDescription ()  [protected]
```

12.129.3.2 AddPurposeOfReferenceCodeSequence()

```
bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (
            DataSet & ds)  [protected]
```

12.129.3.3 AddReference()

```
bool gdcm::FileDerivation::AddReference (
            const char * referencedsopclassuid,
            const char * referencedsopinstanceuid)
```

Create the proper reference. Need to pass the original SOP Class UID and the original SOP Instance UID, so that those value can be used as Reference.

Warning

referencedsopclassuid and referencedsopinstanceuid needs to be \0 padded. This is not compatible with how ByteValue->GetPointer works.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

12.129.3.4 AddSourceImageSequence()

```
bool gdcm::FileDerivation::AddSourceImageSequence ()  [protected]
```

12.129.3.5 Derive()

```
bool gdcm::FileDerivation::Derive ()
```

Change.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

12.129.3.6 GetFile() [1/2]

```
File & gdcm::FileDerivation::GetFile () [inline]
```

Examples

GenFakeImage.cxx, and ReformatFile.cs.

12.129.3.7 GetFile() [2/2]

```
const File & gdcm::FileDerivation::GetFile () const [inline]
```

12.129.3.8 SetAppendDerivationHistory()

```
void gdcm::FileDerivation::SetAppendDerivationHistory (
            bool b)
```

Specify if Derivation history should be appended (default false) When false, this is an error if input already has a derivation history When true, both Purpose of Reference Code Value and Derivation Code Sequence Code Value can have their history appended.

12.129.3.9 SetDerivationCodeSequenceCodeValue()

```
void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (
            unsigned int codevalue)
```

Specify the Derivation Code Sequence Code Value. Eg 113040.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

12.129.3.10 SetDerivationDescription()

```
void gdcm::FileDerivation::SetDerivationDescription (
            const char * dd)
```

Specify the Derivation Description. Eg "lossy conversion".

12.129.3.11 SetFile()

```
void gdcm::FileDerivation::SetFile (
            const File & f)  [inline]
```

Set/Get File.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

12.129.3.12 SetPurposeOfReferenceCodeSequenceCodeValue()

```
void gdcm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (
            unsigned int codevalue)
```

Specify the Purpose Of Reference Code Value. Eg. 121320.

Examples

DeriveSeries.cxx, GenFakeImage.cxx, and ReformatFile.cs.

The documentation for this class was generated from the following file:

- gdcmFileDerivation.h

12.130 gdcm::FileExplicitFilter Class Reference

FileExplicitFilter class.

```
#include <gdcmFileExplicitFilter.h>
```

Public Member Functions

- FileExplicitFilter ()
- ~FileExplicitFilter ()=default
- bool Change ()

  Set FMI Transfer Syntax.
- File & GetFile ()
- void SetChangePrivateTags (bool b)

  Decide whether or not to VR'ify private tags.
- void SetFile (const File &f)

  Set/Get File.
- void SetRecomputeItemLength (bool b)

  By default set Sequence & Item length to Undefined to avoid recomputing length:
- void SetRecomputeSequenceLength (bool b)
- void SetUseVRUN (bool b)

  When VR=16bits in explicit but Implicit has a 32bits length, use VR=UN.

Protected Member Functions

- bool ChangeFMI ()
- bool ProcessDataSet (DataSet &ds, Dicts const &dicts)

12.130.1 Detailed Description

FileExplicitFilter class.

After changing a file from Implicit to Explicit representation (see ImageChangeTransferSyntax) one operation is to make sure the VR of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit representation, the VR is not stored directly in the file.

Warning

changing an implicit dataset to an explicit dataset is NOT a trivial task of simply changing the VR to the dict one:

- One has to make sure SQ is properly set
- One has to recompute the explicit length SQ
- One has to make sure that VR is valid for the encoding
- One has to make sure that VR 16bits can store the original value length

Examples

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

12.130.2 Constructor & Destructor Documentation

12.130.2.1 FileExplicitFilter()

```
gdcm::FileExplicitFilter::FileExplicitFilter () [inline]
```

12.130.2.2 ~FileExplicitFilter()

```
gdcm::FileExplicitFilter::~FileExplicitFilter () [default]
```

12.130.3 Member Function Documentation

12.130.3.1 Change()

```
bool gdcm::FileExplicitFilter::Change ()
```

Set FMI Transfer Syntax.

Change

Examples

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

12.130.3.2 ChangeFMI()

```
bool gdcm::FileExplicitFilter::ChangeFMI ()  [protected]
```

12.130.3.3 GetFile()

```
File & gdcm::FileExplicitFilter::GetFile ()  [inline]
```

12.130.3.4 ProcessDataSet()

```
bool gdcm::FileExplicitFilter::ProcessDataSet (
            DataSet & ds,
            Dicts const & dicts)  [protected]
```

12.130.3.5 SetChangePrivateTags()

```
void gdcm::FileExplicitFilter::SetChangePrivateTags (
            bool b)  [inline]
```

Decide whether or not to VR'ify private tags.

12.130.3.6 SetFile()

```
void gdcm::FileExplicitFilter::SetFile (
            const File & f)  [inline]
```

Set/Get File.

Examples

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

12.130.3.7 SetRecomputeItemLength()

```
void gdcm::FileExplicitFilter::SetRecomputeItemLength (
            bool b)
```

By default set Sequence & Item length to Undefined to avoid recomputing length:

12.130.3.8 SetRecomputeSequenceLength()

```
void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (
            bool b)
```

12.130.3.9 SetUseVRUN()

```
void gdcm::FileExplicitFilter::SetUseVRUN (
            bool b) [inline]
```

When VR=16bits in explicit but Implicit has a 32bits length, use VR=UN.

The documentation for this class was generated from the following file:

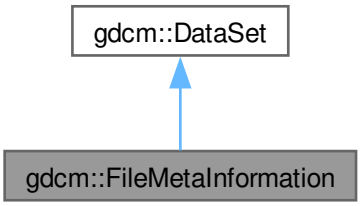
- gdcmFileExplicitFilter.h

12.131 gdcm::FileMetaInformation Class Reference

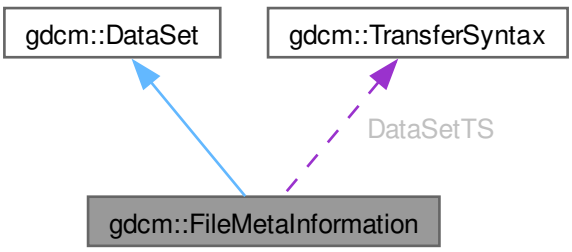
Class to represent a File Meta Information.

```
#include <gdcmFileMetaInformation.h>
```

Inheritance diagram for gdcm::FileMetaInformation:



Collaboration diagram for gdcm::FileMetaInformation:



Public Member Functions

- FileMetaInformation ()
- FileMetaInformation (FileMetaInformation const &fmi)=default
- ~FileMetaInformation ()
- void FillFromDataSet (DataSet const &ds)

  Construct a FileMetaInformation from an already existing DataSet:
- const TransferSyntax & GetDataSetTransferSyntax () const
- VL GetFullLength () const
- MediaStorage GetMediaStorage () const
- std::string GetMediaStorageAsString () const
- TransferSyntax::NegociatedType GetMetaInformationTS () const
- Preamble & GetPreamble ()
- const Preamble & GetPreamble () const

  Get Preamble.
- void Insert (const DataElement &de)
- bool IsValid () const
- FileMetaInformation & operator= (const FileMetaInformation &fmi)=default
- std::istream & Read (std::istream &is)

  Read.
- std::istream & ReadCompat (std::istream &is)
- void Replace (const DataElement &de)
- void SetDataSetTransferSyntax (const TransferSyntax &ts)
- void SetPreamble (const Preamble &p)
- std::ostream & Write (std::ostream &os) const

  Write.

Public Member Functions inherited from gdcm::DataSet

- Iterator Begin ()
- ConstIterator Begin () const
- void Clear ()
- template<typename TDE>
  unsigned int ComputeGroupLength (Tag const &tag) const
- Iterator End ()
- ConstIterator End () const
- bool FindDataElement (const PrivateTag &t) const

  Look up if private tag 't' is present in the dataset:
- bool FindDataElement (const Tag &t) const
- const DataElement & FindNextDataElement (const Tag &t) const
- const DataElement & GetDataElement (const PrivateTag &t) const

  Return the dataelement.
- const DataElement & GetDataElement (const Tag &t) const
- DataElementSet & GetDES ()
- const DataElementSet & GetDES () const
- template<typename TDE>
  VL GetLength () const
- MediaStorage GetMediaStorage () const
- std::string GetPrivateCreator (const Tag &t) const

- PrivateTag GetPrivateTag (const Tag &t) const

  Return the private tag of the private tag 't', private creator will be set to empty if not found.
- void Insert (const DataElement &de)
- bool IsEmpty () const

  Returns if the dataset is empty.
- const DataElement & operator() (uint16_t group, uint16_t element) const
- DataSet & operator= (DataSet const &)=default
- const DataElement & operator[ ] (const Tag &t) const
- void Print (std::ostream &os, std::string const &indent="") const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadNested (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedPrivateTags (std::istream &is, const std::set< PrivateTag > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedPrivateTagsWithLength (std::istream &is, const std::set< PrivateTag > &tags, VL &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedTags (std::istream &is, const std::set< Tag > &tags, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadSelectedTagsWithLength (std::istream &is, const std::set< Tag > &tags, VL &length, bool readvalues=true)
- template<typename TDE, typename TSwap>
  std::istream & ReadUpToTag (std::istream &is, const Tag &t, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadUpToTagWithLength (std::istream &is, const Tag &t, std::set< Tag > const &skiptags, VL &length)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- SizeType Remove (const Tag &tag)

  Completely remove a dataelement from the dataset.
- void Replace (const DataElement &de)

  Replace a dataelement with another one.
- void ReplaceEmpty (const DataElement &de)

  Only replace a DICOM attribute when it is missing or empty.
- SizeType Size () const
- template<typename TDE, typename TSwap>
  std::ostream const & Write (std::ostream &os) const

Static Public Member Functions

- static void AppendImplementationClassUID (const char *imp)
- static const char * GetImplementationClassUID ()
- static const char * GetImplementationVersionName ()
- static const char * GetSourceApplicationEntityTitle ()
- static void SetImplementationClassUID (const char *imp)

  Override the GDCM default values:
- static void SetImplementationVersionName (const char *version)
- static void SetSourceApplicationEntityTitle (const char *title)

Protected Member Functions

- void ComputeDataSetMediaStorageSOPClass ()
- void ComputeDataSetTransferSyntax ()
- void Default ()
- template<typename TSwap>
  std::istream & ReadCompatInternal (std::istream &is)

Protected Member Functions inherited from gdcm::DataSet

- Tag ComputeDataElement (const PrivateTag &t) const
- const DataElement & GetDEEnd () const
- void InsertDataElement (const DataElement &de)

Static Protected Member Functions

- static const char * GetFileMetaInformationVersion ()
- static const char * GetGDCMImplementationClassUID ()
- static const char * GetGDCMImplementationVersionName ()
- static const char * GetGDCMSourceApplicationEntityTitle ()

Protected Attributes

- MediaStorage::MSType DataSetMS
- TransferSyntax DataSetTS
- TransferSyntax::NegociatedType MetaInformationTS

Friends

- std::ostream & operator<< (std::ostream &_os, const FileMetaInformation &_val)

Additional Inherited Members

Public Types inherited from gdcm::DataSet

- typedef DataElementSet::const_iterator ConstIterator
- typedef std::set< DataElement > DataElementSet
- typedef DataElementSet::iterator Iterator
- typedef DataElementSet::size_type SizeType

12.131.1 Detailed Description

Class to represent a File Meta Information.

FileMetaInformation is a Explicit Structured Set. Whenever the file contains an ImplicitDataElement DataSet, a conversion will take place.

Definition: The File Meta Information includes identifying information on the encapsulated Data Set. This header consists of a 128 byte File Preamble, followed by a 4 byte DICOM prefix, followed by the File Meta Elements shown in Table 7.1-1. This header shall be present in every DICOM file.

See also

Writer Reader

Examples

DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ReadAndDumpDICOMDIR.cxx, ReformatFile.cs, and StandardizeFiles.cs.

12.131.2 Constructor & Destructor Documentation

12.131.2.1 FileMetaInformation() [1/2]

```
gdcm::FileMetaInformation::FileMetaInformation ()
```

Referenced by FileMetaInformation(), ~FileMetaInformation(), operator<<, and operator=().

12.131.2.2 ~FileMetaInformation()

```
gdcm::FileMetaInformation::~FileMetaInformation ()
```

References FileMetaInformation(), and operator<<.

12.131.2.3 FileMetaInformation() [2/2]

```
gdcm::FileMetaInformation::FileMetaInformation (
            FileMetaInformation const & fmi)  [default]
```

References FileMetaInformation().

12.131.3 Member Function Documentation

12.131.3.1 AppendImplementationClassUID()

```
void gdcm::FileMetaInformation::AppendImplementationClassUID (
            const char * imp)  [static]
```

12.131.3.2 ComputeDataSetMediaStorageSOPClass()

```
void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass () [protected]
```

12.131.3.3 ComputeDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax () [protected]
```

12.131.3.4 Default()

```
void gdcm::FileMetaInformation::Default () [protected]
```

12.131.3.5 FillFromDataSet()

```
void gdcm::FileMetaInformation::FillFromDataSet (
            DataSet const & ds)
```

Construct a FileMetaInformation from an already existing DataSet:

12.131.3.6 GetDataSetTransferSyntax()

```
const TransferSyntax & gdcm::FileMetaInformation::GetDataSetTransferSyntax () const [inline]
```

Examples

GetJPEGSamplePrecision.cxx, and MergeTwoFiles.cxx.

References DataSetTS.

12.131.3.7 GetFileMetaInformationVersion()

```
const char * gdcm::FileMetaInformation::GetFileMetaInformationVersion () [static], [protected]
```

12.131.3.8 GetFullLength()

```
VL gdcm::FileMetaInformation::GetFullLength () const [inline]
```

References gdcm::DataSet::GetLength(), and gdcm::VL::GetLength().

12.131.3.9 GetGDCMImplementationClassUID()

```
const char * gdcm::FileMetaInformation::GetGDCMImplementationClassUID () [static], [protected]
```

12.131.3.10 GetGDCMImplementationVersionName()

```
const char * gdcm::FileMetaInformation::GetGDCMImplementationVersionName () [static], [protected]
```

12.131.3.11 GetGDCMSourceApplicationEntityTitle()

```
const char * gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle () [static], [protected]
```

12.131.3.12 GetImplementationClassUID()

```
const char * gdcm::FileMetaInformation::GetImplementationClassUID () [static]
```

12.131.3.13 GetImplementationVersionName()

```
const char * gdcm::FileMetaInformation::GetImplementationVersionName () [static]
```

12.131.3.14 GetMediaStorage()

```
MediaStorage gdcm::FileMetaInformation::GetMediaStorage () const
```

12.131.3.15 GetMediaStorageAsString()

```
std::string gdcm::FileMetaInformation::GetMediaStorageAsString () const
```

12.131.3.16 GetMetaInformationTS()

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS () const [inline]
```

References MetaInformationTS.

12.131.3.17 GetPreamble() [1/2]

```
Preamble & gdcm::FileMetaInformation::GetPreamble () [inline]
```

12.131.3.18 GetPreamble() [2/2]

```
const Preamble & gdcm::FileMetaInformation::GetPreamble () const  [inline]
```

Get Preamble.

Referenced by operator<<.

12.131.3.19 GetSourceApplicationEntityTitle()

```
const char * gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ()  [static]
```

12.131.3.20 Insert()

```
void gdcm::FileMetaInformation::Insert (
            const DataElement & de)  [inline]
```

References gdcmErrorMacro, gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), and gdcm::DataSet::InsertDataElement().

Referenced by Replace().

12.131.3.21 IsValid()

```
bool gdcm::FileMetaInformation::IsValid () const  [inline]
```

12.131.3.22 operator=()

```
FileMetaInformation & gdcm::FileMetaInformation::operator= (
            const FileMetaInformation & fmi)  [default]
```

References FileMetaInformation().

12.131.3.23 Read()

```
std::istream & gdcm::FileMetaInformation::Read (
            std::istream & is)
```

Read.

12.131.3.24 ReadCompat()

```
std::istream & gdcm::FileMetaInformation::ReadCompat (
            std::istream & is)
```

12.131.3.25 ReadCompatInternal()

```
template<typename TSwap>
std::istream & gdcm::FileMetaInformation::ReadCompatInternal (
            std::istream & is)  [protected]
```

12.131.3.26 Replace()

```
void gdcm::FileMetaInformation::Replace (
            const DataElement & de)  [inline]
```

Examples

LargeVRDSExplicit.cxx.

References gdcm::DataElement::GetTag(), Insert(), and gdcm::DataSet::Remove().

12.131.3.27 SetDataSetTransferSyntax()

```
void gdcm::FileMetaInformation::SetDataSetTransferSyntax (
            const TransferSyntax & ts)
```

Examples

CreateJPIPDataSet.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MpegVideoInfo.cs, QIDO-RS.cxx, StreamImageReaderTest.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.131.3.28 SetImplementationClassUID()

```
void gdcm::FileMetaInformation::SetImplementationClassUID (
            const char * imp)  [static]
```

Override the GDCM default values:

12.131.3.29 SetImplementationVersionName()

```
void gdcm::FileMetaInformation::SetImplementationVersionName (
            const char * version)  [static]
```

12.131.3.30 SetPreamble()

```
void gdcm::FileMetaInformation::SetPreamble (
            const Preamble & p)  [inline]
```

12.131.3.31 SetSourceApplicationEntityTitle()

```
void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (
            const char * title)  [static]
```

Examples

FixJAIBugJPEGLS.cxx, GenerateDICOMDIR.cs, ReformatFile.cs, and StandardizeFiles.cs.

12.131.3.32 Write()

```
std::ostream & gdcm::FileMetaInformation::Write (
            std::ostream & os) const
```

Write.

12.131.4 Friends And Related Symbol Documentation

12.131.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const FileMetaInformation & _val)  [friend]
```

References FileMetaInformation(), GetPreamble(), operator<<, and gdcm::DataSet::Print().

Referenced by ~FileMetaInformation(), and operator<<.

12.131.5 Member Data Documentation

12.131.5.1 DataSetMS

```
MediaStorage::MSType gdcm::FileMetaInformation::DataSetMS  [protected]
```

12.131.5.2 DataSetTS

```
TransferSyntax gdcm::FileMetaInformation::DataSetTS  [protected]
```

Referenced by GetDataSetTransferSyntax().

12.131.5.3 MetaInformationTS

```
TransferSyntax::NegociatedType gdcm::FileMetaInformation::MetaInformationTS [protected]
```

Referenced by GetMetaInformationTS().

The documentation for this class was generated from the following file:

- gdcmFileMetaInformation.h

12.132 gdcm::Filename Class Reference

Class to manipulate file name's.

```
#include <gdcmFilename.h>
```

Public Member Functions

- Filename (const char *filename="")
- bool EndWith (const char ending[]) const

  Does the filename ends with a particular string ?
- const char * GetExtension ()

  return only the extension part of a filename
- const char * GetFileName () const

  Return the full filename.
- const char * GetName ()

  return only the name part of a filename
- const char * GetPath ()

  Return only the path component of a filename.
- bool IsEmpty () const

  return whether the filename is empty
- bool IsIdentical (Filename const &fn) const
- operator const char * () const
- const char * ToUnixSlashes ()

  Convert backslash (windows style) to UNIX style slash.
- const char * ToWindowsSlashes ()

  Convert forward slash (UNIX style) to windows style slash.

Static Public Member Functions

- static const char * Join (const char *path, const char *filename)

12.132.1 Detailed Description

Class to manipulate file name's.

Note

OS independent representation of a filename (to query path, name and extension from a filename)

12.132.2 Constructor & Destructor Documentation

12.132.2.1 Filename()

```
gdcm::Filename::Filename (
            const char * filename = "")  [inline]
```

Referenced by IsIdentical().

12.132.3 Member Function Documentation

12.132.3.1 EndWith()

```
bool gdcm::Filename::EndWith (
            const char ending[]) const
```

Does the filename ends with a particular string ?

12.132.3.2 GetExtension()

```
const char * gdcm::Filename::GetExtension ()
```

return only the extension part of a filename

12.132.3.3 GetFileName()

```
const char * gdcm::Filename::GetFileName () const  [inline]
```

Return the full filename.

Referenced by operator const char *().

12.132.3.4 GetName()

```
const char * gdcm::Filename::GetName ()
```

return only the name part of a filename

12.132.3.5 GetPath()

```
const char * gdcm::Filename::GetPath ()
```

Return only the path component of a filename.

Examples

ClinicalTrialIdentificationWorkflow.cs.

12.132.3.6 IsEmpty()

```
bool gdcm::Filename::IsEmpty () const  [inline]
```

return whether the filename is empty

12.132.3.7 IsIdentical()

```
bool gdcm::Filename::IsIdentical (
            Filename const & fn) const
```

References Filename().

12.132.3.8 Join()

```
const char * gdcm::Filename::Join (
            const char * path,
            const char * filename)  [static]
```

Join two paths NOT THREAD SAFE

12.132.3.9 operator const char *()

```
gdcm::Filename::operator const char * () const  [inline]
```

Simple operator to allow Filename myfilename("..."); const char * s = myfilename;

References GetFileName().

12.132.3.10 ToUnixSlashes()

```
const char * gdcm::Filename::ToUnixSlashes ()
```

Convert backslash (windows style) to UNIX style slash.

12.132.3.11 ToWindowsSlashes()

```
const char * gdcm::Filename::ToWindowsSlashes ()
```

Convert forward slash (UNIX style) to windows style slash.

The documentation for this class was generated from the following file:

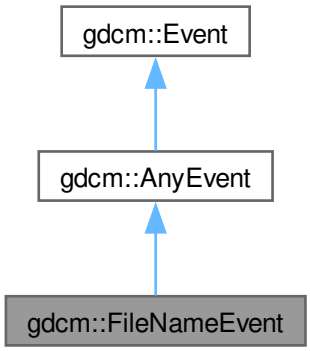
- gdcmFilename.h

12.133 gdcm::FileNameEvent Class Reference

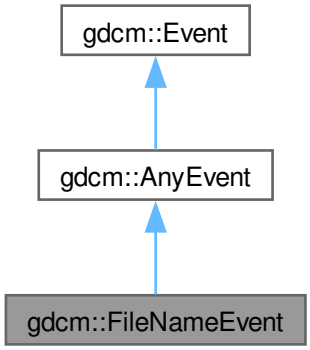
FileNameEvent.

```
#include <gdcmFileNameEvent.h>
```

Inheritance diagram for gdcm::FileNameEvent:



Collaboration diagram for gdcm::FileNameEvent:



Public Types

- typedef FileNameEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- FileNameEvent (const char *s="")
- FileNameEvent (const Self &s)
- ~FileNameEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- const char * GetEventName () const override
- const char * GetFileName () const
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete
- void SetFileName (const char *f)

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

12.133.1 Detailed Description

FileNameEvent.

Special type of event triggered during processing of FileSet

See also

AnyEvent

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

12.133.2 Member Typedef Documentation

12.133.2.1 Self

```
typedef FileNameEvent gdcm::FileNameEvent::Self
```

12.133.2.2 Superclass

```
typedef AnyEvent gdcm::FileNameEvent::Superclass
```

12.133.3 Constructor & Destructor Documentation

12.133.3.1 FileNameEvent() [1/2]

```
gdcm::FileNameEvent::FileNameEvent (
            const char * s = "")  [inline]
```

12.133.3.2 ~FileNameEvent()

```
gdcm::FileNameEvent::~FileNameEvent ()  [override], [default]
```

12.133.3.3 FileNameEvent() [2/2]

```
gdcm::FileNameEvent::FileNameEvent (
            const Self & s)  [inline]
```

12.133.4 Member Function Documentation

12.133.4.1 CheckEvent()

```
bool gdcm::FileNameEvent::CheckEvent (
            const ::gdcm::Event * e) const  [inline], [override]
```

12.133.4.2 GetEventName()

```
const char * gdcm::FileNameEvent::GetEventName () const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

12.133.4.3 GetFileName()

```
const char * gdcm::FileNameEvent::GetFileName () const  [inline]
```

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

12.133.4.4 MakeObject()

```
::gdcm::Event * gdcm::FileNameEvent::MakeObject () const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

12.133.4.5 operator=()

```
void gdcm::FileNameEvent::operator= (
            const Self & )  [delete]
```

12.133.4.6 SetFileName()

```
void gdcm::FileNameEvent::SetFileName (
            const char * f)  [inline]
```

The documentation for this class was generated from the following file:

- gdcmFileNameEvent.h

12.134 gdcm::FilenameGenerator Class Reference

FilenameGenerator.

```
#include <gdcmFilenameGenerator.h>
```

Public Types

- typedef std::vector< FilenameType > FilenamesType
- typedef std::string FilenameType
- typedef FilenamesType::size_type SizeType

Public Member Functions

- FilenameGenerator ()
- ~FilenameGenerator ()=default
- bool Generate ()

  Generate (return success).
- const char * GetFilename (SizeType n) const

  Get a particular filename (call after Generate).
- FilenamesType const & GetFilenames () const

  Return all filenames.
- SizeType GetNumberOfFilenames () const
- const char * GetPattern () const
- const char * GetPrefix () const
- void SetNumberOfFilenames (SizeType nfiles)

  Set/Get the number of filenames to generate.
- void SetPattern (const char *pattern)

  Set/Get pattern.
- void SetPrefix (const char *prefix)

  Set/Get prefix.

12.134.1 Detailed Description

FilenameGenerator.

class to generate filenames based on a pattern (C-style)

Output will be:

for i = 0, number of filenames: outfilename[i] = prefix + (pattern % i)

where pattern % i means C-style snprintf of Pattern using value 'i'

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

12.134.2 Member Typedef Documentation

12.134.2.1 FilenamesType

```
typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType
```

12.134.2.2 FilenameType

```
typedef std::string gdcm::FilenameGenerator::FilenameType
```

12.134.2.3 SizeType

```
typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType
```

12.134.3 Constructor & Destructor Documentation

12.134.3.1 FilenameGenerator()

```
gdcm::FilenameGenerator::FilenameGenerator () [inline]
```

12.134.3.2 ~FilenameGenerator()

```
gdcm::FilenameGenerator::~FilenameGenerator () [default]
```

12.134.4 Member Function Documentation

12.134.4.1 Generate()

```
bool gdcm::FilenameGenerator::Generate ()
```

Generate (return success).

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

12.134.4.2 GetFilename()

```
const char * gdcm::FilenameGenerator::GetFilename (
            SizeType n) const
```

Get a particular filename (call after Generate).

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

12.134.4.3 GetFilenames()

```
FilenamesType const & gdcm::FilenameGenerator::GetFilenames () const  [inline]
```

Return all filenames.

12.134.4.4 GetNumberOfFilenames()

```
SizeType gdcm::FilenameGenerator::GetNumberOfFilenames () const
```

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

12.134.4.5 GetPattern()

```
const char * gdcm::FilenameGenerator::GetPattern () const  [inline]
```

12.134.4.6 GetPrefix()

```
const char * gdcm::FilenameGenerator::GetPrefix () const  [inline]
```

12.134.4.7 SetNumberOfFilenames()

```
void gdcm::FilenameGenerator::SetNumberOfFilenames (
            SizeType nfiles)
```

Set/Get the number of filenames to generate.

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

12.134.4.8 SetPattern()

```
void gdcm::FilenameGenerator::SetPattern (
            const char * pattern)  [inline]
```

Set/Get pattern.

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

12.134.4.9 SetPrefix()

```
void gdcm::FilenameGenerator::SetPrefix (
            const char * prefix)  [inline]
```

Set/Get prefix.

The documentation for this class was generated from the following file:

- gdcmFilenameGenerator.h

12.135 gdcm::FileSet Class Reference

```
#include <gdcmFileSet.h>
```

Public Types

- typedef std::vector< FileType > FilesType
- typedef std::string FileType

Public Member Functions

- FileSet ()
- bool AddFile (const char *filename)
- void AddFile (File const &)
- FilesType const & GetFiles () const
- void SetFiles (FilesType const &files)

Friends

- std::ostream & operator<< (std::ostream &_os, const FileSet &d)

12.135.1 Detailed Description

File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which File IDs are unique.

12.135.2 Member Typedef Documentation

12.135.2.1 FilesType

```
typedef std::vector<FileType> gdcm::FileSet::FilesType
```

12.135.2.2 FileType

```
typedef std::string gdcm::FileSet::FileType
```

12.135.3 Constructor & Destructor Documentation

12.135.3.1 FileSet()

```
gdcm::FileSet::FileSet () [inline]
```

Referenced by operator<<.

12.135.4 Member Function Documentation

12.135.4.1 AddFile() [1/2]

```
bool gdcm::FileSet::AddFile (
            const char * filename)
```

Add a file 'filename' to the list of files. Return true on success, false in case filename could not be found on system.

12.135.4.2 AddFile() [2/2]

```
void gdcm::FileSet::AddFile (
            File const & ) [inline]
```

Deprecated . Does nothing

12.135.4.3 GetFiles()

```
FilesType const & gdcm::FileSet::GetFiles () const  [inline]
```

12.135.4.4 SetFiles()

```
void gdcm::FileSet::SetFiles (
            FilesType const & files)
```

12.135.5 Friends And Related Symbol Documentation

12.135.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const FileSet & d)  [friend]
```

References FileSet().

The documentation for this class was generated from the following file:

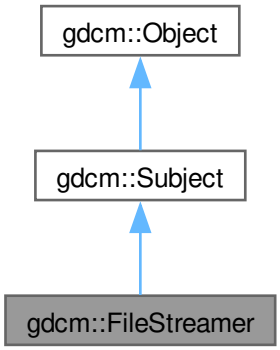
- gdcmFileSet.h

12.136 gdcm::FileStreamer Class Reference

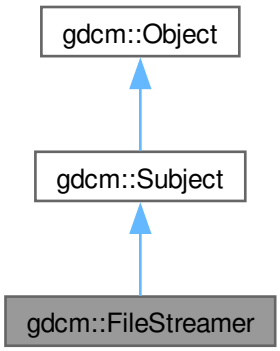
FileStreamer.

```
#include <gdcmFileStreamer.h>
```

Inheritance diagram for gdcm::FileStreamer:



Collaboration diagram for gdcm::FileStreamer:



Public Member Functions

- FileStreamer ()
- ~FileStreamer () override
- bool AppendToDataElement (const Tag &t, const char *array, size_t len)

  Append to previously started Tag t.
- bool AppendToGroupDataElement (const PrivateTag &pt, const char *array, size_t len)

  Append to previously started private creator.
- bool CheckDataElement (const Tag &t)
- void CheckTemplateFileName (bool check)
- bool ReserveDataElement (size_t len)
- bool ReserveGroupDataElement (unsigned short ndataelement)
- void SetOutputFileName (const char *filename_native)

  Set output filename (target file).
- void SetTemplateFileName (const char *filename_native)

  Set input DICOM template filename.
- bool StartDataElement (const Tag &t)
- bool StartGroupDataElement (const PrivateTag &pt, size_t maxsizede=0, uint8_t startoffset=0)
- bool StopDataElement (const Tag &t)

  Stop appending to tag t. This will compute the proper attribute length.
- bool StopGroupDataElement (const PrivateTag &pt)

  Stop appending to private creator.

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)

- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static SmartPointer< FileStreamer > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.136.1 Detailed Description

FileStreamer.

This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data.

This class support two mode of operation:

1. Creating a single DataElement by appending chunk after chunk of data.
2. Creating a set of DataElement within the same group, using a private creator for start. New DataElement are added any time the user defined maximum size for data element is reached.

Warning

any existing DataElement is removed, pick carefully which DataElement to add.

Examples

FileStreaming.cs.

12.136.2 Constructor & Destructor Documentation

12.136.2.1 FileStreamer()

```
gdcm::FileStreamer::FileStreamer ()
```

Referenced by New().

12.136.2.2 ~FileStreamer()

```
gdcm::FileStreamer::~FileStreamer () [override]
```

12.136.3 Member Function Documentation

12.136.3.1 AppendToDataElement()

```
bool gdcm::FileStreamer::AppendToDataElement (
            const Tag & t,
            const char * array,
            size_t len)
```

Append to previously started Tag t.

12.136.3.2 AppendToGroupDataElement()

```
bool gdcm::FileStreamer::AppendToGroupDataElement (
            const PrivateTag & pt,
            const char * array,
            size_t len)
```

Append to previously started private creator.

Examples

FileStreaming.cs.

12.136.3.3 CheckDataElement()

```
bool gdcm::FileStreamer::CheckDataElement (
            const Tag & t)
```

Decide to check the Data Element to be written (default: off) The implementation has default strategy for checking validity of DataElement. Currently it only support checking for the following tags:

- (7fe0,0010) Pixel Data

12.136.3.4 CheckTemplateFileName()

```
void gdcm::FileStreamer::CheckTemplateFileName (
            bool check)
```

Instead of simply blindly copying the input DICOM Template file, GDCM will be used to check the input file, and correct any issues recognized within the file. Only use if you do not have control over the input template file.

12.136.3.5 New()

```
SmartPointer< FileStreamer > gdcm::FileStreamer::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

References FileStreamer().

12.136.3.6 ReserveDataElement()

```
bool gdcm::FileStreamer::ReserveDataElement (
            size_t len)
```

Add a hint on the final size of the dataelement. When optimally chosen, this reduce the number of file in-place copying. Should be called before StartDataElement

12.136.3.7 ReserveGroupDataElement()

```
bool gdcm::FileStreamer::ReserveGroupDataElement (
            unsigned short ndataelement)
```

Optimisation: pre-allocate the number of dataelement within the private group (ndataelement <= 256). Should be called before StartGroupDataElement

12.136.3.8 SetOutputFileName()

```
void gdcm::FileStreamer::SetOutputFileName (
            const char * filename_native)
```

Set output filename (target file).

Examples

FileStreaming.cs.

12.136.3.9 SetTemplateFileName()

```
void gdcm::FileStreamer::SetTemplateFileName (
            const char * filename_native)
```

Set input DICOM template filename.

Examples

FileStreaming.cs.

12.136.3.10 StartDataElement()

```
bool gdcm::FileStreamer::StartDataElement (
            const Tag & t)
```

Start Single Data Element Operation This will delete any existing Tag t. Need to call it only once.

12.136.3.11 StartGroupDataElement()

```
bool gdcm::FileStreamer::StartGroupDataElement (
            const PrivateTag & pt,
            size_t maxsizede = 0,
            uint8_t startoffset = 0)
```

Start Private Group (multiple DataElement) Operation. Each newly added DataElement will have a length lower than

Parameters

maxsizede	. When not specified, maxsizede is set to maximum size allowed by DICOM (= 2^32). startoffset can be used to specify the very first element you want to start with (instead of the first possible). Value should be in [0x0, 0xff] This will find the first available private creator.

Bug maxsizede should be a value lower than the actual total size of the buffer to be copied

Examples

FileStreaming.cs.

12.136.3.12 StopDataElement()

```
bool gdcm::FileStreamer::StopDataElement (
            const Tag & t)
```

Stop appending to tag t. This will compute the proper attribute length.

12.136.3.13 StopGroupDataElement()

```
bool gdcm::FileStreamer::StopGroupDataElement (
            const PrivateTag & pt)
```

Stop appending to private creator.

Examples

FileStreaming.cs.

The documentation for this class was generated from the following file:

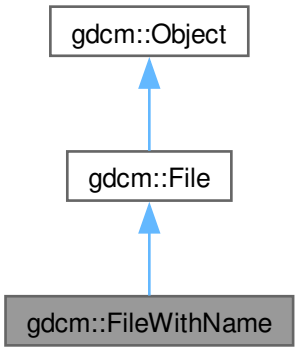
- gdcmFileStreamer.h

12.137 gdcm::FileWithName Class Reference

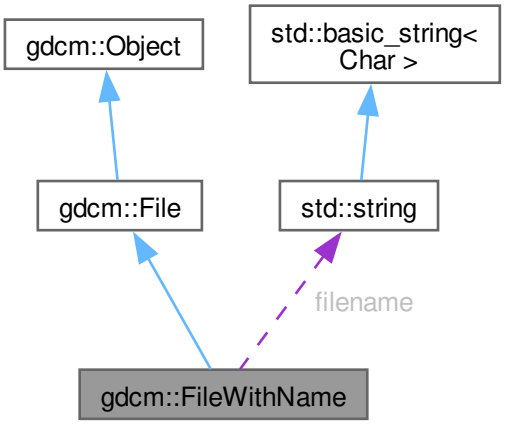
FileWithName.

```
#include <gdcmSerieHelper.h>
```

Inheritance diagram for gdcm::FileWithName:



Collaboration diagram for gdcm::FileWithName:



Public Member Functions

- FileWithName (File &f)

Public Member Functions inherited from gdcm::File

- File ()
- ~File () override
- DataSet & GetDataSet ()

  Get Data Set.
- const DataSet & GetDataSet () const

  Get Data Set.
- FileMetaInformation & GetHeader ()

  Get File Meta Information.
- const FileMetaInformation & GetHeader () const

  Get File Meta Information.
- std::istream & Read (std::istream &is)

  Read.
- void SetDataSet (const DataSet &ds)

  Set Data Set.
- void SetHeader (const FileMetaInformation &fmi)

  Set File Meta Information.
- std::ostream const & Write (std::ostream &os) const

  Write.

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Public Attributes

- std::string filename

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.137.1 Detailed Description

FileWithName.

Backward only class do not use in newer code

12.137.2 Constructor & Destructor Documentation

12.137.2.1 FileWithName()

```
gdcm::FileWithName::FileWithName (
            File & f)  [inline]
```

References gdcm::File::File(), and filename.

12.137.3 Member Data Documentation

12.137.3.1 filename

```
std::string gdcm::FileWithName::filename
```

Referenced by FileWithName().

The documentation for this class was generated from the following file:

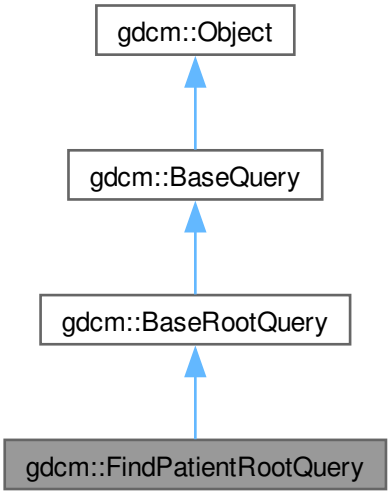
- gdcmSerieHelper.h

12.138 gdcm::FindPatientRootQuery Class Reference

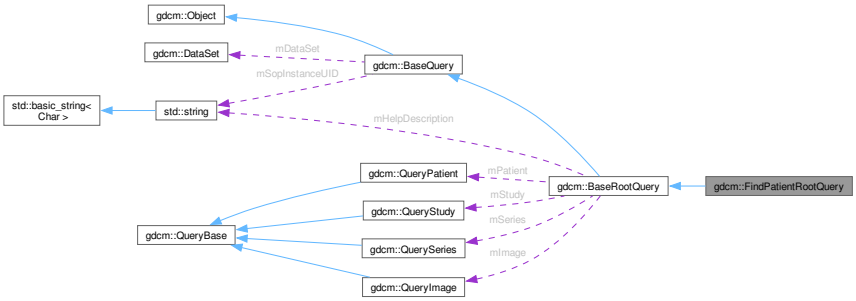
PatientRootQuery.

```
#include <gdcmFindPatientRootQuery.h>
```

Inheritance diagram for gdcm::FindPatientRootQuery:



Collaboration diagram for gdcm::FindPatientRootQuery:



Public Member Functions

- FindPatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Public Member Functions inherited from gdcm::BaseRootQuery

- ~BaseRootQuery () override=default
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)

Public Member Functions inherited from gdcm::BaseQuery

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Friends

- class QueryFactory

Additional Inherited Members

Static Public Member Functions inherited from gdcm::BaseRootQuery

- static QueryBase * Construct (ERootType inRootType, EQueryLevel qlevel)
- static int GetQueryLevelFromString (const char *str)
- static const char * GetQueryLevelString (EQueryLevel ql)

Protected Member Functions inherited from gdcm::BaseRootQuery

- BaseRootQuery ()

Protected Member Functions inherited from gdcm::BaseQuery

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes inherited from gdcm::BaseRootQuery

- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Protected Attributes inherited from gdcm::BaseQuery

- DataSet mDataSet
- std::string mSopInstanceUID

12.138.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

12.138.2 Constructor & Destructor Documentation

12.138.2.1 FindPatientRootQuery()

```
gdcm::FindPatientRootQuery::FindPatientRootQuery ()
```

12.138.3 Member Function Documentation

12.138.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID () const  [override], [virtual]
```

Implements gdcm::BaseQuery.

12.138.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::FindPatientRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

12.138.3.3 InitializeDataSet()

```
void gdcm::FindPatientRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

12.138.3.4 ValidateQuery()

```
bool gdcm::FindPatientRootQuery::ValidateQuery (
            bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

12.138.4 Friends And Related Symbol Documentation

12.138.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References QueryFactory.

Referenced by QueryFactory.

The documentation for this class was generated from the following file:

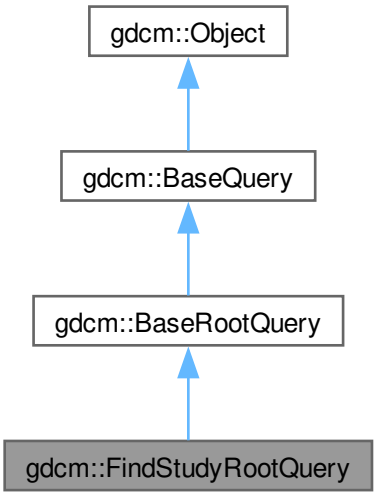
- gdcmFindPatientRootQuery.h

12.139 gdcm::FindStudyRootQuery Class Reference

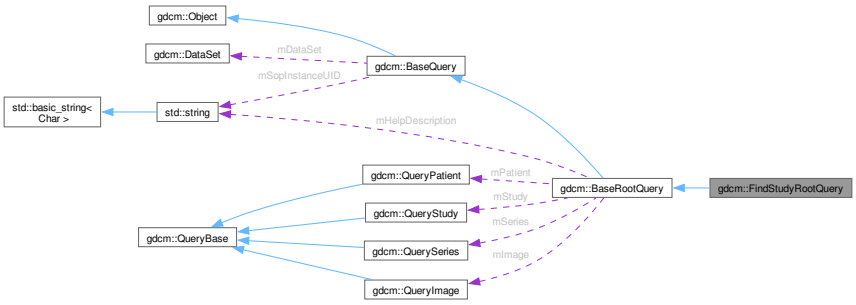
FindStudyRootQuery.

```
#include <gdcmFindStudyRootQuery.h>
```

Inheritance diagram for gdcm::FindStudyRootQuery:



Collaboration diagram for gdcm::FindStudyRootQuery:



Public Member Functions

- FindStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Public Member Functions inherited from gdcm::BaseRootQuery

- ~BaseRootQuery () override=default
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)

Public Member Functions inherited from gdcm::BaseQuery

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Friends

- class QueryFactory

Additional Inherited Members

Static Public Member Functions inherited from gdcm::BaseRootQuery

- static QueryBase * Construct (ERootType inRootType, EQueryLevel qlevel)
- static int GetQueryLevelFromString (const char *str)
- static const char * GetQueryLevelString (EQueryLevel ql)

Protected Member Functions inherited from gdcm::BaseRootQuery

- BaseRootQuery ()

Protected Member Functions inherited from gdcm::BaseQuery

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes inherited from gdcm::BaseRootQuery

- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Protected Attributes inherited from gdcm::BaseQuery

- DataSet mDataSet
- std::string mSopInstanceUID

12.139.1 Detailed Description

FindStudyRootQuery.

contains: the class which will produce a dataset for C-FIND with study root

12.139.2 Constructor & Destructor Documentation

12.139.2.1 FindStudyRootQuery()

```
gdcm::FindStudyRootQuery::FindStudyRootQuery ()
```

12.139.3 Member Function Documentation

12.139.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID () const  [override], [virtual]
```

Implements gdcm::BaseQuery.

12.139.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::FindStudyRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

12.139.3.3 InitializeDataSet()

```
void gdcm::FindStudyRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

12.139.3.4 ValidateQuery()

```
bool gdcm::FindStudyRootQuery::ValidateQuery (
            bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that (0008,0052) is set that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional)

Implements gdcm::BaseRootQuery.

12.139.4 Friends And Related Symbol Documentation

12.139.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References QueryFactory.

Referenced by QueryFactory.

The documentation for this class was generated from the following file:

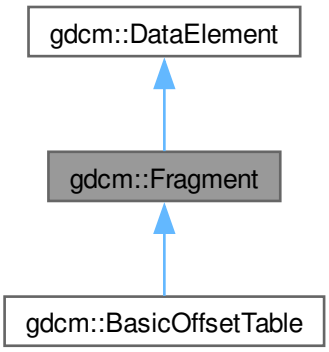
- gdcmFindStudyRootQuery.h

12.140 gdcm::Fragment Class Reference

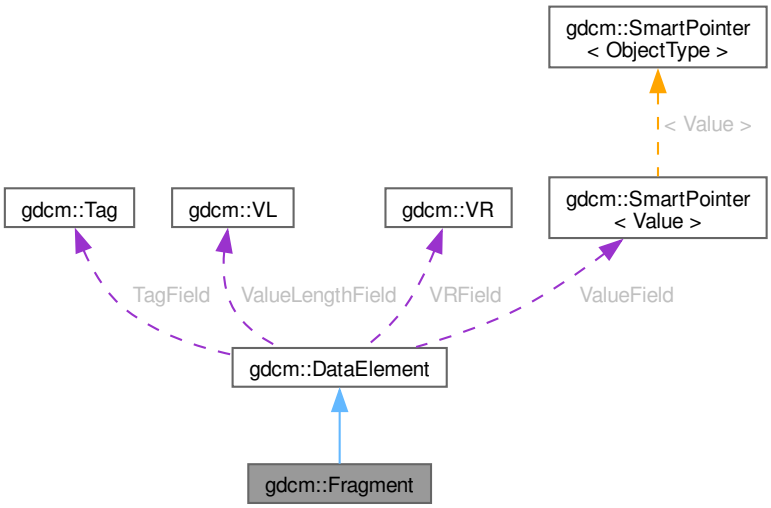
Class to represent a Fragment.

```
#include <gdcmFragment.h>
```

Inheritance diagram for gdcm::Fragment:



Collaboration diagram for gdcm::Fragment:



Public Member Functions

- Fragment ()
- VL ComputeLength () const
- VL GetLength () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadBacktrack (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is)
- template<typename TSwap>
  std::ostream & Write (std::ostream &os) const

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)

- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &os, const Fragment &val)

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.140.1 Detailed Description

Class to represent a Fragment.

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and MpegVideoInfo.cs.

12.140.2 Constructor & Destructor Documentation

12.140.2.1 Fragment()

```
gdcm::Fragment::Fragment () [inline]
```

References gdcm::DataElement::DataElement().

Referenced by gdcm::BasicOffsetTable::BasicOffsetTable(), and operator<<.

12.140.3 Member Function Documentation

12.140.3.1 ComputeLength()

```
VL gdcm::Fragment::ComputeLength () const
```

12.140.3.2 GetLength()

```
VL gdcm::Fragment::GetLength () const
```

12.140.3.3 Read()

```
template<typename TSwap>
std::istream & gdcm::Fragment::Read (
            std::istream & is) [inline]
```

References ReadPreValue(), and ReadValue().

Referenced by gdcm::SequenceOfFragments::ReadValue().

12.140.3.4 ReadBacktrack()

```
template<typename TSwap>
std::istream & gdcm::Fragment::ReadBacktrack (
            std::istream & is) [inline]
```

References gdcmErrorMacro, gdcmWarningMacro, gdcm::ParseException::SetLastElement(), gdcm::DataElement::TagField, gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLengthField.

Referenced by gdcm::SequenceOfFragments::ReadValue().

12.140.3.5 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::Fragment::ReadPreValue (
            std::istream & is)  [inline]
```

References gdcm::DataElement::TagField, and gdcm::DataElement::ValueLengthField.

Referenced by Read().

12.140.3.6 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::Fragment::ReadValue (
            std::istream & is)  [inline]
```

References gdcmWarningMacro, gdcm::ParseException::SetLastElement(), gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLengthField.

Referenced by Read().

12.140.3.7 Write()

```
template<typename TSwap>
std::ostream & gdcm::Fragment::Write (
            std::ostream & os) const  [inline]
```

References gdcm::ByteValue::ComputeLength(), gdcm::DataElement::GetByteValue(), gdcm::ByteValue::GetLength(), gdcm::DataElement::IsEmpty(), gdcm::DataElement::TagField, gdcm::DataElement::ValueLengthField, gdcm::ByteValue::Write(), and gdcm::VL::Write().

12.140.4 Friends And Related Symbol Documentation

12.140.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Fragment & val)  [friend]
```

References Fragment(), operator<<, gdcm::DataElement::TagField, gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLeng

Referenced by operator<<.

The documentation for this class was generated from the following file:

- gdcmFragment.h

12.141 gdcm::Global Class Reference

Global.

```
#include <gdcmGlobal.h>
```

Public Member Functions

- Global ()
- Global (const Global &_val)=delete
- ~Global ()
- bool Append (const char *path)
- Defs const & GetDefs () const
- Dicts & GetDicts ()
- Dicts const & GetDicts () const
- bool LoadResourcesFiles ()
- Global & operator= (const Global &_val)=delete
- bool Prepend (const char *path)

Static Public Member Functions

- static Global & GetInstance ()

  return the singleton instance

Protected Member Functions

- const char * Locate (const char *resfile) const

  Locate a resource file.

Friends

- std::ostream & operator<< (std::ostream &_os, const Global &g)

12.141.1 Detailed Description

Global.

Note

Global should be included in any translation unit that will use Dict or that implements the singleton pattern. It makes sure that the Dict singleton is created before and destroyed after all other singletons in GDCM.

Examples

BasicAnonymizer.cs, Cleaner.cs, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenerateStandardSOPClasses.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

12.141.2 Constructor & Destructor Documentation

12.141.2.1 Global() [1/2]

```
gdcm::Global::Global ()
```

Referenced by Global(), GetInstance(), operator<<, and operator=().

12.141.2.2 ~Global()

```
gdcm::Global::~Global ()
```

12.141.2.3 Global() [2/2]

```
gdcm::Global::Global (
            const Global & _val)  [delete]
```

References Global().

12.141.3 Member Function Documentation

12.141.3.1 Append()

```
bool gdcm::Global::Append (
            const char * path)
```

Append path at the end of the path list

Warning

not thread safe !

12.141.3.2 GetDefs()

```
Defs const & gdcm::Global::GetDefs () const
```

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

Examples

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

12.141.3.3 GetDicts() [1/2]

```
Dicts & gdcm::Global::GetDicts ()
```

12.141.3.4 GetDicts() [2/2]

```
Dicts const & gdcm::Global::GetDicts () const
```

retrieve the default/internal dicts (Part 6) This dict is filled up at load time

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, MrProtocol.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

12.141.3.5 GetInstance()

```
Global & gdcm::Global::GetInstance () [static]
```

return the singleton instance

Examples

BasicAnonymizer.cs, Cleaner.cs, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenerateStandardSOPClasses.cxx, MrProtocol.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

References Global().

12.141.3.6 LoadResourcesFiles()

```
bool gdcm::Global::LoadResourcesFiles ()
```

Load all internal XML files, resource path need to have been set before calling this member function (see Append/↩ Prepend members func)

Warning

not thread safe !

Examples

BasicAnonymizer.cs, Cleaner.cs, ClinicalTrialIdentificationWorkflow.cs, GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

12.141.3.7 Locate()

```
const char * gdcm::Global::Locate (
            const char * resfile) const  [protected]
```

Locate a resource file.

12.141.3.8 operator=()

```
Global & gdcm::Global::operator= (
            const Global & _val)  [delete]
```

References Global().

12.141.3.9 Prepend()

```
bool gdcm::Global::Prepend (
            const char * path)
```

Prepend path at the beginning of the path list

Warning

not thread safe !

12.141.4 Friends And Related Symbol Documentation

12.141.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Global & g)  [friend]
```

References Global().

The documentation for this class was generated from the following file:

- gdcmGlobal.h

12.142 gdcm::GroupDict Class Reference

Class to represent the mapping from group number to its abbreviation and name.

```
#include <gdcmGroupDict.h>
```

Public Types

- typedef std::vector< std::string > GroupStringVector

Public Member Functions

- GroupDict ()
- ~GroupDict ()=default
- std::string const & GetAbbreviation (uint16_t num) const
- std::string const & GetName (uint16_t num) const
- size_t Size () const

Protected Member Functions

- void Add (std::string const &abbreviation, std::string const &name)
- void Insert (uint16_t num, std::string const &abbreviation, std::string const &name)

Friends

- std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)

12.142.1 Detailed Description

Class to represent the mapping from group number to its abbreviation and name.

Note

Should I rewrite this class to use a std::map instead of std::vector for problem of memory consumption ?

12.142.2 Member Typedef Documentation

12.142.2.1 GroupStringVector

```
typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector
```

12.142.3 Constructor & Destructor Documentation

12.142.3.1 GroupDict()

```
gdcm::GroupDict::GroupDict () [inline]
```

Referenced by ~GroupDict(), Insert(), and operator<<.

12.142.3.2 ~GroupDict()

```
gdcm::GroupDict::~GroupDict () [default]
```

References GroupDict(), and operator<<.

12.142.4 Member Function Documentation

12.142.4.1 Add()

```
void gdcm::GroupDict::Add (
            std::string const & abbreviation,
            std::string const & name) [protected]
```

12.142.4.2 GetAbbreviation()

```
std::string const & gdcm::GroupDict::GetAbbreviation (
            uint16_t num) const
```

Referenced by operator<<.

12.142.4.3 GetName()

```
std::string const & gdcm::GroupDict::GetName (
            uint16_t num) const
```

Referenced by operator<<.

12.142.4.4 Insert()

```
void gdcm::GroupDict::Insert (
            uint16_t num,
            std::string const & abbreviation,
            std::string const & name) [protected]
```

References GroupDict().

12.142.4.5 Size()

```
size_t gdcm::GroupDict::Size () const [inline]
```

Referenced by operator<<.

12.142.5 Friends And Related Symbol Documentation

12.142.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const GroupDict & _val)  [friend]
```

References GroupDict(), GetAbbreviation(), GetName(), and Size().

Referenced by ~GroupDict().

The documentation for this class was generated from the following file:

- gdcmGroupDict.h

12.143 gdcm::IconImageFilter Class Reference

IconImageFilter.

```
#include <gdcmIconImageFilter.h>
```

Public Member Functions

- IconImageFilter ()
- ~IconImageFilter ()
- bool Extract ()

  Extract all Icon found in File.
- File & GetFile ()
- const File & GetFile () const
- IconImage & GetIconImage (unsigned int i) const
- unsigned int GetNumberOfIconImages () const

  Retrieve extract IconImage (need to call Extract first).
- void SetFile (const File &f)

  Set/Get File.

Protected Member Functions

- void ExtractIconImages ()
- void ExtractVeproIconImages ()

12.143.1 Detailed Description

IconImageFilter.

This filter will extract icons from a File This filter will loop over all known sequence (public and private) that may contains an IconImage and retrieve them. The filter will fails with a value of false if no icon can be found Since it handle both public and private icon type, one should not assume the icon is in uncompress form, some private vendor store private icon in JPEG8/JPEG12

Implementation details: This filter supports the following Icons:

- (0088,0200) Icon Image Sequence
- (0009,10,GEIIS) GE IIS Thumbnail Sequence
- (6003,10,GEMS_Ultrasound_ImageGroup_001) GEMS Image Thumbnail Sequence
- (0055,30,VEPRO VIF 3.0 DATA) Icon Data
- (0055,30,VEPRO VIM 5.0 DATA) ICONDATA2

Warning

the icon stored in those private attribute do not conform to definition of Icon Image Sequence (do not simply copy/paste). For example some private icon can be expressed as 12bits pixel, while the DICOM standard only allow 8bits icons.

See also

ImageReader

Examples

ExtractIconFromFile.cxx.

12.143.2 Constructor & Destructor Documentation

12.143.2.1 IconImageFilter()

```
gdcm::IconImageFilter::IconImageFilter ()
```

12.143.2.2 ~IconImageFilter()

```
gdcm::IconImageFilter::~IconImageFilter ()
```

12.143.3 Member Function Documentation

12.143.3.1 Extract()

```
bool gdcm::IconImageFilter::Extract ()
```

Extract all Icon found in File.

Examples

ExtractIconFromFile.cxx.

12.143.3.2 ExtractIconImages()

```
void gdcm::IconImageFilter::ExtractIconImages ()  [protected]
```

12.143.3.3 ExtractVeproIconImages()

```
void gdcm::IconImageFilter::ExtractVeproIconImages ()  [protected]
```

12.143.3.4 GetFile() [1/2]

```
File & gdcm::IconImageFilter::GetFile ()  [inline]
```

12.143.3.5 GetFile() [2/2]

```
const File & gdcm::IconImageFilter::GetFile () const  [inline]
```

12.143.3.6 GetIconImage()

```
IconImage & gdcm::IconImageFilter::GetIconImage (
            unsigned int i) const
```

Examples

ExtractIconFromFile.cxx.

12.143.3.7 GetNumberOfIconImages()

```
unsigned int gdcm::IconImageFilter::GetNumberOfIconImages () const
```

Retrieve extract IconImage (need to call Extract first).

Examples

ExtractIconFromFile.cxx.

12.143.3.8 SetFile()

```
void gdcm::IconImageFilter::SetFile (
            const File & f)  [inline]
```

Set/Get File.

Examples

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmIconImageFilter.h

12.144 gdcm::IconImageGenerator Class Reference

IconImageGenerator.

```
#include <gdcmIconImageGenerator.h>
```

Public Member Functions

- IconImageGenerator ()
- ~IconImageGenerator ()
- void AutoPixelMinMax (bool b)
- void ConvertRGBToPaletteColor (bool b)
- bool Generate ()

  Generate Icon.
- const IconImage & GetIconImage () const

  Retrieve generated Icon.
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const
- void SetOutputDimensions (const unsigned int dims[2])

  Set Target dimension of output Icon.
- void SetOutsideValuePixel (double v)
- void SetPixelMinMax (double min, double max)
- void SetPixmap (const Pixmap &p)

  Set/Get File.

12.144.1 Detailed Description

IconImageGenerator.

This filter will generate a valid Icon from the Pixel Data element (an instance of Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

- MONOCHROME1
- MONOCHROME2
- PALETTE_COLOR

The Pixel Bits Allocated is restricted to 8bits, therefore 16 bits image needs to be rescaled. By default the filter will use the full scalar range of 16bits image to rescale to unsigned 8bits. This may not be ideal for some situation, in which case the API SetPixelMinMax can be used to overwrite the default min,max interval used.

See also

ImageReader

Examples

ExtractIconFromFile.cxx.

12.144.2 Constructor & Destructor Documentation

12.144.2.1 IconImageGenerator()

```
gdcm::IconImageGenerator::IconImageGenerator ()
```

12.144.2.2 ~IconImageGenerator()

```
gdcm::IconImageGenerator::~IconImageGenerator ()
```

12.144.3 Member Function Documentation

12.144.3.1 AutoPixelMinMax()

```
void gdcm::IconImageGenerator::AutoPixelMinMax (
            bool b)
```

Instead of explicitly specifying the min/max value for the rescale operation, let the internal mechanism compute the min/max of icon and rescale to best appropriate.

Examples

ExtractIconFromFile.cxx.

12.144.3.2 ConvertRGBToPaletteColor()

```
void gdcm::IconImageGenerator::ConvertRGBToPaletteColor (
            bool b)
```

Converting from RGB to PALETTE_COLOR can be a slow operation. However DICOM standard requires that color icon be described as palette. Set this boolean to false only if you understand the consequences. default value is true, false generates invalid Icon Image Sequence

12.144.3.3 Generate()

```
bool gdcm::IconImageGenerator::Generate ()
```

Generate Icon.

Examples

ExtractIconFromFile.cxx.

12.144.3.4 GetIconImage()

```
const IconImage & gdcm::IconImageGenerator::GetIconImage () const  [inline]
```

Retrieve generated Icon.

Examples

ExtractIconFromFile.cxx.

12.144.3.5 GetPixmap() [1/2]

```
Pixmap & gdcm::IconImageGenerator::GetPixmap ()  [inline]
```

12.144.3.6 GetPixmap() [2/2]

```
const Pixmap & gdcm::IconImageGenerator::GetPixmap () const  [inline]
```

12.144.3.7 SetOutputDimensions()

```
void gdcm::IconImageGenerator::SetOutputDimensions (
            const unsigned int dims[2])
```

Set Target dimension of output Icon.

Examples

ExtractIconFromFile.cxx.

12.144.3.8 SetOutsideValuePixel()

```
void gdcm::IconImageGenerator::SetOutsideValuePixel (
            double v)
```

Set a pixel value that should be discarded. This happen typically for CT image, where a pixel has been used to pad outside the image (see Pixel Padding Value). Requires AutoPixelMinMax(true)

12.144.3.9 SetPixelMinMax()

```
void gdcm::IconImageGenerator::SetPixelMinMax (
            double min,
            double max)
```

Override default min/max to compute best rescale for 16bits -> 8bits downscale. Typically those value can be read from the SmallestImagePixelValue LargestImagePixelValue DICOM attribute.

12.144.3.10 SetPixmap()

```
void gdcm::IconImageGenerator::SetPixmap (
            const Pixmap & p)  [inline]
```

Set/Get File.

Examples

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmIconImageGenerator.h

12.145 gdcm::ignore_char Struct Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- ignore_char (char c)

Public Attributes

- char m_char

12.145.1 Constructor & Destructor Documentation

12.145.1.1 ignore_char()

```
gdcm::ignore_char::ignore_char (
            char c) [inline]
```

References m_char.

12.145.2 Member Data Documentation

12.145.2.1 m_char

```
char gdcm::ignore_char::m_char
```

Referenced by ignore_char(), and gdcm::operator>>().

The documentation for this struct was generated from the following file:

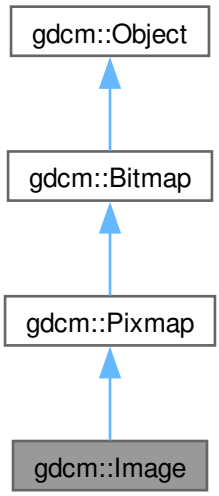
- gdcmElement.h

12.146 gdcm::Image Class Reference

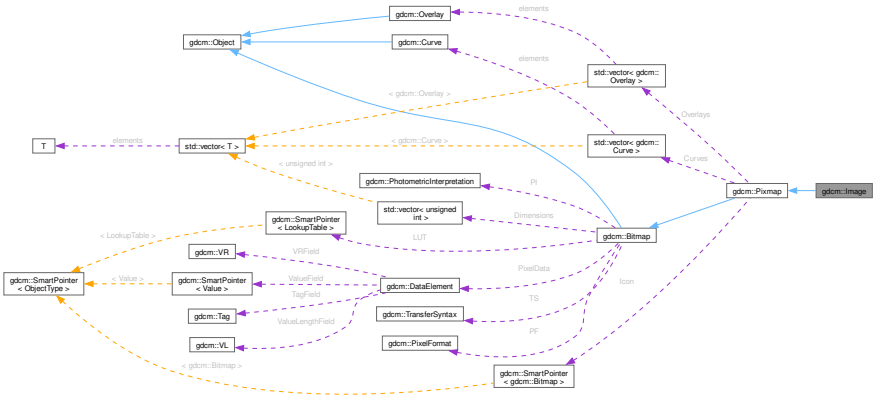
Image.

```
#include <gdcmImage.h>
```

Inheritance diagram for gdcm::Image:



Collaboration diagram for gdcm::Image:



Public Member Functions

- Image ()
- ~Image () override=default
- const double * GetDirectionCosines () const
- double GetDirectionCosines (unsigned int idx) const
- double GetIntercept () const
- const double * GetOrigin () const
- double GetOrigin (unsigned int idx) const
- double GetSlope () const
- const double * GetSpacing () const
- double GetSpacing (unsigned int idx) const
- void Print (std::ostream &os) const override

  print
- void SetDirectionCosines (const double dircos[6])
- void SetDirectionCosines (const float dircos[6])
- void SetDirectionCosines (unsigned int idx, double dircos)
- void SetIntercept (double intercept)

  intercept
- void SetOrigin (const double origin[3])
- void SetOrigin (const float origin[3])
- void SetOrigin (unsigned int idx, double ori)
- void SetSlope (double slope)

  slope
- void SetSpacing (const double spacing[3])
- void SetSpacing (unsigned int idx, double spacing)

Public Member Functions inherited from gdcm::Pixmap

- Pixmap ()
- ~Pixmap () override
- bool AreOverlaysInPixelData () const override

  returns if Overlays are stored in the unused bit of the pixel data:
- Curve & GetCurve (size_t i=0)

  Curve: group 50xx.
- const Curve & GetCurve (size_t i=0) const
- IconImage & GetIconImage ()
- const IconImage & GetIconImage () const

  Set/Get Icon Image.
- size_t GetNumberOfCurves () const
- size_t GetNumberOfOverlays () const
- Overlay & GetOverlay (size_t i=0)

  Overlay: group 60xx.
- const Overlay & GetOverlay (size_t i=0) const
- void Print (std::ostream &) const override
- void RemoveOverlay (size_t i)
- void SetIconImage (IconImage const &ii)
- void SetNumberOfCurves (size_t n)
- void SetNumberOfOverlays (size_t n)
- bool UnusedBitsPresentInPixelData () const override

  returns if there are unused bits in the pixel data

Public Member Functions inherited from gdcm::Bitmap

- Bitmap ()
- ~Bitmap () override
- void Clear ()
- bool GetBuffer (char *buffer) const

  Access the raw data.
- unsigned long GetBufferLength () const
- unsigned int GetColumns () const
- DataElement & GetDataElement ()
- const DataElement & GetDataElement () const
- unsigned int GetDimension (unsigned int idx) const
- const unsigned int * GetDimensions () const

  Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...
- LookupTable & GetLUT ()
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const

  INTERNAL do not use.
- unsigned int GetNumberOfDimensions () const

  Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.
- const PhotometricInterpretation & GetPhotometricInterpretation () const

  return the photometric interpretation
- PixelFormat & GetPixelFormat ()

- const PixelFormat & GetPixelFormat () const

  Get/Set PixelFormat.
- unsigned int GetPlanarConfiguration () const

  return the planar configuration
- unsigned int GetRows () const
- const TransferSyntax & GetTransferSyntax () const
- bool IsEmpty () const
- bool IsLossy () const

  Return whether or not the image was compressed using a lossy compressor or not.
- bool IsTransferSyntaxCompatible (TransferSyntax const &ts) const
- void SetColumns (unsigned int col)
- void SetDataElement (DataElement const &de)
- void SetDimension (unsigned int idx, unsigned int dim)
- void SetDimensions (const unsigned int dims[3])
- void SetLossyFlag (bool f)

  Specifically set that the image was compressed using a lossy compression mechanism.
- void SetLUT (LookupTable const &lut)

  Set/Get LUT.
- void SetNeedByteSwap (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)
- void SetRows (unsigned int rows)
- void SetTransferSyntax (TransferSyntax const &ts)

  Transfer syntax.

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Additional Inherited Members

Protected Types inherited from gdcm::Bitmap

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions inherited from gdcm::Bitmap

- bool ComputeLossyFlag ()
- bool GetBuffer2 (std::ostream &os) const
- bool TryJPEG2000Codec (char ∗buffer, bool &lossyflag) const
- bool TryJPEG2000Codec2 (std::ostream &os) const
- bool TryJPEGCodec (char ∗buffer, bool &lossyflag) const
- bool TryJPEGCodec2 (std::ostream &os) const
- bool TryJPEGLSCodec (char ∗buffer, bool &lossyflag) const
- bool TryKAKADUCodec (char ∗buffer, bool &lossyflag) const
- bool TryPVRGCodec (char ∗buffer, bool &lossyflag) const
- bool TryRAWCodec (char ∗buffer, bool &lossyflag) const
- bool TryRLECodec (char ∗buffer, bool &lossyflag) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes inherited from gdcm::Pixmap

- std::vector< Curve > Curves
- SmartPointer< IconImage > Icon
- std::vector< Overlay > Overlays

Protected Attributes inherited from gdcm::Bitmap

- std::vector< unsigned int > Dimensions
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- DataElement PixelData
- unsigned int PlanarConfiguration
- TransferSyntax TS

12.146.1 Detailed Description

Image.

This is the container for an Image in the general sense. From this container you should be able to request information like:

- Origin
- Dimension
- PixelFormat ... But also to retrieve the image as a raw buffer (char ∗) Since we have to deal with both RAW data and JPEG stream (which internally encode all the above information) this API might seems redundant. One way to solve that would be to subclass Image with JPEGImage which would from the stream extract the header info and fill it to please Image...well except origin for instance

Basically you can see it as a storage for the Pixel Data element (7fe0,0010).

Warning

This class does some heuristics to guess the Spacing but is not compatible with DICOM CP-586. In case of doubt use PixmapReader instead

See also

ImageReader PixmapReader

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, MpegVideoInfo.cs, PatchFile.cxx, PrintLUT.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, TemplateEmptyImage.cxx, csa2img.cxx, iU22tomultisc.cxx, and threadgdcm.cxx.

12.146.2 Constructor & Destructor Documentation

12.146.2.1 Image()

```
gdcm::Image::Image () [inline]
```

12.146.2.2 ∼Image()

```
gdcm::Image::~Image () [override], [default]
```

12.146.3 Member Function Documentation

12.146.3.1 GetDirectionCosines() [1/2]

```
const double * gdcm::Image::GetDirectionCosines () const
```

Return a 6-tuples specifying the direction cosines A default value of (1,0,0,0,1,0) will be return when the direction cosines was not specified.

12.146.3.2 GetDirectionCosines() [2/2]

```
double gdcm::Image::GetDirectionCosines (
            unsigned int idx) const
```

12.146.3.3 GetIntercept()

```
double gdcm::Image::GetIntercept () const  [inline]
```

12.146.3.4 GetOrigin() [1/2]

```
const double * gdcm::Image::GetOrigin () const
```

Return a 3-tuples specifying the origin Will return (0,0,0) if the origin was not specified.

Examples

HelloVizWorld.cxx.

12.146.3.5 GetOrigin() [2/2]

```
double gdcm::Image::GetOrigin (
            unsigned int idx) const
```

12.146.3.6 GetSlope()

```
double gdcm::Image::GetSlope () const  [inline]
```

12.146.3.7 GetSpacing() [1/2]

```
const double * gdcm::Image::GetSpacing () const
```

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an arbitrary 1 value when the spacing was not specified (ex. 2D image). WARNING: when the spacing is not specifier, a default value of 1 will be returned

12.146.3.8 GetSpacing() [2/2]

```
double gdcm::Image::GetSpacing (
            unsigned int idx) const
```

12.146.3.9 Print()

```
void gdcm::Image::Print (
            std::ostream & os) const  [override], [virtual]
```

print

Reimplemented from gdcm::Bitmap.

Examples

CompressImage.cxx, and PatchFile.cxx.

12.146.3.10 SetDirectionCosines() [1/3]

```
void gdcm::Image::SetDirectionCosines (
            const double dircos[6])
```

12.146.3.11 SetDirectionCosines() [2/3]

```
void gdcm::Image::SetDirectionCosines (
            const float dircos[6])
```

12.146.3.12 SetDirectionCosines() [3/3]

```
void gdcm::Image::SetDirectionCosines (
            unsigned int idx,
            double dircos)
```

12.146.3.13 SetIntercept()

```
void gdcm::Image::SetIntercept (
            double intercept)  [inline]
```

intercept

Examples

TemplateEmptyImage.cxx.

12.146.3.14 SetOrigin() [1/3]

```
void gdcm::Image::SetOrigin (
            const double origin[3])
```

12.146.3.15 SetOrigin() [2/3]

```
void gdcm::Image::SetOrigin (
            const float origin[3])
```

12.146.3.16 SetOrigin() [3/3]

```
void gdcm::Image::SetOrigin (
            unsigned int idx,
            double ori)
```

12.146.3.17 SetSlope()

```
void gdcm::Image::SetSlope (
            double slope)  [inline]
```

slope

Examples

TemplateEmptyImage.cxx.

12.146.3.18 SetSpacing() [1/2]

```
void gdcm::Image::SetSpacing (
            const double spacing[3])
```

Examples

csa2img.cxx, and iU22tomultisc.cxx.

12.146.3.19 SetSpacing() [2/2]

```
void gdcm::Image::SetSpacing (
            unsigned int idx,
            double spacing)
```

The documentation for this class was generated from the following file:

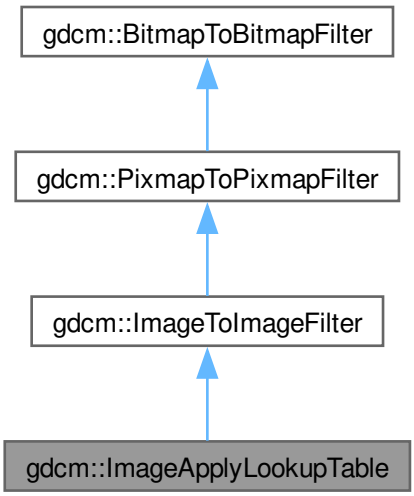
- gdcmImage.h

12.147 gdcm::ImageApplyLookupTable Class Reference

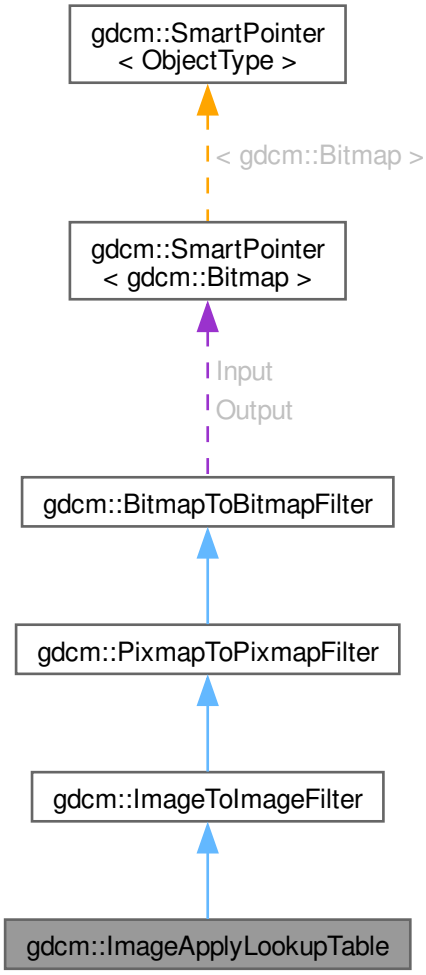
ImageApplyLookupTable class.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for gdcm::ImageApplyLookupTable:



Public Member Functions

- ImageApplyLookupTable ()
- ~ImageApplyLookupTable ()
- bool Apply ()

  Apply.
- void SetRGB8 (bool b)

  RGB8 ?

Public Member Functions inherited from gdcm::ImageToImageFilter

- ImageToImageFilter ()
- ~ImageToImageFilter ()=default
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Public Member Functions inherited from gdcm::PixmapToPixmapFilter

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()=default
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Public Member Functions inherited from gdcm::BitmapToBitmapFilter

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Additional Inherited Members

Protected Attributes inherited from gdcm::BitmapToBitmapFilter

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

12.147.1 Detailed Description

ImageApplyLookupTable class.

It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image

12.147.2 Constructor & Destructor Documentation

12.147.2.1 ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::ImageApplyLookupTable ()
```

12.147.2.2 ~ImageApplyLookupTable()

```
gdcm::ImageApplyLookupTable::~ImageApplyLookupTable ()
```

12.147.3 Member Function Documentation

12.147.3.1 Apply()

```
bool gdcm::ImageApplyLookupTable::Apply ()
```

Apply.

12.147.3.2 SetRGB8()

```
void gdcm::ImageApplyLookupTable::SetRGB8 (
            bool b)
```

RGB8 ?

The documentation for this class was generated from the following file:

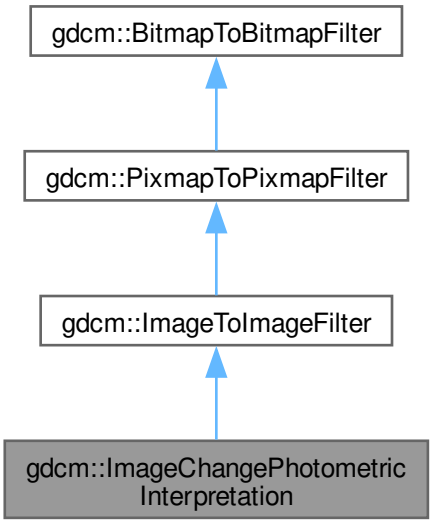
- gdcmImageApplyLookupTable.h

12.148 gdcm::ImageChangePhotometricInterpretation Class Reference

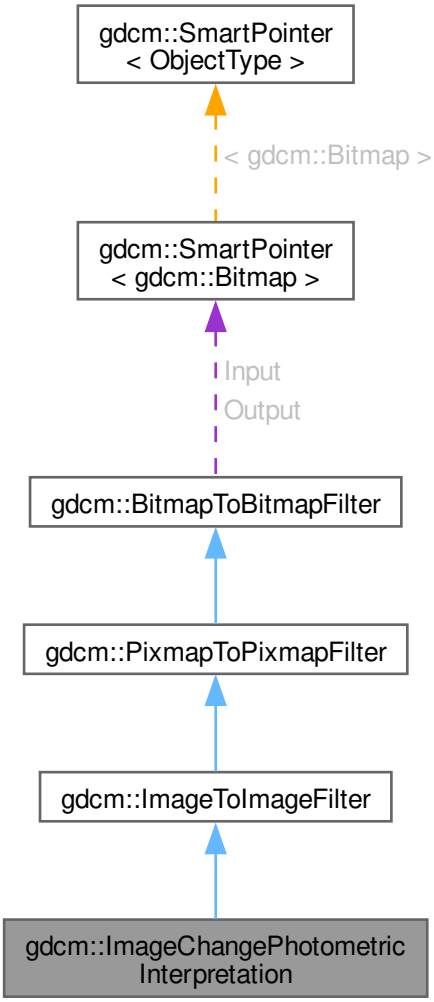
ImageChangePhotometricInterpretation class.

```
#include <gdcmImageChangePhotometricInterpretation.h>
```

Inheritance diagram for gdcm::ImageChangePhotometricInterpretation:



Collaboration diagram for gdcm::ImageChangePhotometricInterpretation:



Public Member Functions

- ImageChangePhotometricInterpretation ()
- ~ImageChangePhotometricInterpretation ()=default
- bool Change ()

  Change.
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)

  Set/Get requested PhotometricInterpretation.

Public Member Functions inherited from gdcm::ImageToImageFilter

- ImageToImageFilter ()
- ~ImageToImageFilter ()=default
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Public Member Functions inherited from gdcm::PixmapToPixmapFilter

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()=default
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Public Member Functions inherited from gdcm::BitmapToBitmapFilter

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Static Public Member Functions

- template<typename T>
  static void RGB2YBR (T ybr[3], const T rgb[3], unsigned short storedbits=8)
- template<typename T>
  static void YBR2RGB (T rgb[3], const T ybr[3], unsigned short storedbits=8)

Protected Member Functions

- bool ChangeMonochrome ()
- bool ChangeRGB2YBR ()
- bool ChangeYBR2RGB ()

Additional Inherited Members

Protected Attributes inherited from gdcm::BitmapToBitmapFilter

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

12.148.1 Detailed Description

ImageChangePhotometricInterpretation class.

Class to change the Photometric Interpretation of an input DICOM

12.148.2 Constructor & Destructor Documentation

12.148.2.1 ImageChangePhotometricInterpretation()

```
gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation () [inline]
```

12.148.2.2 ∼ImageChangePhotometricInterpretation()

```
gdcm::ImageChangePhotometricInterpretation::∼ImageChangePhotometricInterpretation () [default]
```

12.148.3 Member Function Documentation

12.148.3.1 Change()

```
bool gdcm::ImageChangePhotometricInterpretation::Change ()
```

Change.

References RGB2YBR(), and YBR2RGB().

12.148.3.2 ChangeMonochrome()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome () [protected]
```

12.148.3.3 ChangeRGB2YBR()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeRGB2YBR () [protected]
```

12.148.3.4 ChangeYBR2RGB()

```
bool gdcm::ImageChangePhotometricInterpretation::ChangeYBR2RGB () [protected]
```

12.148.3.5 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::ImageChangePhotometricInterpretation::GetPhotometric↩
Interpretation () const  [inline]
```

12.148.3.6 RGB2YBR()

```
template<typename T>
void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (
            T ybr[3],
            const T rgb[3],
            unsigned short storedbits = 8)  [static]
```

colorspace conversion (based on CCIR Recommendation 601-2) -> T.871

References gdcm::Clamp(), and gdcm::Round().

Referenced by Change().

12.148.3.7 SetPhotometricInterpretation()

```
void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi)  [inline]
```

Set/Get requested PhotometricInterpretation.

12.148.3.8 YBR2RGB()

```
template<typename T>
void gdcm::ImageChangePhotometricInterpretation::YBR2RGB (
            T rgb[3],
            const T ybr[3],
            unsigned short storedbits = 8)  [static]
```

References gdcm::Clamp(), and gdcm::Round().

Referenced by Change().

The documentation for this class was generated from the following file:

- gdcmImageChangePhotometricInterpretation.h

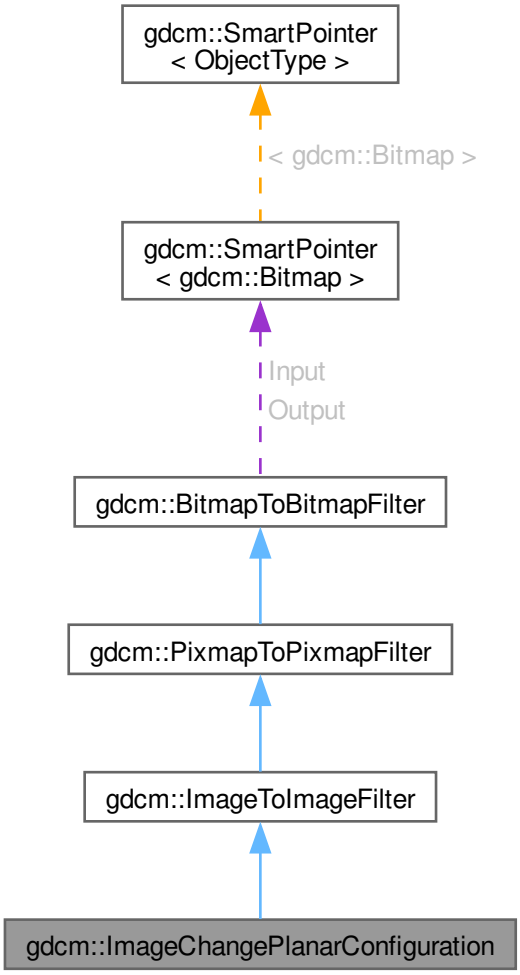
12.149 gdcm::ImageChangePlanarConfiguration Class Reference

ImageChangePlanarConfiguration class.

```
#include <gdcmImageChangePlanarConfiguration.h>
```

Inheritance diagram for gdcm::ImageChangePlanarConfiguration:

Collaboration diagram for gdcm::ImageChangePlanarConfiguration:



Public Member Functions

- ImageChangePlanarConfiguration ()
- ~ImageChangePlanarConfiguration ()=default
- bool Change ()

  Change.
- unsigned int GetPlanarConfiguration () const
- void SetPlanarConfiguration (unsigned int pc)

  Set/Get requested PlanarConfigation.

Public Member Functions inherited from gdcm::ImageToImageFilter

- ImageToImageFilter ()
- ~ImageToImageFilter ()=default
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Public Member Functions inherited from gdcm::PixmapToPixmapFilter

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()=default
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Public Member Functions inherited from gdcm::BitmapToBitmapFilter

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Static Public Member Functions

- template<typename T>
  static size_t RGBPixelsToRGBPlanes (T *r, T *g, T *b, const T *rgb, size_t s)
- template<typename T>
  static size_t RGBPlanesToRGBPixels (T *out, const T *r, const T *g, const T *b, size_t s)

Additional Inherited Members

Protected Attributes inherited from gdcm::BitmapToBitmapFilter

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

12.149.1 Detailed Description

ImageChangePlanarConfiguration class.

Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0

12.149.2 Constructor & Destructor Documentation

12.149.2.1 ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration () [inline]
```

12.149.2.2 ~ImageChangePlanarConfiguration()

```
gdcm::ImageChangePlanarConfiguration::~ImageChangePlanarConfiguration () [default]
```

12.149.3 Member Function Documentation

12.149.3.1 Change()

```
bool gdcm::ImageChangePlanarConfiguration::Change ()
```

Change.

12.149.3.2 GetPlanarConfiguration()

```
unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration () const [inline]
```

12.149.3.3 RGBPixelsToRGBPlanes()

```
template<typename T>
size_t gdcm::ImageChangePlanarConfiguration::RGBPixelsToRGBPlanes (
            T * r,
            T * g,
            T * b,
            const T * rgb,
            size_t s) [static]
```

Convert a regular RGB pixel image (R,G,B,R,G,B...) into a planar R,G,B image (R,R..,G,G...B,B)

Warning

this works on a frame basis, you need to loop over all frames in multiple frames image to apply this function

12.149.3.4 RGBPlanesToRGBPixels()

```
template<typename T>
size_t gdcm::ImageChangePlanarConfiguration::RGBPlanesToRGBPixels (
            T * out,
            const T * r,
            const T * g,
            const T * b,
            size_t s)  [static]
```

s is the size of one plane (r,g or b). Thus the output buffer needs to be at least 3∗s bytes long s can be seen as the number of RGB pixels in the output

12.149.3.5 SetPlanarConfiguration()

```
void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (
            unsigned int pc)  [inline]
```

Set/Get requested PlanarConfigation.

The documentation for this class was generated from the following file:

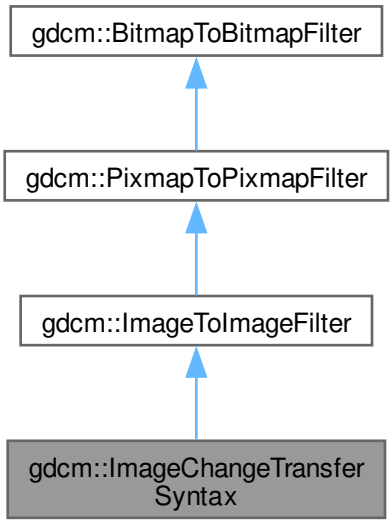
- gdcmImageChangePlanarConfiguration.h

12.150 gdcm::ImageChangeTransferSyntax Class Reference

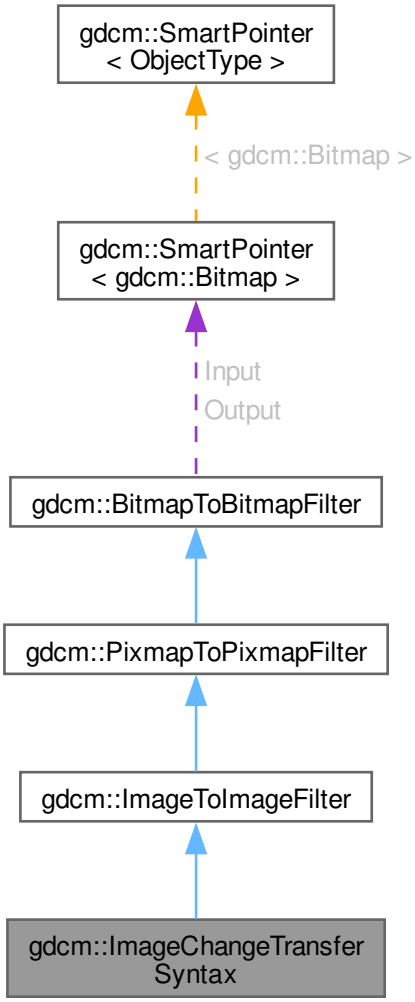
ImageChangeTransferSyntax class.

`#include <gdcmImageChangeTransferSyntax.h>`

Inheritance diagram for gdcm::ImageChangeTransferSyntax:



Collaboration diagram for gdcm::ImageChangeTransferSyntax:



Public Member Functions

- ImageChangeTransferSyntax ()
- ~ImageChangeTransferSyntax ()=default
- bool Change ()

  Change.
- const TransferSyntax & GetTransferSyntax () const

  Get Transfer Syntax.
- void SetCompressIconImage (bool b)
- void SetForce (bool f)

- void SetTransferSyntax (const TransferSyntax &ts)

  Set target Transfer Syntax.
- void SetUserCodec (ImageCodec *ic)

Public Member Functions inherited from gdcm::ImageToImageFilter

- ImageToImageFilter ()
- ~ImageToImageFilter ()=default
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Public Member Functions inherited from gdcm::PixmapToPixmapFilter

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()=default
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Public Member Functions inherited from gdcm::BitmapToBitmapFilter

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Protected Member Functions

- bool TryJPEG2000Codec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryJPEGCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryJPEGLSCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryRAWCodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)
- bool TryRLECodec (const DataElement &pixelde, Bitmap const &input, Bitmap &output)

Additional Inherited Members

Protected Attributes inherited from gdcm::BitmapToBitmapFilter

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

12.150.1 Detailed Description

ImageChangeTransferSyntax class.

Class to change the transfer syntax of an input DICOM

If only Force param is set but no input TransferSyntax is set, it is assumed that user only wants to inspect encapsulated stream (advanced dev. option).

When using UserCodec it is very important that the TransferSyntax (as set in SetTransferSyntax) is actually understood by UserCodec (ie. UserCodec->CanCode(TransferSyntax)). Otherwise the behavior is to use a default codec.

See also

JPEGCodec JPEGLSCodec JPEG2000Codec

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, and StandardizeFiles.cs.

12.150.2 Constructor & Destructor Documentation

12.150.2.1 ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax () [inline]
```

12.150.2.2 ~ImageChangeTransferSyntax()

```
gdcm::ImageChangeTransferSyntax::~ImageChangeTransferSyntax () [default]
```

12.150.3 Member Function Documentation

12.150.3.1 Change()

```
bool gdcm::ImageChangeTransferSyntax::Change ()
```

Change.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, and StandardizeFiles.cs.

12.150.3.2 GetTransferSyntax()

```
const TransferSyntax & gdcm::ImageChangeTransferSyntax::GetTransferSyntax () const  [inline]
```

Get Transfer Syntax.

12.150.3.3 SetCompressIconImage()

```
void gdcm::ImageChangeTransferSyntax::SetCompressIconImage (
            bool b)  [inline]
```

Decide whether or not to also compress the Icon Image using the same Transfer Syntax. Default is to simply decompress icon image

Examples

StandardizeFiles.cs.

12.150.3.4 SetForce()

```
void gdcm::ImageChangeTransferSyntax::SetForce (
            bool f)  [inline]
```

When target Transfer Syntax is identical to input target syntax, no operation is actually done. This is an issue when someone wants to re-compress using GDCM internal implementation a JPEG (for example) image

Examples

StandardizeFiles.cs.

12.150.3.5 SetTransferSyntax()

```
void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (
            const TransferSyntax & ts)  [inline]
```

Set target Transfer Syntax.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, and StandardizeFiles.cs.

12.150.3.6 SetUserCodec()

```
void gdcm::ImageChangeTransferSyntax::SetUserCodec (
            ImageCodec * ic)  [inline]
```

Allow user to specify exactly which codec to use. this is needed to specify special qualities or compression option.

Warning

if the codec 'ic' is not compatible with the TransferSyntax requested, it will not be used. It is the user responsibility to check that UserCodec->CanCode(TransferSyntax)

Examples

CompressLossyJPEG.cs.

12.150.3.7 TryJPEG2000Codec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output)  [protected]
```

12.150.3.8 TryJPEGCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output)  [protected]
```

12.150.3.9 TryJPEGLSCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output)  [protected]
```

12.150.3.10 TryRAWCodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output)  [protected]
```

12.150.3.11 TryRLECodec()

```
bool gdcm::ImageChangeTransferSyntax::TryRLECodec (
            const DataElement & pixelde,
            Bitmap const & input,
            Bitmap & output)  [protected]
```

The documentation for this class was generated from the following file:

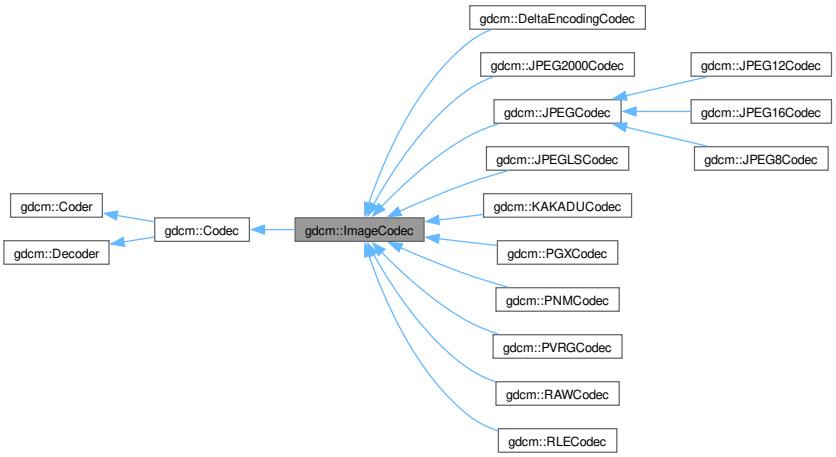
- gdcmImageChangeTransferSyntax.h

12.151 gdcm::ImageCodec Class Reference

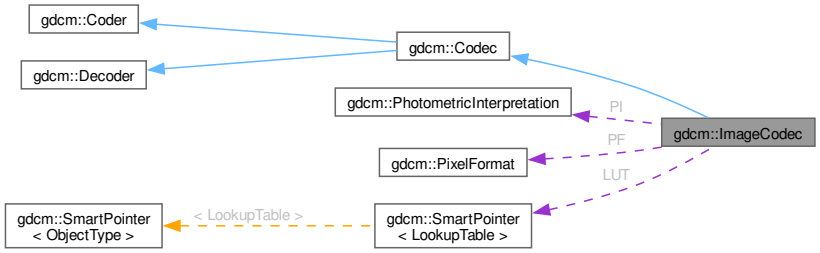
ImageCodec.

```
#include <gdcmImageCodec.h>
```

Inheritance diagram for gdcm::ImageCodec:



Collaboration diagram for gdcm::ImageCodec:



Public Member Functions

- ImageCodec ()
- ~ImageCodec () override
- bool CanCode (TransferSyntax const &) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &) const override

  Return whether this decoder support this transfer syntax (can decode it).
- bool CleanupUnusedBits (char *data, size_t datalen)
- virtual ImageCodec * Clone () const =0
- bool Decode (DataElement const &is_, DataElement &os) override

  Decode.
- const unsigned int * GetDimensions () const
- virtual bool GetHeaderInfo (std::istream &is_, TransferSyntax &ts)
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Types

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions

- virtual bool AppendFrameEncode (std::ostream &out, const char ∗data, size_t datalen)
- virtual bool AppendRowEncode (std::ostream &out, const char ∗data, size_t datalen)
- bool DecodeByStreams (std::istream &is_, std::ostream &os) override
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsFrameEncoder ()
- virtual bool IsRowEncoder ()
- virtual bool IsValid (PhotometricInterpretation const &pi)
- virtual bool StartEncode (std::ostream &os)
- virtual bool StopEncode (std::ostream &os)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char ∗bv, unsigned long len, std::ostream &os)

Protected Attributes

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

Friends

- class FileChangeTransferSyntax
- class ImageChangePhotometricInterpretation

12.151.1 Detailed Description

ImageCodec.

Note

Main codec, this is a central place for all implementation

Examples

FileChangeTSLossy.cs.

12.151.2 Member Typedef Documentation

12.151.2.1 LUTPtr

```
typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr  [protected]
```

12.151.3 Constructor & Destructor Documentation

12.151.3.1 ImageCodec()

```
gdcm::ImageCodec::ImageCodec ()
```

Referenced by Clone(), gdcm::JPEG2000Codec::Clone(), gdcm::JPEGCodec::Clone(), gdcm::JPEGLSCodec::Clone(), gdcm::KAKADUCodec::Clone(), gdcm::PGXCodec::Clone(), gdcm::PNMCodec::Clone(), gdcm::PVRGCodec::Clone(), gdcm::RAWCodec::Clone(), and gdcm::RLECodec::Clone().

12.151.3.2 ~ImageCodec()

```
gdcm::ImageCodec::~ImageCodec ()  [override]
```

12.151.4 Member Function Documentation

12.151.4.1 AppendFrameEncode()

```
virtual bool gdcm::ImageCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen)  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

12.151.4.2 AppendRowEncode()

```
virtual bool gdcm::ImageCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen)  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

12.151.4.3 CanCode()

```
bool gdcm::ImageCodec::CanCode (
            TransferSyntax const & ) const  [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Implements gdcm::Coder.

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.151.4.4 CanDecode()

```
bool gdcm::ImageCodec::CanDecode (
            TransferSyntax const & ) const  [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Implements gdcm::Decoder.

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.151.4.5 CleanupUnusedBits()

```
bool gdcm::ImageCodec::CleanupUnusedBits (
            char * data,
            size_t datalen)
```

12.151.4.6 Clone()

```
virtual ImageCodec * gdcm::ImageCodec::Clone () const  [pure virtual]
```

Implemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

References ImageCodec().

12.151.4.7 Decode()

```
bool gdcm::ImageCodec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::KAKADUCodec, gdcm::PVRGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.151.4.8 DecodeByStreams()

```
bool gdcm::ImageCodec::DecodeByStreams (
            std::istream & is_,
            std::ostream & os)  [override], [protected], [virtual]
```

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG2000Codec, gdcm::JPEG8Codec, gdcm::JPEGCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.151.4.9 DoByteSwap()

```
bool gdcm::ImageCodec::DoByteSwap (
            std::istream & is_,
            std::ostream & os)  [protected]
```

12.151.4.10 DoInvertMonochrome()

```
bool gdcm::ImageCodec::DoInvertMonochrome (
            std::istream & is_,
            std::ostream & os)  [protected]
```

12.151.4.11 DoOverlayCleanup()

```
bool gdcm::ImageCodec::DoOverlayCleanup (
            std::istream & is_,
            std::ostream & os)  [protected]
```

12.151.4.12 DoPaddedCompositePixelCode()

```
bool gdcm::ImageCodec::DoPaddedCompositePixelCode (
            std::istream & is_,
            std::ostream & os)  [protected]
```

12.151.4.13 DoPlanarConfiguration()

```
bool gdcm::ImageCodec::DoPlanarConfiguration (
            std::istream & is_,
            std::ostream & os)  [protected]
```

12.151.4.14 DoSimpleCopy()

```
bool gdcm::ImageCodec::DoSimpleCopy (
            std::istream & is_,
            std::ostream & os)  [protected]
```

12.151.4.15 DoYBR()

```
bool gdcm::ImageCodec::DoYBR (
            std::istream & is_,
            std::ostream & os)  [protected]
```

12.151.4.16 DoYBRFull422()

```
bool gdcm::ImageCodec::DoYBRFull422 (
            std::istream & is_,
            std::ostream & os)  [protected]
```

12.151.4.17 GetDimensions()

```
const unsigned int * gdcm::ImageCodec::GetDimensions () const  [inline]
```

References Dimensions.

12.151.4.18 GetHeaderInfo()

```
virtual bool gdcm::ImageCodec::GetHeaderInfo (
            std::istream & is_,
            TransferSyntax & ts)  [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG2000Codec, gdcm::JPEG8Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::PGXCodec, gdcm::PNMCodec, gdcm::RAWCodec, and gdcm::RLECodec.

12.151.4.19 GetLossyFlag()

```
bool gdcm::ImageCodec::GetLossyFlag () const
```

12.151.4.20 GetLUT()

```
const LookupTable & gdcm::ImageCodec::GetLUT () const  [inline]
```

References LUT.

12.151.4.21 GetNeedByteSwap()

```
bool gdcm::ImageCodec::GetNeedByteSwap () const  [inline]
```

References NeedByteSwap.

12.151.4.22 GetNumberOfDimensions()

```
unsigned int gdcm::ImageCodec::GetNumberOfDimensions () const
```

12.151.4.23 GetPhotometricInterpretation()

```
const PhotometricInterpretation & gdcm::ImageCodec::GetPhotometricInterpretation () const
```

12.151.4.24 GetPixelFormat() [1/2]

```
PixelFormat & gdcm::ImageCodec::GetPixelFormat ()  [inline]
```

Examples

GetJPEGSamplePrecision.cxx.

References PF.

12.151.4.25 GetPixelFormat() [2/2]

```
const PixelFormat & gdcm::ImageCodec::GetPixelFormat () const  [inline]
```

References PF.

12.151.4.26 GetPlanarConfiguration()

```
unsigned int gdcm::ImageCodec::GetPlanarConfiguration () const  [inline]
```

References PlanarConfiguration.

12.151.4.27 IsFrameEncoder()

```
virtual bool gdcm::ImageCodec::IsFrameEncoder () [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

12.151.4.28 IsLossy()

```
bool gdcm::ImageCodec::IsLossy () const
```

12.151.4.29 IsRowEncoder()

```
virtual bool gdcm::ImageCodec::IsRowEncoder () [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

12.151.4.30 IsValid()

```
virtual bool gdcm::ImageCodec::IsValid (
            PhotometricInterpretation const & pi) [protected], [virtual]
```

Reimplemented in gdcm::JPEGCodec.

12.151.4.31 SetDimensions() [1/2]

```
void gdcm::ImageCodec::SetDimensions (
            const std::vector< unsigned int > & d)
```

12.151.4.32 SetDimensions() [2/2]

```
void gdcm::ImageCodec::SetDimensions (
            const unsigned int d[3])
```

Examples

ExtractIconFromFile.cxx.

12.151.4.33 SetLossyFlag()

```
void gdcm::ImageCodec::SetLossyFlag (
            bool l)
```

12.151.4.34 SetLUT()

```
void gdcm::ImageCodec::SetLUT (
            LookupTable const & lut)  [inline]
```

Examples

ExtractIconFromFile.cxx.

References LUT.

12.151.4.35 SetNeedByteSwap()

```
void gdcm::ImageCodec::SetNeedByteSwap (
            bool b)  [inline]
```

References NeedByteSwap.

12.151.4.36 SetNeedOverlayCleanup()

```
void gdcm::ImageCodec::SetNeedOverlayCleanup (
            bool b)  [inline]
```

References NeedOverlayCleanup.

12.151.4.37 SetNumberOfDimensions()

```
void gdcm::ImageCodec::SetNumberOfDimensions (
            unsigned int dim)
```

12.151.4.38 SetPhotometricInterpretation()

```
void gdcm::ImageCodec::SetPhotometricInterpretation (
            PhotometricInterpretation const & pi)
```

Examples

ExtractIconFromFile.cxx.

12.151.4.39 SetPixelFormat()

```
virtual void gdcm::ImageCodec::SetPixelFormat (
            PixelFormat const & pf)  [inline], [virtual]
```

Reimplemented in gdcm::JPEGCodec.

Examples

ExtractIconFromFile.cxx.

References PF.

12.151.4.40 SetPlanarConfiguration()

```
void gdcm::ImageCodec::SetPlanarConfiguration (
            unsigned int pc)  [inline]
```

References PlanarConfiguration.

12.151.4.41 StartEncode()

```
virtual bool gdcm::ImageCodec::StartEncode (
            std::ostream & os)  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

12.151.4.42 StopEncode()

```
virtual bool gdcm::ImageCodec::StopEncode (
            std::ostream & os)  [protected], [virtual]
```

Reimplemented in gdcm::JPEG2000Codec, gdcm::JPEGCodec, gdcm::JPEGLSCodec, and gdcm::RLECodec.

12.151.5 Friends And Related Symbol Documentation

12.151.5.1 FileChangeTransferSyntax

```
friend class FileChangeTransferSyntax  [friend]
```

This is a high level API to encode in a streaming fashion. Each plugin will handle differently the caching mechanism so that a limited memory is used when compressing dataset. Codec will fall into two categories:

- Full row encoder: only a single scanline (row) of data is needed to be loaded at a time;
- Full frame encoder (default): a complete frame (row x col) is needed to be loaded at a time

References FileChangeTransferSyntax.

Referenced by FileChangeTransferSyntax.

12.151.5.2 ImageChangePhotometricInterpretation

```
friend class ImageChangePhotometricInterpretation [friend]
```

References ImageChangePhotometricInterpretation.

Referenced by ImageChangePhotometricInterpretation.

12.151.6 Member Data Documentation

12.151.6.1 Dimensions

```
unsigned int gdcm::ImageCodec::Dimensions[3] [protected]
```

Referenced by GetDimensions().

12.151.6.2 LossyFlag

```
bool gdcm::ImageCodec::LossyFlag [protected]
```

12.151.6.3 LUT

```
LUTPtr gdcm::ImageCodec::LUT [protected]
```

Referenced by GetLUT(), and SetLUT().

12.151.6.4 NeedByteSwap

```
bool gdcm::ImageCodec::NeedByteSwap [protected]
```

Referenced by GetNeedByteSwap(), and SetNeedByteSwap().

12.151.6.5 NeedOverlayCleanup

```
bool gdcm::ImageCodec::NeedOverlayCleanup [protected]
```

Referenced by SetNeedOverlayCleanup().

12.151.6.6 NumberOfDimensions

```
unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]
```

12.151.6.7 PF

```
PixelFormat gdcm::ImageCodec::PF [protected]
```

Referenced by GetPixelFormat(), GetPixelFormat(), and SetPixelFormat().

12.151.6.8 PI

```
PhotometricInterpretation gdcm::ImageCodec::PI [protected]
```

12.151.6.9 PlanarConfiguration

```
unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]
```

Referenced by GetPlanarConfiguration(), and SetPlanarConfiguration().

12.151.6.10 RequestPaddedCompositePixelCode

```
bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]
```

12.151.6.11 RequestPlanarConfiguration

```
bool gdcm::ImageCodec::RequestPlanarConfiguration [protected]
```

The documentation for this class was generated from the following file:

- gdcmImageCodec.h

12.152 gdcm::ImageConverter Class Reference

Image Converter.

```
#include <gdcmImageConverter.h>
```

Public Member Functions

- ImageConverter ()
- ~ImageConverter ()
- void Convert ()
- const Image & GetOuput () const
- void SetInput (Image const &input)

12.152.1 Detailed Description

Image Converter.

Note

This is the class used to convert from on Image to another This is typically used to convert let say YBR JPEG compressed Image to a RAW RGB Image. So that the buffer can be directly pass to third party application. This filter is application level and not integrated directly in GDCM

12.152.2 Constructor & Destructor Documentation

12.152.2.1 ImageConverter()

```
gdcm::ImageConverter::ImageConverter ()
```

12.152.2.2 ~ImageConverter()

```
gdcm::ImageConverter::~ImageConverter ()
```

12.152.3 Member Function Documentation

12.152.3.1 Convert()

```
void gdcm::ImageConverter::Convert ()
```

12.152.3.2 GetOuput()

```
const Image & gdcm::ImageConverter::GetOuput () const
```

12.152.3.3 SetInput()

```
void gdcm::ImageConverter::SetInput (
            Image const & input)
```

The documentation for this class was generated from the following file:

- gdcmImageConverter.h

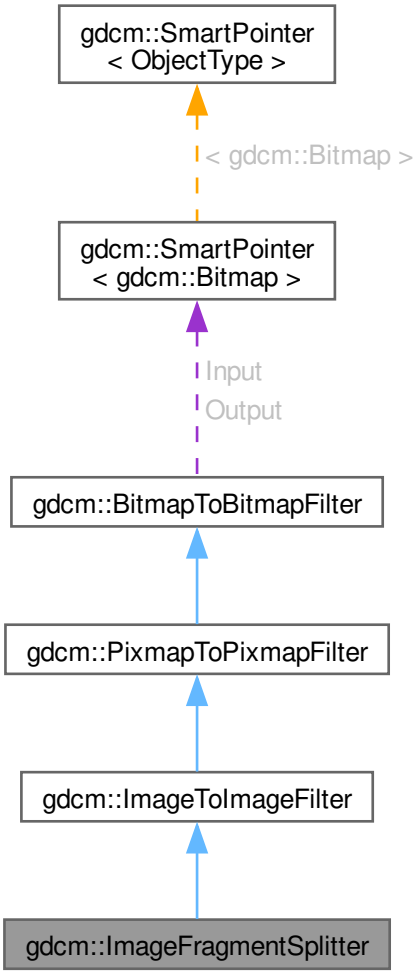
12.153 gdcm::ImageFragmentSplitter Class Reference

ImageFragmentSplitter class.

```
#include <gdcmImageFragmentSplitter.h>
```

Inheritance diagram for gdcm::ImageFragmentSplitter:

Collaboration diagram for gdcm::ImageFragmentSplitter:



Public Member Functions

- ImageFragmentSplitter ()
- ~ImageFragmentSplitter ()=default
- unsigned int GetFragmentSizeMax () const
- void SetForce (bool f)
- void SetFragmentSizeMax (unsigned int fragsize)

  FragmentSizeMax needs to be an even number.
- bool Split ()

  Split.

Public Member Functions inherited from gdcm::ImageToImageFilter

- ImageToImageFilter ()
- ~ImageToImageFilter ()=default
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Public Member Functions inherited from gdcm::PixmapToPixmapFilter

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()=default
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Public Member Functions inherited from gdcm::BitmapToBitmapFilter

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Additional Inherited Members

Protected Attributes inherited from gdcm::BitmapToBitmapFilter

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

12.153.1 Detailed Description

ImageFragmentSplitter class.

For single frame image, DICOM standard allow splitting the frame into multiple fragments

12.153.2 Constructor & Destructor Documentation

12.153.2.1 ImageFragmentSplitter()

```
gdcm::ImageFragmentSplitter::ImageFragmentSplitter () [inline]
```

12.153.2.2 ~ImageFragmentSplitter()

```
gdcm::ImageFragmentSplitter::~ImageFragmentSplitter () [default]
```

12.153.3 Member Function Documentation

12.153.3.1 GetFragmentSizeMax()

```
unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax () const [inline]
```

12.153.3.2 SetForce()

```
void gdcm::ImageFragmentSplitter::SetForce (
            bool f) [inline]
```

When file already has all it's segment < FragmentSizeMax there is not need to run the filter. Unless the user explicitly say 'force' recomputation !

12.153.3.3 SetFragmentSizeMax()

```
void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (
            unsigned int fragsize)
```

FragmentSizeMax needs to be an even number.

12.153.3.4 Split()

```
bool gdcm::ImageFragmentSplitter::Split ()
```

Split.

The documentation for this class was generated from the following file:

- gdcmImageFragmentSplitter.h

12.154 gdcm::ImageHelper Class Reference

ImageHelper (internal class, not intended for user level).

```
#include <gdcmImageHelper.h>
```

Static Public Member Functions

- static MediaStorage ComputeMediaStorageFromModality (const char *modality, unsigned int dimension=2, PixelFormat const &pf=PixelFormat(), PhotometricInterpretation const &pi=PhotometricInterpretation(), double rescaleintercept=0, double rescaleslope=1)

  Moved from MediaStorage here, since we need extra info stored in PixelFormat & PhotometricInterpretation.
- static bool ComputeSpacingFromImagePositionPatient (const std::vector< double > &imageposition, std::vector< double > &spacing)

  DO NOT USE.
- static std::vector< unsigned int > GetDimensionsValue (const File &f)
- static bool GetDirectionCosinesFromDataSet (DataSet const &ds, std::vector< double > &dircos)
- static std::vector< double > GetDirectionCosinesValue (File const &f)
- static bool GetForcePixelSpacing ()
- static bool GetForceRescaleInterceptSlope ()
- static SmartPointer< LookupTable > GetLUT (File const &f)

  returns the lookup table of an image file
- static std::vector< double > GetOriginValue (File const &f)

  Set/Get Origin (IPP) from/to a file.
- static PhotometricInterpretation GetPhotometricInterpretationValue (File const &f)
- static PixelFormat GetPixelFormatValue (const File &f)
- static unsigned int GetPlanarConfigurationValue (const File &f)
- static bool GetPMSRescaleInterceptSlope ()
- static const ByteValue * GetPointerFromElement (Tag const &tag, File const &f)
- static bool GetRealWorldValueMappingContent (File const &f, RealWorldValueMappingContent &rwvmc)
- static std::vector< double > GetRescaleInterceptSlopeValue (File const &f)
- static bool GetSecondaryCaptureImagePlaneModule ()
- static std::vector< double > GetSpacingValue (File const &f)

  Set/Get Spacing from/to a File.
- static void SetDimensionsValue (File &f, const Pixmap &img)
- static void SetDirectionCosinesValue (DataSet &ds, const std::vector< double > &dircos)
- static void SetForcePixelSpacing (bool)
- static void SetForceRescaleInterceptSlope (bool)
- static void SetOriginValue (DataSet &ds, const Image &img)
- static void SetPMSRescaleInterceptSlope (bool)
- static void SetRescaleInterceptSlopeValue (File &f, const Image &img)
- static void SetSecondaryCaptureImagePlaneModule (bool)
- static void SetSpacingValue (DataSet &ds, const std::vector< double > &spacing)

Static Protected Member Functions

- static Tag GetSpacingTagFromMediaStorage (MediaStorage const &ms)
- static Tag GetZSpacingTagFromMediaStorage (MediaStorage const &ms)

12.154.1 Detailed Description

ImageHelper (internal class, not intended for user level).

Helper for writing World images in DICOM. DICOM has a 'template' approach to image where MR Image Storage are distinct object from Enhanced MR Image Storage. For example the Pixel Spacing in one object is not at the same position (ie Tag) as in the other this class is the central (read: fragile) place where all the dispatching is done from a unified view of a world image (typically VTK or ITK point of view) down to the low level DICOM point of view.

Warning

: do not expect the API of this class to be maintained at any point, since as Modalities are added the API might have to be augmented or behavior changed to cope with new modalities.

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, and ExtractOneFrame.cs.

12.154.2 Member Function Documentation

12.154.2.1 ComputeMediaStorageFromModality()

```
MediaStorage gdcm::ImageHelper::ComputeMediaStorageFromModality (
            const char * modality,
            unsigned int dimension = 2,
            PixelFormat const & pf = PixelFormat(),
            PhotometricInterpretation const & pi = PhotometricInterpretation(),
            double rescaleintercept = 0,
            double rescaleslope = 1)  [static]
```

Moved from MediaStorage here, since we need extra info stored in PixelFormat & PhotometricInterpretation.

12.154.2.2 ComputeSpacingFromImagePositionPatient()

```
bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (
            const std::vector< double > & imageposition,
            std::vector< double > & spacing)  [static]
```

DO NOT USE.

12.154.2.3 GetDimensionsValue()

```
std::vector< unsigned int > gdcm::ImageHelper::GetDimensionsValue (
            const File & f)  [static]
```

This function checks tags (0x0028, 0x0010) and (0x0028, 0x0011) for the rows and columns of the image in pixels (as opposed to actual distances). The output is {col , row}

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.154.2.4 GetDirectionCosinesFromDataSet()

```
bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (
            DataSet const & ds,
            std::vector< double > & dircos)  [static]
```

12.154.2.5 GetDirectionCosinesValue()

```
std::vector< double > gdcm::ImageHelper::GetDirectionCosinesValue (
            File const & f)  [static]
```

Get Direction Cosines (IOP) from/to a file Requires a file because mediastorage must be known

12.154.2.6 GetForcePixelSpacing()

```
bool gdcm::ImageHelper::GetForcePixelSpacing ()  [static]
```

12.154.2.7 GetForceRescaleInterceptSlope()

```
bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ()  [static]
```

12.154.2.8 GetLUT()

```
SmartPointer< LookupTable > gdcm::ImageHelper::GetLUT (
            File const & f)  [static]
```

returns the lookup table of an image file

12.154.2.9 GetOriginValue()

```
std::vector< double > gdcm::ImageHelper::GetOriginValue (
            File const & f)  [static]
```

Set/Get Origin (IPP) from/to a file.

12.154.2.10 GetPhotometricInterpretationValue()

```
PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (
            File const & f)  [static]
```

Examples

ExtractImageRegion.cs.

12.154.2.11 GetPixelFormatValue()

```
PixelFormat gdcm::ImageHelper::GetPixelFormatValue (
            const File & f)  [static]
```

This function returns pixel information about an image from its dataset That includes samples per pixel and bit depth (in that order)

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, and ExtractOneFrame.cs.

12.154.2.12 GetPlanarConfigurationValue()

```
unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (
            const File & f)  [static]
```

12.154.2.13 GetPMSRescaleInterceptSlope()

```
bool gdcm::ImageHelper::GetPMSRescaleInterceptSlope ()  [static]
```

12.154.2.14 GetPointerFromElement()

```
const ByteValue * gdcm::ImageHelper::GetPointerFromElement (
            Tag const & tag,
            File const & f)  [static]
```

12.154.2.15 GetRealWorldValueMappingContent()

```
bool gdcm::ImageHelper::GetRealWorldValueMappingContent (
            File const & f,
            RealWorldValueMappingContent & rwvmc)  [static]
```

12.154.2.16 GetRescaleInterceptSlopeValue()

```
std::vector< double > gdcm::ImageHelper::GetRescaleInterceptSlopeValue (
            File const & f)  [static]
```

Set/Get shift/scale from/to a file

Warning

this function reads/sets the Slope/Intercept in appropriate class storage, but also Grid Scaling in RT Dose Storage Can't take a dataset because the mediastorage of the file must be known

12.154.2.17 GetSecondaryCaptureImagePlaneModule()

```
bool gdcm::ImageHelper::GetSecondaryCaptureImagePlaneModule ()  [static]
```

12.154.2.18 GetSpacingTagFromMediaStorage()

```
Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (
            MediaStorage const & ms)  [static], [protected]
```

12.154.2.19 GetSpacingValue()

```
std::vector< double > gdcm::ImageHelper::GetSpacingValue (
            File const & f)  [static]
```

Set/Get Spacing from/to a File.

12.154.2.20 GetZSpacingTagFromMediaStorage()

```
Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (
            MediaStorage const & ms)  [static], [protected]
```

12.154.2.21 SetDimensionsValue()

```
void gdcm::ImageHelper::SetDimensionsValue (
            File & f,
            const Pixmap & img)  [static]
```

12.154.2.22 SetDirectionCosinesValue()

```
void gdcm::ImageHelper::SetDirectionCosinesValue (
            DataSet & ds,
            const std::vector< double > & dircos)  [static]
```

Set Direction Cosines (IOP) from/to a file When IOD does not defines what is IOP (eg. typically Secondary Capture Image Storage) this call will simply remove the IOP attribute. Else in case of MR/CT image storage, this call will properly lookup the correct attribute to store the IOP.

12.154.2.23 SetForcePixelSpacing()

```
void gdcm::ImageHelper::SetForcePixelSpacing (
            bool )  [static]
```

GDCM 1.x compatibility issue: When using ReWrite an MR Image Storage would be rewritten as Secondary Capture Object while still having a Pixel Spacing tag (0028,0030). If you have deal with those files, use this very special flag to handle them Unless explicitly set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel Spacing of the Image

12.154.2.24 SetForceRescaleInterceptSlope()

```
void gdcm::ImageHelper::SetForceRescaleInterceptSlope (
            bool )  [static]
```

GDCM 1.x compatibility issue: Do not use anymore. This hack was used for some MR Image Storage generated by Philips Modality. When "Combine MR Rescaling" is set to TRUE, rescaling is removed. But when set to FALSE, the Modality LUT was exported. Internally GDCM now handles this gracefully.

12.154.2.25 SetOriginValue()

```
void gdcm::ImageHelper::SetOriginValue (
            DataSet & ds,
            const Image & img)  [static]
```

12.154.2.26 SetPMSRescaleInterceptSlope()

```
void gdcm::ImageHelper::SetPMSRescaleInterceptSlope (
            bool )  [static]
```

Since GDCM 2.6.1 Philips Medical System are read using the Private Field For Rescale Slope/Intercept by default. This mechanism can be deactivated using the following API: This option has no effect when ForceRescaleInterceptSlope is set to true GDCM will only read those private attribute but never write them out.

12.154.2.27 SetRescaleInterceptSlopeValue()

```
void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (
            File & f,
            const Image & img)  [static]
```

12.154.2.28 SetSecondaryCaptureImagePlaneModule()

```
void gdcm::ImageHelper::SetSecondaryCaptureImagePlaneModule (
            bool )  [static]
```

Opt into Image Plane Module for Secondary Capture Image Storage Enable reading Image Position Patient (IPP), Image Orientation Patient (IOP) and Pixel Spacing (0028,0030) This is a custom extension for some existing dataset (academic)

12.154.2.29 SetSpacingValue()

```
void gdcm::ImageHelper::SetSpacingValue (
            DataSet & ds,
            const std::vector< double > & spacing)  [static]
```

Warning

You need to call SetSpacingValue after SetOriginValue / SetDirectionCosinesValue

The documentation for this class was generated from the following file:

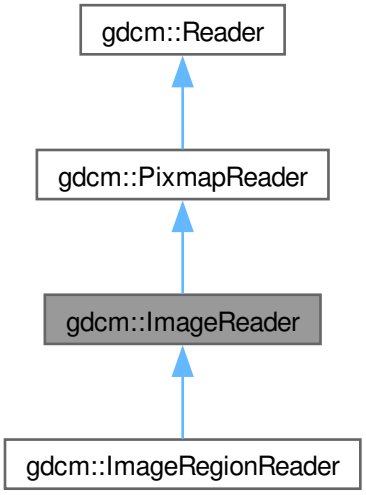
- gdcmImageHelper.h

12.155 gdcm::ImageReader Class Reference

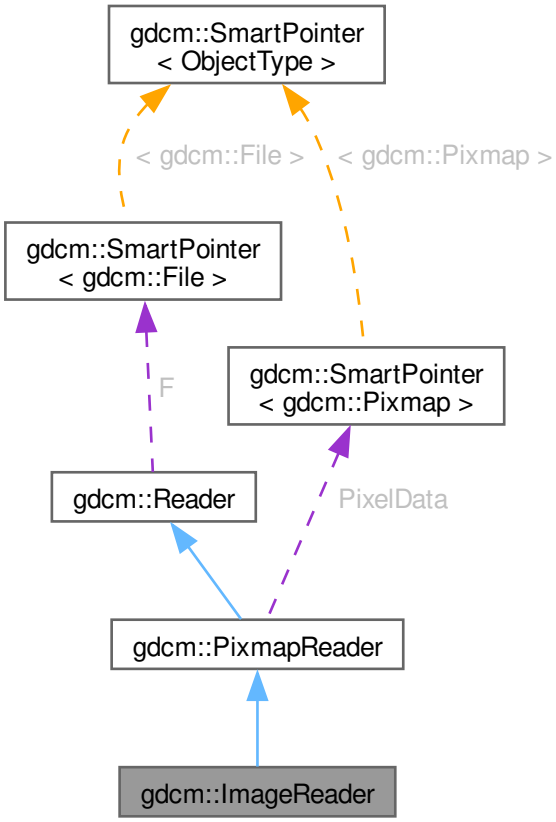
ImageReader.

```
#include <gdcmImageReader.h>
```

Inheritance diagram for gdcm::ImageReader:



Collaboration diagram for gdcm::ImageReader:



Public Member Functions

- ImageReader ()
- ~ImageReader () override
- Image & GetImage ()
- const Image & GetImage () const

  Return the read image.
- bool Read () override

Public Member Functions inherited from gdcm::PixmapReader

- PixmapReader ()
- ~PixmapReader () override
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const

  Return the read image (need to call Read() first).
- bool Read () override

Public Member Functions inherited from gdcm::Reader

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- File & GetFile ()

  Set/Get File.
- const File & GetFile () const

  Set/Get File.
- size_t GetStreamCurrentPosition () const
- bool ReadSelectedPrivateTags (std::set< PrivateTag > const &ptags, bool readvalues=true)

  Will only read the specified selected private tags.
- bool ReadSelectedTags (std::set< Tag > const &tags, bool readvalues=true)

  Will only read the specified selected tags.
- bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())
- void SetFile (File &file)

  Set/Get File.
- void SetFileName (const char *filename_native)
- void SetStream (std::istream &input_stream)

  Set the open-ed stream directly.

Protected Member Functions

- bool ReadACRNEMAImage () override
- bool ReadImage (MediaStorage const &ms) override

Protected Member Functions inherited from gdcm::PixmapReader

- bool ReadImageInternal (MediaStorage const &ms, bool handlepixeldata=true)

Protected Member Functions inherited from gdcm::Reader

- std::istream * GetStreamPtr () const
- bool ReadDataSet ()
- bool ReadMetaInformation ()
- bool ReadPreamble ()

Additional Inherited Members

Protected Attributes inherited from gdcm::PixmapReader

- SmartPointer< Pixmap > PixelData

Protected Attributes inherited from gdcm::Reader

- SmartPointer< File > F

12.155.1 Detailed Description

ImageReader.

Note

its role is to convert the DICOM DataSet into a Image representation Image is different from Pixmap has it has a position and a direction in Space.

See also

Image

Examples

BasicImageAnonymizer.cs, CheckBigEndianBug.cxx, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, MrProtocol.cxx, PatchFile.cxx, PrintLUT.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, and threadgdcm.cxx.

12.155.2 Constructor & Destructor Documentation

12.155.2.1 ImageReader()

```
gdcm::ImageReader::ImageReader ()
```

12.155.2.2 ~ImageReader()

```
gdcm::ImageReader::~ImageReader ()  [override]
```

12.155.3 Member Function Documentation

12.155.3.1 GetImage() [1/2]

```
Image & gdcm::ImageReader::GetImage ()
```

12.155.3.2 GetImage() [2/2]

```
const Image & gdcm::ImageReader::GetImage () const
```

Return the read image.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, ExtractImageRegionWithLUT.cs, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, PatchFile.cxx, PrintLUT.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, TemplateEmptyImage.cxx, and threadgdcm.cxx.

12.155.3.3 Read()

```
bool gdcm::ImageReader::Read () [override], [virtual]
```

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an Image.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::ImageRegionReader.

Examples

BasicImageAnonymizer.cs, CheckBigEndianBug.cxx, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, DecompressImage.cs, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, PrintLUT.cxx, ReadMultiTimesException.cxx, RescaleImage.cs, and threadgdcm.cxx.

12.155.3.4 ReadACRNEMAImage()

```
bool gdcm::ImageReader::ReadACRNEMAImage () [override], [protected], [virtual]
```

Reimplemented from gdcm::PixmapReader.

12.155.3.5 ReadImage()

```
bool gdcm::ImageReader::ReadImage (
            MediaStorage const & ms) [override], [protected], [virtual]
```

Reimplemented from gdcm::PixmapReader.

The documentation for this class was generated from the following file:

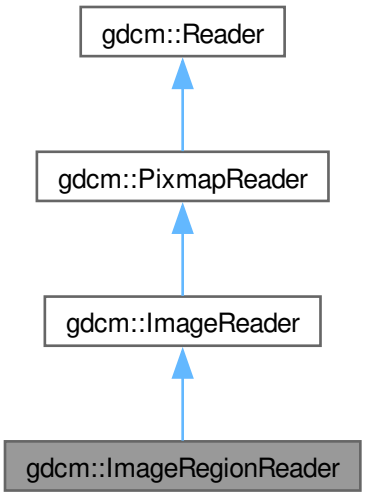
- gdcmImageReader.h

12.156 gdcm::ImageRegionReader Class Reference

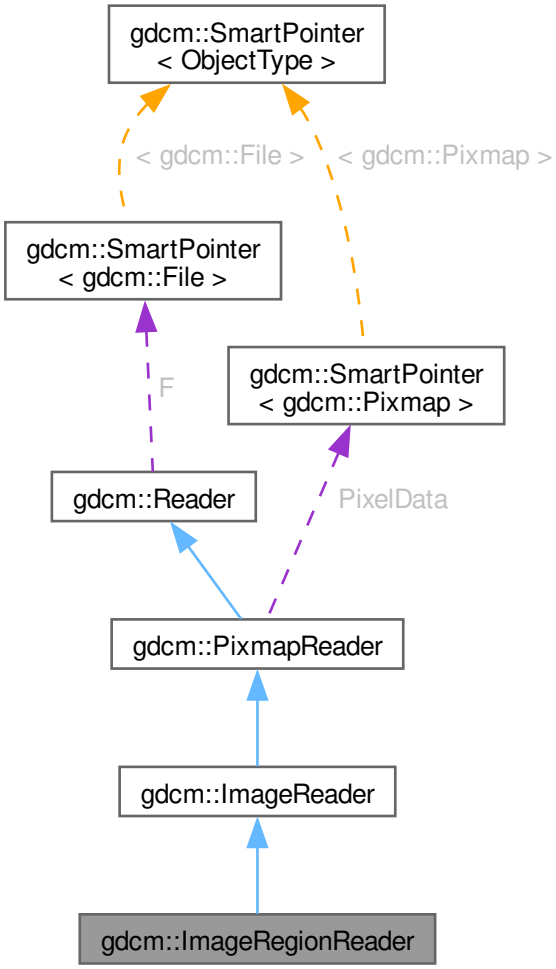
ImageRegionReader.

```
#include <gdcmImageRegionReader.h>
```

Inheritance diagram for gdcm::ImageRegionReader:



Collaboration diagram for gdcm::ImageRegionReader:



Public Member Functions

- ImageRegionReader ()
- ~ImageRegionReader () override
- size_t ComputeBufferLength () const
- Region const & GetRegion () const
- bool ReadInformation ()
- bool ReadIntoBuffer (char *inreadbuffer, size_t buflen)
- void SetRegion (Region const &region)

  Set/Get Region to be read.

Public Member Functions inherited from gdcm::ImageReader

- ImageReader ()
- ~ImageReader () override
- Image & GetImage ()
- const Image & GetImage () const

  Return the read image.

Public Member Functions inherited from gdcm::PixmapReader

- PixmapReader ()
- ~PixmapReader () override
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const

  Return the read image (need to call Read() first).
- bool Read () override

Public Member Functions inherited from gdcm::Reader

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- File & GetFile ()

  Set/Get File.
- const File & GetFile () const

  Set/Get File.
- size_t GetStreamCurrentPosition () const
- bool ReadSelectedPrivateTags (std::set< PrivateTag > const &ptags, bool readvalues=true)

  Will only read the specified selected private tags.
- bool ReadSelectedTags (std::set< Tag > const &tags, bool readvalues=true)

  Will only read the specified selected tags.
- bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())
- void SetFile (File &file)

  Set/Get File.
- void SetFileName (const char *filename_native)
- void SetStream (std::istream &input_stream)

  Set the open-ed stream directly.

Protected Member Functions

- bool Read () override

  To prevent user from calling super class Read() function.

Protected Member Functions inherited from gdcm::ImageReader

- bool ReadACRNEMAImage () override
- bool ReadImage (MediaStorage const &ms) override

Protected Member Functions inherited from gdcm::PixmapReader

- bool ReadImageInternal (MediaStorage const &ms, bool handlepixeldata=true)

Protected Member Functions inherited from gdcm::Reader

- std::istream * GetStreamPtr () const
- bool ReadDataSet ()
- bool ReadMetaInformation ()
- bool ReadPreamble ()

Additional Inherited Members

Protected Attributes inherited from gdcm::PixmapReader

- SmartPointer< Pixmap > PixelData

Protected Attributes inherited from gdcm::Reader

- SmartPointer< File > F

12.156.1 Detailed Description

ImageRegionReader.

This class is able to read a region from a DICOM file containing an image. This implementation requires that the information stored in the DICOM header are consistent with what is in the encapsulated Pixel Data. This is technically not required by DICOM standard, which makes this implementation illegal with regards to the famous JPEG note: `http://dicom.nema.org/medical/dicom/current/output/chtml/part05/sect_8.2.html#para_4bcb841e-c6bf-4e26-82a5-3fad3c942da0`

See also

ImageReader

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, and TemplateEmptyImage.cxx.

12.156.2 Constructor & Destructor Documentation

12.156.2.1 ImageRegionReader()

```
gdcm::ImageRegionReader::ImageRegionReader ()
```

12.156.2.2 ~ImageRegionReader()

```
gdcm::ImageRegionReader::~ImageRegionReader () [override]
```

12.156.3 Member Function Documentation

12.156.3.1 ComputeBufferLength()

```
size_t gdcm::ImageRegionReader::ComputeBufferLength () const
```

Explicit call which will compute the minimal buffer length that can hold the whole uncompressed image as defined by Region `region`.

Returns

0 upon error

12.156.3.2 GetRegion()

```
Region const & gdcm::ImageRegionReader::GetRegion () const
```

12.156.3.3 Read()

```
bool gdcm::ImageRegionReader::Read () [override], [protected], [virtual]
```

To prevent user from calling super class Read() function.

Reimplemented from gdcm::ImageReader.

12.156.3.4 ReadInformation()

```
bool gdcm::ImageRegionReader::ReadInformation ()
```

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, and TemplateEmptyImage.cxx.

12.156.3.5 ReadIntoBuffer()

```
bool gdcm::ImageRegionReader::ReadIntoBuffer (
            char * inreadbuffer,
            size_t buflen)
```

Read into buffer: For Python, the `buflen` param is deduced directly from the input bytearray passed as parameter (function only takes one param).

Returns

false upon error

Examples

ExtractImageRegion.cs, and ExtractImageRegionWithLUT.cs.

12.156.3.6 SetRegion()

```
void gdcm::ImageRegionReader::SetRegion (
            Region const & region)
```

Set/Get Region to be read.

Examples

ExtractImageRegion.cs, and ExtractImageRegionWithLUT.cs.

The documentation for this class was generated from the following file:

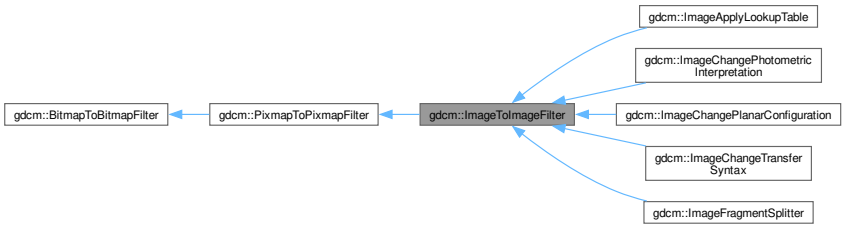
- gdcmImageRegionReader.h

12.157 gdcm::ImageToImageFilter Class Reference

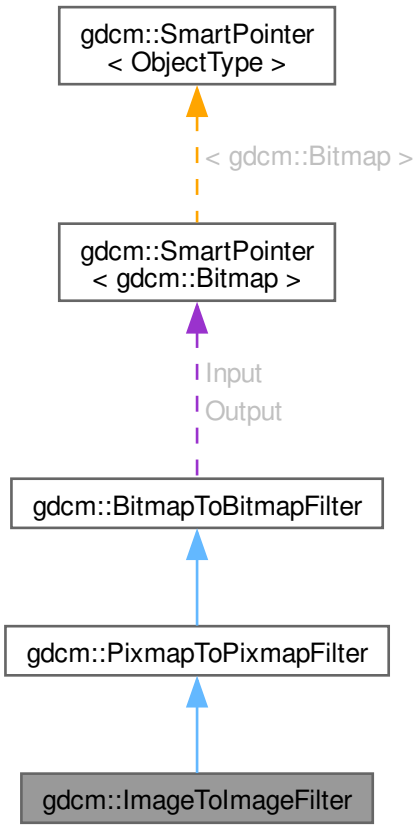
ImageToImageFilter class.

```
#include <gdcmImageToImageFilter.h>
```

Inheritance diagram for gdcm::ImageToImageFilter:



Collaboration diagram for gdcm::ImageToImageFilter:



Public Member Functions

- ImageToImageFilter ()
- ~ImageToImageFilter ()=default
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Public Member Functions inherited from gdcm::PixmapToPixmapFilter

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()=default
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Public Member Functions inherited from gdcm::BitmapToBitmapFilter

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Additional Inherited Members

Protected Attributes inherited from gdcm::BitmapToBitmapFilter

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

12.157.1 Detailed Description

ImageToImageFilter class.

Super class for all filter taking an image and producing an output image

12.157.2 Constructor & Destructor Documentation

12.157.2.1 ImageToImageFilter()

```
gdcm::ImageToImageFilter::ImageToImageFilter ()
```

12.157.2.2 ~ImageToImageFilter()

```
gdcm::ImageToImageFilter::~ImageToImageFilter ()  [default]
```

12.157.3 Member Function Documentation

12.157.3.1 GetInput()

```
Image & gdcm::ImageToImageFilter::GetInput ()
```

12.157.3.2 GetOutput()

```
const Image & gdcm::ImageToImageFilter::GetOutput () const
```

Get Output image.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, and CompressLossyJPEG.cs.

The documentation for this class was generated from the following file:

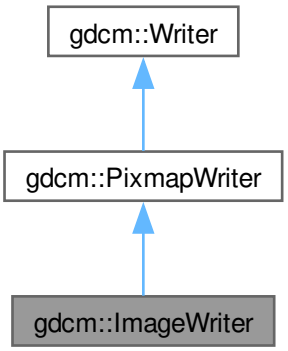
- gdcmImageToImageFilter.h

12.158 gdcm::ImageWriter Class Reference

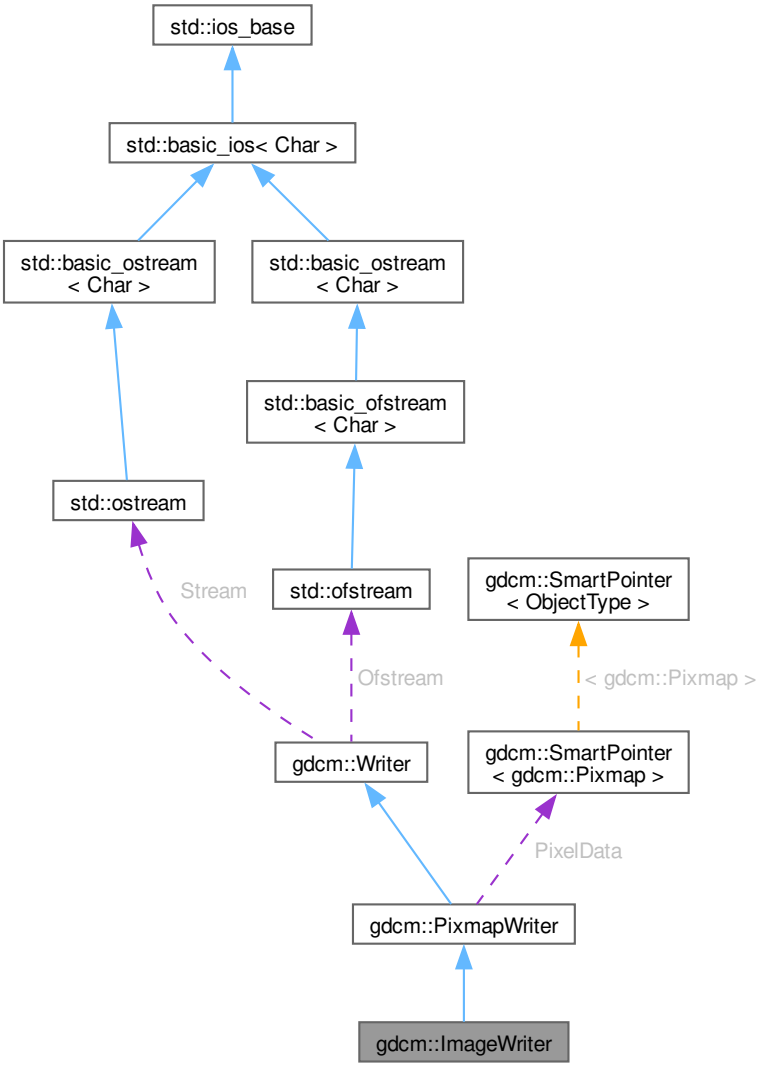
ImageWriter.

```
#include <gdcmImageWriter.h>
```

Inheritance diagram for gdcm::ImageWriter:



Collaboration diagram for gdcm::ImageWriter:



Public Member Functions

- ImageWriter ()
- ~ImageWriter () override
- MediaStorage ComputeTargetMediaStorage ()
- const Image & GetImage () const override
- Image & GetImage () override
- bool Write () override

  Write.

Public Member Functions inherited from gdcm::PixmapWriter

- PixmapWriter ()
- ~PixmapWriter () override
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const
- virtual void SetImage (Pixmap const &img)
- void SetPixmap (Pixmap const &img)
- bool Write () override

  Write.

Public Member Functions inherited from gdcm::Writer

- Writer ()
- virtual ~Writer ()
- void CheckFileMetaInformationOff ()
- void CheckFileMetaInformationOn ()
- File & GetFile ()
- void SetCheckFileMetaInformation (bool b)

  Undocumented function, do not use (= leave default).
- void SetFile (const File &f)

  Set/Get the DICOM file (DataSet + Header).
- void SetFileName (const char *filename_native)

  Set the filename of DICOM file to write:
- void SetStream (std::ostream &output_stream)

  Set user ostream buffer.

Additional Inherited Members

Protected Member Functions inherited from gdcm::PixmapWriter

- void DoIconImage (DataSet &ds, Pixmap const &image)
- bool PrepareWrite (MediaStorage const &refms)

Protected Member Functions inherited from gdcm::Writer

- bool GetCheckFileMetaInformation () const
- std::ostream * GetStreamPtr () const
- void SetWriteDataSetOnly (bool b)

Protected Attributes inherited from gdcm::PixmapWriter

- SmartPointer< Pixmap > PixelData

Protected Attributes inherited from gdcm::Writer

- std::ofstream * Ofstream
- std::ostream * Stream

12.158.1 Detailed Description

ImageWriter.

This is an extended version of the PixmapWriter. Pay attention that:

1. It will populate missing attribute for Secondary Capture Image Storage instances,
2. It may also change an input MR Image Storage instance into a pseudo Enhanced MR Image Storage instance whenever Modality LUT is required.
3. Some DataElement related to gdcm::Image may be slightly altered.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, TemplateEmptyImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

12.158.2 Constructor & Destructor Documentation

12.158.2.1 ImageWriter()

```
gdcm::ImageWriter::ImageWriter ()
```

12.158.2.2 ~ImageWriter()

```
gdcm::ImageWriter::~ImageWriter ()  [override]
```

12.158.3 Member Function Documentation

12.158.3.1 ComputeTargetMediaStorage()

```
MediaStorage gdcm::ImageWriter::ComputeTargetMediaStorage ()
```

internal function used to compute a target MediaStorage the most appropriate User may want to call this function ahead of time (before Write)

Examples

TemplateEmptyImage.cxx.

12.158.3.2 GetImage() [1/2]

```
const Image & gdcm::ImageWriter::GetImage () const  [inline], [override], [virtual]
```

Set/Get Image to be written It will overwrite anything Image infos found in DataSet (see parent class to see how to pass dataset)

Reimplemented from gdcm::PixmapWriter.

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

12.158.3.3 GetImage() [2/2]

```
Image & gdcm::ImageWriter::GetImage ()  [inline], [override], [virtual]
```

Reimplemented from gdcm::PixmapWriter.

12.158.3.4 Write()

```
bool gdcm::ImageWriter::Write ()  [override], [virtual]
```

Write.

Reimplemented from gdcm::Writer.

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, TemplateEmptyImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

The documentation for this class was generated from the following file:

- gdcmImageWriter.h

12.159 gdcm::network::ImplementationClassUIDSub Class Reference

ImplementationClassUIDSub.

```
#include <gdcmImplementationClassUIDSub.h>
```

Public Member Functions

- ImplementationClassUIDSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.159.1 Detailed Description

ImplementationClassUIDSub.

PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

12.159.2 Constructor & Destructor Documentation

12.159.2.1 ImplementationClassUIDSub()

```
gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ()
```

12.159.3 Member Function Documentation

12.159.3.1 Print()

```
void gdcm::network::ImplementationClassUIDSub::Print (
            std::ostream & os) const
```

12.159.3.2 Read()

```
std::istream & gdcm::network::ImplementationClassUIDSub::Read (
            std::istream & is)
```

12.159.3.3 Size()

```
size_t gdcm::network::ImplementationClassUIDSub::Size () const
```

12.159.3.4 Write()

```
const std::ostream & gdcm::network::ImplementationClassUIDSub::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

- gdcmImplementationClassUIDSub.h

12.160 gdcm::network::ImplementationUIDSub Class Reference

ImplementationUIDSub.

```
#include <gdcmImplementationUIDSub.h>
```

Public Member Functions

- ImplementationUIDSub ()
- const std::ostream & Write (std::ostream &os) const

12.160.1 Detailed Description

ImplementationUIDSub.

Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

12.160.2 Constructor & Destructor Documentation

12.160.2.1 ImplementationUIDSub()

```
gdcm::network::ImplementationUIDSub::ImplementationUIDSub ()
```

12.160.3 Member Function Documentation

12.160.3.1 Write()

```
const std::ostream & gdcm::network::ImplementationUIDSub::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

- gdcmImplementationUIDSub.h

12.161 gdcm::network::ImplementationVersionNameSub Class Reference

ImplementationVersionNameSub.

```
#include <gdcmImplementationVersionNameSub.h>
```

Public Member Functions

- ImplementationVersionNameSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.161.1 Detailed Description

ImplementationVersionNameSub.

Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

12.161.2 Constructor & Destructor Documentation

12.161.2.1 ImplementationVersionNameSub()

```
gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ()
```

12.161.3 Member Function Documentation

12.161.3.1 Print()

```
void gdcm::network::ImplementationVersionNameSub::Print (
            std::ostream & os) const
```

12.161.3.2 Read()

```
std::istream & gdcm::network::ImplementationVersionNameSub::Read (
            std::istream & is)
```

12.161.3.3 Size()

```
size_t gdcm::network::ImplementationVersionNameSub::Size () const
```

12.161.3.4 Write()

```
const std::ostream & gdcm::network::ImplementationVersionNameSub::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

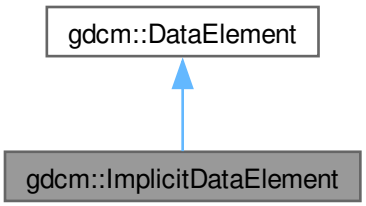
- gdcmImplementationVersionNameSub.h

12.162 gdcm::ImplicitDataElement Class Reference

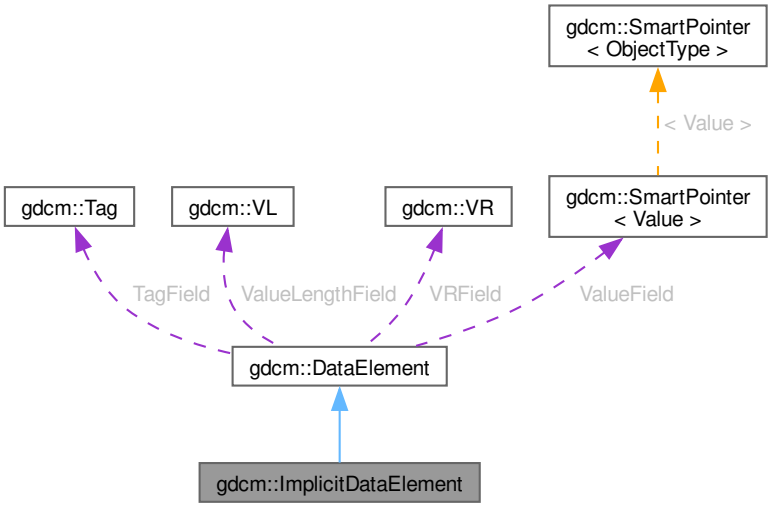
Class to represent an *Implicit VR* Data Element.

```
#include <gdcmImplicitDataElement.h>
```

Inheritance diagram for gdcm::ImplicitDataElement:



Collaboration diagram for gdcm::ImplicitDataElement:



Public Member Functions

- VL GetLength () const

- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, bool readvalues=true)
- template<typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length, bool readvalues=true)
- template<typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)

- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.162.1 Detailed Description

Class to represent an *Implicit VR* Data Element.

Note

bla

Examples

ReadExplicitLengthSQIVR.cxx.

12.162.2 Member Function Documentation

12.162.2.1 GetLength()

```
VL gdcm::ImplicitDataElement::GetLength () const
```

12.162.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::Read (
            std::istream & is)
```

12.162.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::ReadPreValue (
            std::istream & is)
```

12.162.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::ReadValue (
            std::istream & is,
            bool readvalues = true)
```

12.162.2.5 ReadValueWithLength()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::ReadValueWithLength (
            std::istream & is,
            VL & length,
            bool readvalues = true)
```

12.162.2.6 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::ImplicitDataElement::ReadWithLength (
            std::istream & is,
            VL & length,
            bool readvalues = true)
```

12.162.2.7 Write()

```
template<typename TSwap>
const std::ostream & gdcm::ImplicitDataElement::Write (
            std::ostream & os) const
```

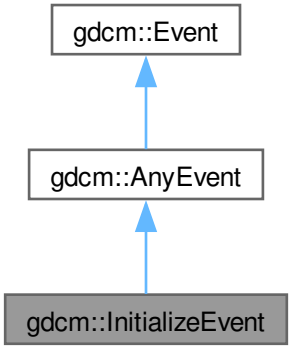
The documentation for this class was generated from the following file:

- gdcmImplicitDataElement.h

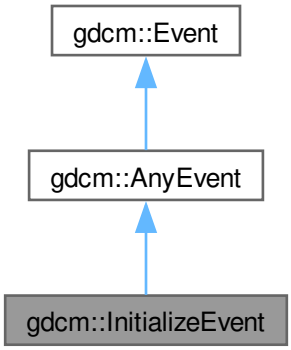
12.163 gdcm::InitializeEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::InitializeEvent:



Collaboration diagram for gdcm::InitializeEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

- gdcmEvent.h

12.164 gdcm::IOD Class Reference

Class for representing a IOD.

```
#include <gdcmIOD.h>
```

Public Types

- typedef std::vector< IODEntry > MapIODEntry
- typedef MapIODEntry::size_type SizeType

Public Member Functions

- IOD ()=default
- void AddIODEntry (const IODEntry &iode)
- void Clear ()
- const IODEntry & GetIODEntry (SizeType idx) const
- SizeType GetNumberOfIODs () const
- Type GetTypeFromTag (const Defs &defs, const Tag &tag) const

Friends

- std::ostream & operator<< (std::ostream &_os, const IOD &_val)

12.164.1 Detailed Description

Class for representing a IOD.

Note

bla

See also

Dict

Examples

TraverseModules.cxx.

12.164.2 Member Typedef Documentation

12.164.2.1 MapIODEntry

```
typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry
```

12.164.2.2 SizeType

```
typedef MapIODEntry::size_type gdcm::IOD::SizeType
```

12.164.3 Constructor & Destructor Documentation

12.164.3.1 IOD()

```
gdcm::IOD::IOD () [default]
```

References IOD(), and operator<<.

Referenced by IOD(), and operator<<.

12.164.4 Member Function Documentation

12.164.4.1 AddIODEntry()

```
void gdcm::IOD::AddIODEntry (
            const IODEntry & iode) [inline]
```

12.164.4.2 Clear()

```
void gdcm::IOD::Clear () [inline]
```

12.164.4.3 GetIODEntry()

```
const IODEntry & gdcm::IOD::GetIODEntry (
            SizeType idx) const [inline]
```

Examples

TraverseModules.cxx.

12.164.4.4 GetNumberOfIODs()

```
SizeType gdcm::IOD::GetNumberOfIODs () const [inline]
```

Examples

TraverseModules.cxx.

12.164.4.5 GetTypeFromTag()

```
Type gdcm::IOD::GetTypeFromTag (
            const Defs & defs,
            const Tag & tag) const
```

12.164.5 Friends And Related Symbol Documentation

12.164.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const IOD & _val)  [friend]
```

References IOD().

Referenced by IOD().

The documentation for this class was generated from the following file:

- gdcmIOD.h

12.165 gdcm::IODEntry Class Reference

Class for representing a IODEntry.

```
#include <gdcmIODEntry.h>
```

Public Member Functions

- IODEntry (const char *name="", const char *ref="", const char *inUsage="")
- const char * GetIE () const
- const char * GetName () const
- const char * GetRef () const
- const char * GetUsage () const
- Usage::UsageType GetUsageType () const
- void SetIE (const char *ie)
- void SetName (const char *name)
- void SetRef (const char *ref)
- void SetUsage (const char *inUsage)

Friends

- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)

12.165.1 Detailed Description

Class for representing a IODEntry.

Note

A.1.3 IOD Module Table and Functional Group Macro Table This Section of each IOD defines in a tabular form the Modules comprising the IOD. The following information must be specified for each Module in the table:

- The name of the Module or Functional Group
- A reference to the Section in Annex C which defines the Module or Functional Group
- The usage of the Module or Functional Group; whether it is:
- Mandatory (see A.1.3.1) , abbreviated M
- Conditional (see A.1.3.2) , abbreviated C
- User Option (see A.1.3.3) , abbreviated U The Modules referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each IOD, Mandatory Modules shall be supported per the definitions, semantics and requirements defined in Annex C. PS 3.3 - 2008 Page 96
- Standard - A.1.3.2 CONDITIONAL MODULES Conditional Modules are Mandatory Modules if specific conditions are met. If the specified conditions are not met, this Module shall not be supported; that is, no information defined in that Module shall be sent. A.1.3.3 USER OPTION MODULES User Option Modules may or may not be supported. If an optional Module is supported, the Attribute Types specified in the Modules in Annex C shall be supported.

See also

DictEntry

Examples

TraverseModules.cxx.

12.165.2 Constructor & Destructor Documentation

12.165.2.1 IODEntry()

```
gdcm::IODEntry::IODEntry (
            const char * name = "",
            const char * ref = "",
            const char * inUsage = "")  [inline]
```

Referenced by operator<<.

12.165.3 Member Function Documentation

12.165.3.1 GetIE()

```
const char * gdcm::IODEntry::GetIE () const  [inline]
```

12.165.3.2 GetName()

```
const char * gdcm::IODEntry::GetName () const  [inline]
```

12.165.3.3 GetRef()

```
const char * gdcm::IODEntry::GetRef () const  [inline]
```

Examples

TraverseModules.cxx.

12.165.3.4 GetUsage()

```
const char * gdcm::IODEntry::GetUsage () const  [inline]
```

12.165.3.5 GetUsageType()

```
Usage::UsageType gdcm::IODEntry::GetUsageType () const
```

12.165.3.6 SetIE()

```
void gdcm::IODEntry::SetIE (
            const char * ie)  [inline]
```

12.165.3.7 SetName()

```
void gdcm::IODEntry::SetName (
            const char * name)  [inline]
```

12.165.3.8 SetRef()

```
void gdcm::IODEntry::SetRef (
            const char * ref)  [inline]
```

12.165.3.9 SetUsage()

```
void gdcm::IODEntry::SetUsage (
            const char * inUsage)  [inline]
```

12.165.4 Friends And Related Symbol Documentation

12.165.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const IODEntry & _val)  [friend]
```

References IODEntry().

The documentation for this class was generated from the following file:

- gdcmIODEntry.h

12.166 gdcm::IODs Class Reference

Class for representing a IODs.

```
#include <gdcmIODs.h>
```

Public Types

- typedef std::map< IODName, IOD > IODMapType
- typedef IODMapType::const_iterator IODMapTypeConstIterator
- typedef std::string IODName

Public Member Functions

- IODs ()=default
- void AddIOD (const char *name, const IOD &module)
- IODMapTypeConstIterator Begin () const
- void Clear ()
- IODMapTypeConstIterator End () const
- const IOD & GetIOD (const char *name) const

Friends

- std::ostream & operator<< (std::ostream &_os, const IODs &_val)

12.166.1 Detailed Description

Class for representing a IODs.

Note

bla

See also

IOD

Examples

TraverseModules.cxx.

12.166.2 Member Typedef Documentation

12.166.2.1 IODMapType

```
typedef std::map<IODName, IOD> gdcm::IODs::IODMapType
```

12.166.2.2 IODMapTypeConstIterator

```
typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator
```

Examples

TraverseModules.cxx.

12.166.2.3 IODName

```
typedef std::string gdcm::IODs::IODName
```

Examples

TraverseModules.cxx.

12.166.3 Constructor & Destructor Documentation

12.166.3.1 IODs()

```
gdcm::IODs::IODs () [default]
```

References IODs(), and operator<<.

Referenced by IODs(), and operator<<.

12.166.4 Member Function Documentation

12.166.4.1 AddIOD()

```
void gdcm::IODs::AddIOD (
            const char * name,
            const IOD & module)  [inline]
```

12.166.4.2 Begin()

```
IODMapTypeConstIterator gdcm::IODs::Begin () const  [inline]
```

Examples

TraverseModules.cxx.

12.166.4.3 Clear()

```
void gdcm::IODs::Clear ()  [inline]
```

12.166.4.4 End()

```
IODMapTypeConstIterator gdcm::IODs::End () const  [inline]
```

Examples

TraverseModules.cxx.

12.166.4.5 GetIOD()

```
const IOD & gdcm::IODs::GetIOD (
            const char * name) const  [inline]
```

12.166.5 Friends And Related Symbol Documentation

12.166.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const IODs & _val)  [friend]
```

References IODs().

Referenced by IODs().

The documentation for this class was generated from the following file:

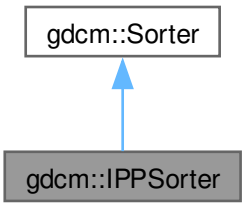
- gdcmIODs.h

12.167 gdcm::IPPSorter Class Reference

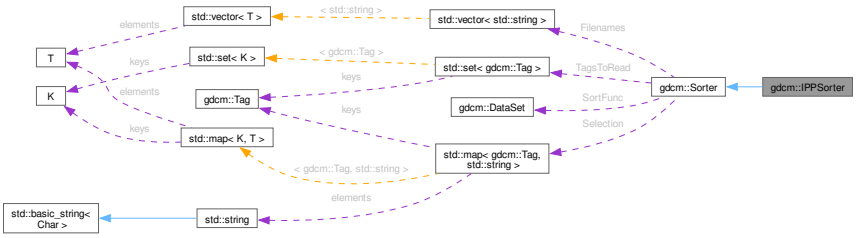
IPPSorter.

```
#include <gdcmIPPSorter.h>
```

Inheritance diagram for gdcm::IPPSorter:



Collaboration diagram for gdcm::IPPSorter:



Public Member Functions

- IPPSorter ()
- double GetDirectionCosinesTolerance () const
- double GetZSpacing () const
- double GetZSpacingTolerance () const
- void SetComputeZSpacing (bool b)
- void SetDirectionCosinesTolerance (double tol)
- void SetDropDuplicatePositions (bool b)
- void SetZSpacingTolerance (double tol)
- bool Sort (std::vector< std::string > const &filenames) override

Public Member Functions inherited from gdcm::Sorter

- Sorter ()
- virtual ~Sorter ()
- bool AddSelect (Tag const &tag, const char *value)

  UNSUPPORTED FOR NOW.
- const std::vector< std::string > & GetFilenames () const
- void Print (std::ostream &os) const

  Print.
- void SetSortFunction (SortFunction f)
- void SetTagsToRead (std::set< Tag > const &tags)
- virtual bool StableSort (std::vector< std::string > const &filenames)

Protected Attributes

- bool ComputeZSpacing
- double DirCosTolerance
- bool DropDuplicatePositions
- double ZSpacing
- double ZTolerance

Protected Attributes inherited from gdcm::Sorter

- std::vector< std::string > Filenames
- std::map< Tag, std::string > Selection
- SortFunction SortFunc
- std::set< Tag > TagsToRead

Additional Inherited Members

Public Types inherited from gdcm::Sorter

- typedef bool(* SortFunction) (DataSet const &, DataSet const &)

  Set the sort function which compares one dataset to the other.

Protected Types inherited from gdcm::Sorter

- typedef std::map< Tag, std::string > SelectionMap

12.167.1 Detailed Description

IPPSorter.

Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

Warning

See special note for SetZSpacingTolerance when computing the ZSpacing from the IPP of each DICOM files (default tolerance for consistent spacing is: 1e-6mm)

For more information on Spacing, and how it is defined in DICOM, advanced users may refers to:

`http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing`

Bug There are currently a couple of bugs in this implementation:

- Gantry Tilt is not considered (always an error)
- Application programmer should only sort valid DataSet (eg. MRImageStorage, CTImageStorage, PETImage↩ Storage)

Examples

Compute3DSpacing.cxx, VolumeSorter.cxx, and gdcmorthoplanes.cxx.

12.167.2 Constructor & Destructor Documentation

12.167.2.1 IPPSorter()

```
gdcm::IPPSorter::IPPSorter ()
```

12.167.3 Member Function Documentation

12.167.3.1 GetDirectionCosinesTolerance()

```
double gdcm::IPPSorter::GetDirectionCosinesTolerance () const  [inline]
```

References DirCosTolerance.

12.167.3.2 GetZSpacing()

```
double gdcm::IPPSorter::GetZSpacing () const  [inline]
```

Read-only function to provide access to the computed value for the Z-Spacing The ComputeZSpacing must have been set to true before execution of sort algorithm. Call this function *after* calling Sort(); Z-Spacing will be 0 on 2 occasions:

- Sorting simply failed, potentially duplicate IPP => ZSpacing = 0
- ZSpacing could not be computed (Z-Spacing is not constant, or ZTolerance is too low)

Examples

Compute3DSpacing.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

References ZSpacing.

12.167.3.3 GetZSpacingTolerance()

```
double gdcm::IPPSorter::GetZSpacingTolerance () const  [inline]
```

References ZTolerance.

12.167.3.4 SetComputeZSpacing()

```
void gdcm::IPPSorter::SetComputeZSpacing (
            bool b)  [inline]
```

Functions related to Z-Spacing computation Set to true when sort algorithm should also perform a regular Z-Spacing computation using the Image Position (Patient) Potential reason for failure:

1. ALL slices are taken into account, if one slice if missing then ZSpacing will be set to 0 since the spacing will not be found to be regular along the Series

Examples

Compute3DSpacing.cxx, VolumeSorter.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

References ComputeZSpacing.

12.167.3.5 SetDirectionCosinesTolerance()

```
void gdcm::IPPSorter::SetDirectionCosinesTolerance (
            double tol) [inline]
```

Sometimes IOP along a series is slightly changing for example: "0.999081\\0.0426953\\0.00369272\\-0.0419025\\0.↩955059\\0.293439", "0.999081\\0.0426953\\0.00369275\\-0.0419025\\0.955059\\0.293439", "0.999081\\0.0426952\\0.↩00369272\\-0.0419025\\0.955059\\0.293439", We need an API to define the tolerance which is allowed. Internally the cross vector of each direction cosines is computed. The tolerance then define the distance in between 1.0 to the dot product of those cross vectors. In a perfect world this dot product is of course 1.0 which imply a DirectionCosines tolerance of exactly 0.0 (default).

References DirCosTolerance.

12.167.3.6 SetDropDuplicatePositions()

```
void gdcm::IPPSorter::SetDropDuplicatePositions (
            bool b) [inline]
```

Makes the IPPSorter ignore multiple images located at the same position. Only the first occurrence will be kept. Drop↩DuplicatePositions defaults to false.

References DropDuplicatePositions.

12.167.3.7 SetZSpacingTolerance()

```
void gdcm::IPPSorter::SetZSpacingTolerance (
            double tol) [inline]
```

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the series, a human can determine that this is ok and change the tolerance from its default value: 1e-6

Examples

Compute3DSpacing.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

References ZTolerance.

12.167.3.8 Sort()

```
bool gdcm::IPPSorter::Sort (
            std::vector< std::string > const & filenames) [override], [virtual]
```

Main entry point to the sorter. It will execute the filter, option should be set before running this function (SetZSpacing↩Tolerance, ...) Return value indicate if sorting could be achieved,. Warning this does *NOT* imply that spacing is consistent, it only means the file are sorted according to IPP You should check if ZSpacing is 0 or not to deduce if file are actually a 3D volume

Reimplemented from gdcm::Sorter.

Examples

Compute3DSpacing.cxx, VolumeSorter.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

12.167.4 Member Data Documentation

12.167.4.1 ComputeZSpacing

```
bool gdcm::IPPSorter::ComputeZSpacing [protected]
```

Referenced by SetComputeZSpacing().

12.167.4.2 DirCosTolerance

```
double gdcm::IPPSorter::DirCosTolerance [protected]
```

Referenced by GetDirectionCosinesTolerance(), and SetDirectionCosinesTolerance().

12.167.4.3 DropDuplicatePositions

```
bool gdcm::IPPSorter::DropDuplicatePositions [protected]
```

Referenced by SetDropDuplicatePositions().

12.167.4.4 ZSpacing

```
double gdcm::IPPSorter::ZSpacing [protected]
```

Referenced by GetZSpacing().

12.167.4.5 ZTolerance

```
double gdcm::IPPSorter::ZTolerance [protected]
```

Referenced by GetZSpacingTolerance(), and SetZSpacingTolerance().

The documentation for this class was generated from the following file:

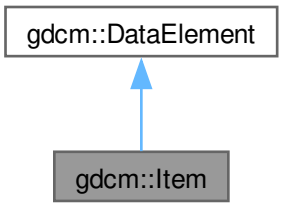
- gdcmIPPSorter.h

12.168 gdcm::Item Class Reference

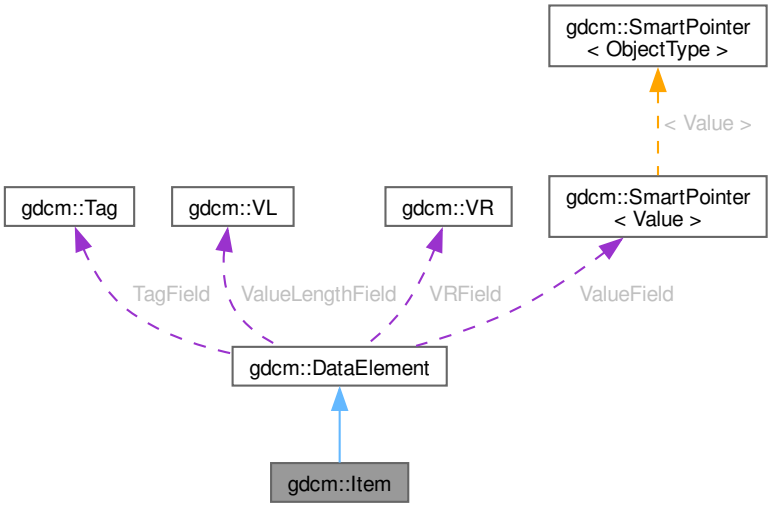
Class to represent an Item.

```
#include <gdcmItem.h>
```

Inheritance diagram for gdcm::Item:



Collaboration diagram for gdcm::Item:



Public Member Functions

- Item ()

- Item (Item const &val)
- void Clear ()
- bool FindDataElement (const Tag &t) const
- const DataElement & GetDataElement (const Tag &t) const
- template<typename TDE>
  VL GetLength () const
- DataSet & GetNestedDataSet ()
- const DataSet & GetNestedDataSet () const
- void InsertDataElement (const DataElement &de)
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- void SetNestedDataSet (const DataSet &nested)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)

- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &os, const Item &val)

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.168.1 Detailed Description

Class to represent an Item.

A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standard Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit

Note

ITEM: A component of the Value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set.

Examples

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, NewSequence.cs, SimplePrint.cs, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.168.2 Constructor & Destructor Documentation

12.168.2.1 Item() [1/2]

```
gdcm::Item::Item () [inline]
```

References gdcm::DataElement::DataElement().

Referenced by Item(), and operator<<.

12.168.2.2 Item() [2/2]

```
gdcm::Item::Item (
            Item const & val) [inline]
```

References gdcm::DataElement::DataElement(), and Item().

12.168.3 Member Function Documentation

12.168.3.1 Clear()

```
void gdcm::Item::Clear () [inline]
```

References gdcm::DataElement::Clear().

Referenced by gdcm::SequenceOfItems::Read().

12.168.3.2 FindDataElement()

```
bool gdcm::Item::FindDataElement (
            const Tag & t) const  [inline]
```

12.168.3.3 GetDataElement()

```
const DataElement & gdcm::Item::GetDataElement (
            const Tag & t) const  [inline]
```

References gdcm::DataElement::DataElement().

12.168.3.4 GetLength()

```
template<typename TDE>
VL gdcm::Item::GetLength () const
```

12.168.3.5 GetNestedDataSet() [1/2]

```
DataSet & gdcm::Item::GetNestedDataSet ()  [inline]
```

12.168.3.6 GetNestedDataSet() [2/2]

```
const DataSet & gdcm::Item::GetNestedDataSet () const  [inline]
```

Examples

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenSeqs.cxx, GetSequenceUltrasound.cx GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, NewSequence.cs, SimplePrint.cs, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

Referenced by gdcm::SequenceOfItems::Read().

12.168.3.7 InsertDataElement()

```
void gdcm::Item::InsertDataElement (
            const DataElement & de)  [inline]
```

References gdcm::DataElement::DataElement(), and gdcm::DataElement::IsUndefinedLength().

12.168.3.8 Read()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::Item::Read (
            std::istream & is)  [inline]
```

References gdcm::ByteSwapFilter::ByteSwap(), gdcm::DataSet::Clear(), gdcmDebugMacro, gdcmErrorMacro, gdcmWarningMacro, gdcm::DataSet::IsEmpty(), gdcm::DataElement::ReadWithLength(), gdcm::ByteSwapFilter::SetByteSwapTag(), gdcm::SwapperDoOp::Swap(), gdcm::DataElement::TagField, and gdcm::DataElement::ValueLengthField.

Referenced by gdcm::SequenceOfItems::Read().

12.168.3.9 SetNestedDataSet()

```
void gdcm::Item::SetNestedDataSet (
            const DataSet & nested)  [inline]
```

12.168.3.10 Write()

```
template<typename TDE, typename TSwap>
const std::ostream & gdcm::Item::Write (
            std::ostream & os) const  [inline]
```

References gdcmWarningMacro, gdcm::VL::GetLength(), gdcm::DataElement::TagField, gdcm::DataElement::ValueLengthField, gdcm::Tag::Write(), and gdcm::VL::Write().

12.168.4 Friends And Related Symbol Documentation

12.168.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Item & val)  [friend]
```

References Item(), operator<<, gdcm::DataSet::Print(), gdcm::DataElement::TagField, and gdcm::DataElement::ValueLengthField.

Referenced by operator<<.

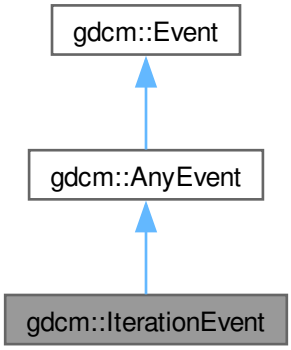
The documentation for this class was generated from the following file:

- gdcmItem.h

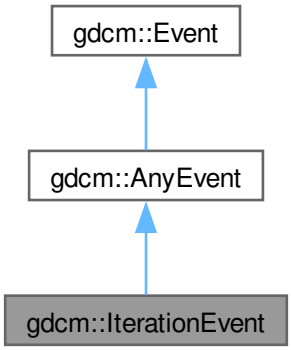
12.169 gdcm::IterationEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()

- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

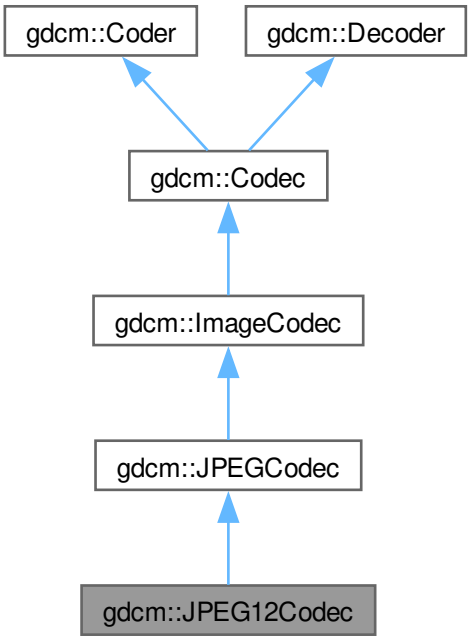
- gdcmEvent.h

12.170 gdcm::JPEG12Codec Class Reference

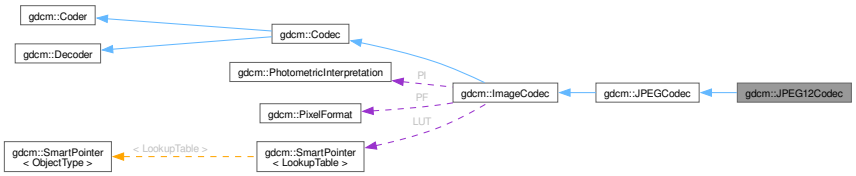
Class to do JPEG 12bits (lossy & lossless).

```
#include <gdcmJPEG12Codec.h>
```

Inheritance diagram for gdcm::JPEG12Codec:



Collaboration diagram for gdcm::JPEG12Codec:



Public Member Functions

- JPEG12Codec ()
- ~JPEG12Codec () override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool InternalCode (const char *input, unsigned long len, std::ostream &os) override

Public Member Functions inherited from gdcm::JPEGCodec

- JPEGCodec ()
- ~JPEGCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Compress into JPEG.
- void ComputeOffsetTable (bool b)

  Compute the offset table:
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool GetLossless () const
- double GetQuality () const
- void SetLossless (bool l)
- void SetPixelFormat (PixelFormat const &pf) override
- void SetQuality (double q)

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen) override
- bool IsStateSuspension () const override

Protected Member Functions inherited from gdcm::JPEGCodec

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool IsValid (PhotometricInterpretation const &pi) override
- void SetBitSample (int bit)
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Protected Member Functions inherited from gdcm::ImageCodec

- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Attributes inherited from gdcm::JPEGCodec

- int BitSample
- int Quality

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.170.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless).

Note

internal class

12.170.2 Constructor & Destructor Documentation

12.170.2.1 JPEG12Codec()

```
gdcm::JPEG12Codec::JPEG12Codec ()
```

12.170.2.2 ~JPEG12Codec()

```
gdcm::JPEG12Codec::~JPEG12Codec () [override]
```

12.170.3 Member Function Documentation

12.170.3.1 DecodeByStreams()

```
bool gdcm::JPEG12Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os) [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.170.3.2 EncodeBuffer()

```
bool gdcm::JPEG12Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen) [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

12.170.3.3 GetHeaderInfo()

```
bool gdcm::JPEG12Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts) [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.170.3.4 InternalCode()

```
bool gdcm::JPEG12Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os) [override], [virtual]
```

Reimplemented from gdcm::Coder.

12.170.3.5 IsStateSuspension()

```
bool gdcm::JPEG12Codec::IsStateSuspension () const  [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

The documentation for this class was generated from the following file:

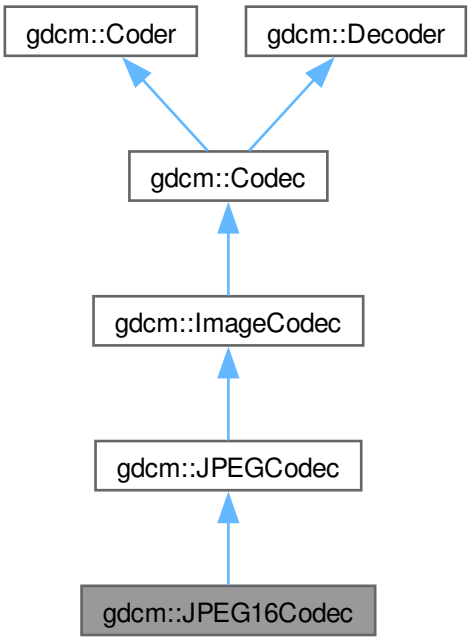
- gdcmJPEG12Codec.h

12.171 gdcm::JPEG16Codec Class Reference

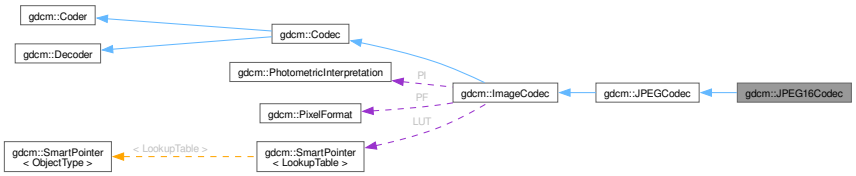
Class to do JPEG 16bits (lossless).

```
#include <gdcmJPEG16Codec.h>
```

Inheritance diagram for gdcm::JPEG16Codec:



Collaboration diagram for gdcm::JPEG16Codec:



Public Member Functions

- JPEG16Codec ()
- ~JPEG16Codec () override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool InternalCode (const char *input, unsigned long len, std::ostream &os) override

Public Member Functions inherited from gdcm::JPEGCodec

- JPEGCodec ()
- ~JPEGCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Compress into JPEG.
- void ComputeOffsetTable (bool b)

  Compute the offset table:
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool GetLossless () const
- double GetQuality () const
- void SetLossless (bool l)
- void SetPixelFormat (PixelFormat const &pf) override
- void SetQuality (double q)

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen) override
- bool IsStateSuspension () const override

Protected Member Functions inherited from gdcm::JPEGCodec

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool IsValid (PhotometricInterpretation const &pi) override
- void SetBitSample (int bit)
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Protected Member Functions inherited from gdcm::ImageCodec

- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Attributes inherited from gdcm::JPEGCodec

- int BitSample
- int Quality

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.171.1 Detailed Description

Class to do JPEG 16bits (lossless).

Note

internal class

12.171.2 Constructor & Destructor Documentation

12.171.2.1 JPEG16Codec()

```
gdcm::JPEG16Codec::JPEG16Codec ()
```

12.171.2.2 ~JPEG16Codec()

```
gdcm::JPEG16Codec::~JPEG16Codec () [override]
```

12.171.3 Member Function Documentation

12.171.3.1 DecodeByStreams()

```
bool gdcm::JPEG16Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os) [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.171.3.2 EncodeBuffer()

```
bool gdcm::JPEG16Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen) [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

12.171.3.3 GetHeaderInfo()

```
bool gdcm::JPEG16Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts) [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.171.3.4 InternalCode()

```
bool gdcm::JPEG16Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os) [override], [virtual]
```

Reimplemented from gdcm::Coder.

12.171.3.5 IsStateSuspension()

```
bool gdcm::JPEG16Codec::IsStateSuspension () const  [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

The documentation for this class was generated from the following file:

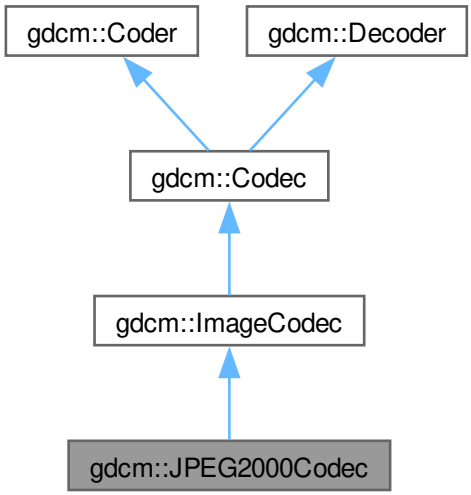
- gdcmJPEG16Codec.h

12.172 gdcm::JPEG2000Codec Class Reference

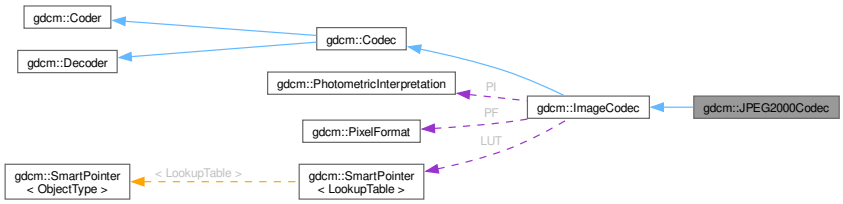
Class to do JPEG 2000.

```
#include <gdcmJPEG2000Codec.h>
```

Inheritance diagram for gdcm::JPEG2000Codec:



Collaboration diagram for gdcm::JPEG2000Codec:



Public Member Functions

- JPEG2000Codec ()
- ~JPEG2000Codec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- double GetQuality (unsigned int idx=0) const
- double GetRate (unsigned int idx=0) const
- void SetMCT (unsigned int mct)
- void SetNumberOfResolutions (unsigned int nres)
- void SetNumberOfThreadsForDecompression (int nThreads)
- void SetQuality (unsigned int idx, double q)
- void SetRate (unsigned int idx, double rate)
- void SetReversible (bool res)
- void SetTileSize (unsigned int tx, unsigned int ty)

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Protected Member Functions inherited from gdcm::ImageCodec

- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsValid (PhotometricInterpretation const &pi)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Friends

- class Bitmap
- class ImageRegionReader

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.172.1 Detailed Description

Class to do JPEG 2000.

Note

the class will produce JPC (JPEG 2000 codestream), since some private implementor are using full jp2 file the decoder tolerate jp2 input this is an implementation of an ImageCodec

12.172.2 Constructor & Destructor Documentation

12.172.2.1 JPEG2000Codec()

```
gdcm::JPEG2000Codec::JPEG2000Codec ()
```

12.172.2.2 ~JPEG2000Codec()

```
gdcm::JPEG2000Codec::~JPEG2000Codec ()  [override]
```

12.172.3 Member Function Documentation

12.172.3.1 AppendFrameEncode()

```
bool gdcm::JPEG2000Codec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.172.3.2 AppendRowEncode()

```
bool gdcm::JPEG2000Codec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.172.3.3 CanCode()

```
bool gdcm::JPEG2000Codec::CanCode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

12.172.3.4 CanDecode()

```
bool gdcm::JPEG2000Codec::CanDecode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.172.3.5 Clone()

```
ImageCodec * gdcm::JPEG2000Codec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.172.3.6 Code()

```
bool gdcm::JPEG2000Codec::Code (
            DataElement const & in_,
            DataElement & out_)  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

12.172.3.7 Decode()

```
bool gdcm::JPEG2000Codec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

12.172.3.8 DecodeByStreams()

```
bool gdcm::JPEG2000Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.172.3.9 DecodeExtent()

```
bool gdcm::JPEG2000Codec::DecodeExtent (
            char * buffer,
            unsigned int xmin,
            unsigned int xmax,
            unsigned int ymin,
            unsigned int ymax,
            unsigned int zmin,
            unsigned int zmax,
            std::istream & is)  [protected]
```

12.172.3.10 GetHeaderInfo()

```
bool gdcm::JPEG2000Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

Referenced by StopEncode().

12.172.3.11 GetQuality()

```
double gdcm::JPEG2000Codec::GetQuality (
            unsigned int idx = 0) const
```

12.172.3.12 GetRate()

```
double gdcm::JPEG2000Codec::GetRate (
            unsigned int idx = 0) const
```

12.172.3.13 IsFrameEncoder()

```
bool gdcm::JPEG2000Codec::IsFrameEncoder () [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.172.3.14 IsRowEncoder()

```
bool gdcm::JPEG2000Codec::IsRowEncoder () [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.172.3.15 SetMCT()

```
void gdcm::JPEG2000Codec::SetMCT (
            unsigned int mct)
```

12.172.3.16 SetNumberOfResolutions()

```
void gdcm::JPEG2000Codec::SetNumberOfResolutions (
            unsigned int nres)
```

12.172.3.17 SetNumberOfThreadsForDecompression()

```
void gdcm::JPEG2000Codec::SetNumberOfThreadsForDecompression (
            int nThreads)
```

Set Number of threads

Parameters

nThreads	: number of threads for decompression codec, if 0 or 1 decompression is done in current thread, if negative value is set determine how many virtual threads are available

12.172.3.18 SetQuality()

```
void gdcm::JPEG2000Codec::SetQuality (
            unsigned int idx,
            double q)
```

12.172.3.19 SetRate()

```
void gdcm::JPEG2000Codec::SetRate (
            unsigned int idx,
            double rate)
```

12.172.3.20 SetReversible()

```
void gdcm::JPEG2000Codec::SetReversible (
            bool res)
```

12.172.3.21 SetTileSize()

```
void gdcm::JPEG2000Codec::SetTileSize (
            unsigned int tx,
            unsigned int ty)
```

12.172.3.22 StartEncode()

```
bool gdcm::JPEG2000Codec::StartEncode (
            std::ostream & )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.172.3.23 StopEncode()

```
bool gdcm::JPEG2000Codec::StopEncode (
            std::ostream & )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

References GetHeaderInfo().

12.172.4 Friends And Related Symbol Documentation

12.172.4.1 Bitmap

```
friend class Bitmap [friend]
```

References Bitmap.

Referenced by Bitmap.

12.172.4.2 ImageRegionReader

```
friend class ImageRegionReader [friend]
```

References ImageRegionReader.

Referenced by ImageRegionReader.

The documentation for this class was generated from the following file:

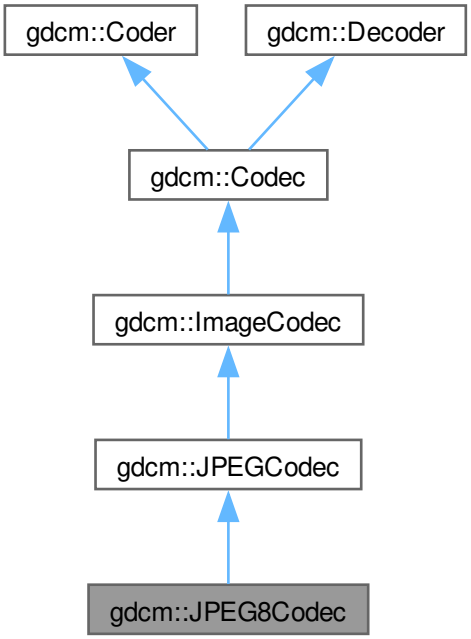
- gdcmJPEG2000Codec.h

12.173 gdcm::JPEG8Codec Class Reference

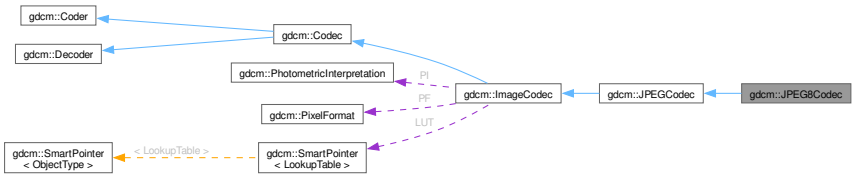
Class to do JPEG 8bits (lossy & lossless).

```
#include <gdcmJPEG8Codec.h>
```

Inheritance diagram for gdcm::JPEG8Codec:



Collaboration diagram for gdcm::JPEG8Codec:



Public Member Functions

- JPEG8Codec ()
- ~JPEG8Codec () override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool InternalCode (const char *input, unsigned long len, std::ostream &os) override

Public Member Functions inherited from gdcm::JPEGCodec

- JPEGCodec ()
- ~JPEGCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Compress into JPEG.
- void ComputeOffsetTable (bool b)

  Compute the offset table:
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool GetLossless () const
- double GetQuality () const
- void SetLossless (bool l)
- void SetPixelFormat (PixelFormat const &pf) override
- void SetQuality (double q)

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen) override
- bool IsStateSuspension () const override

Protected Member Functions inherited from gdcm::JPEGCodec

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool IsValid (PhotometricInterpretation const &pi) override
- void SetBitSample (int bit)
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Protected Member Functions inherited from gdcm::ImageCodec

- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Attributes inherited from gdcm::JPEGCodec

- int BitSample
- int Quality

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.173.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless).

Note

internal class

12.173.2 Constructor & Destructor Documentation

12.173.2.1 JPEG8Codec()

```
gdcm::JPEG8Codec::JPEG8Codec ()
```

12.173.2.2 ~JPEG8Codec()

```
gdcm::JPEG8Codec::~JPEG8Codec () [override]
```

12.173.3 Member Function Documentation

12.173.3.1 DecodeByStreams()

```
bool gdcm::JPEG8Codec::DecodeByStreams (
            std::istream & is,
            std::ostream & os) [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.173.3.2 EncodeBuffer()

```
bool gdcm::JPEG8Codec::EncodeBuffer (
            std::ostream & os,
            const char * data,
            size_t datalen) [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

12.173.3.3 GetHeaderInfo()

```
bool gdcm::JPEG8Codec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts) [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.173.3.4 InternalCode()

```
bool gdcm::JPEG8Codec::InternalCode (
            const char * input,
            unsigned long len,
            std::ostream & os) [override], [virtual]
```

Reimplemented from gdcm::Coder.

12.173.3.5 IsStateSuspension()

```
bool gdcm::JPEG8Codec::IsStateSuspension () const [override], [protected], [virtual]
```

Reimplemented from gdcm::JPEGCodec.

The documentation for this class was generated from the following file:

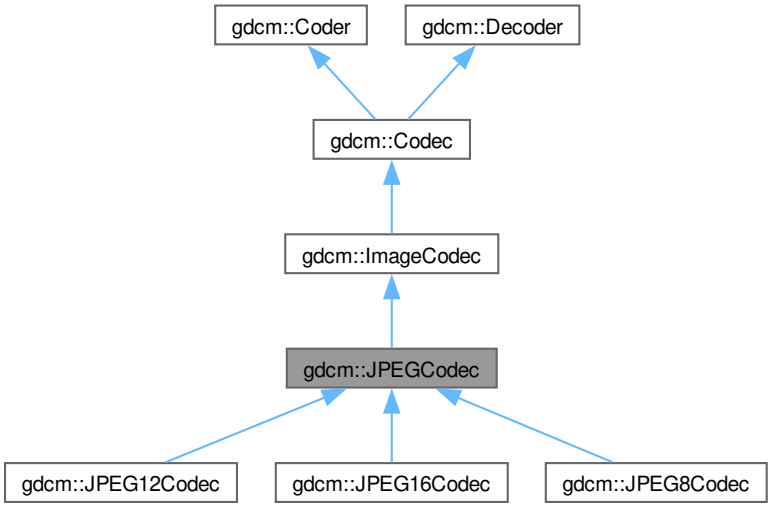
- gdcmJPEG8Codec.h

12.174 gdcm::JPEGCodec Class Reference

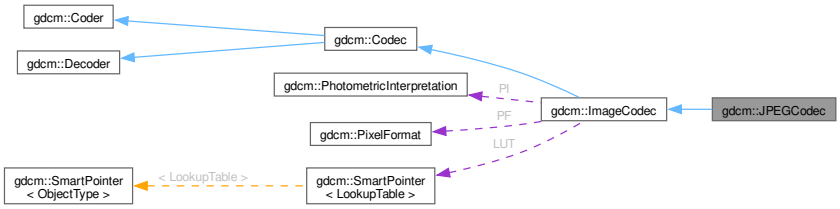
JPEG codec.

```
#include <gdcmJPEGCodec.h>
```

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



Public Member Functions

- JPEGCodec ()
- ~JPEGCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Compress into JPEG.
- void ComputeOffsetTable (bool b)

  Compute the offset table:
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- virtual bool EncodeBuffer (std::ostream &out, const char *inbuffer, size_t inlen)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool GetLossless () const
- double GetQuality () const
- void SetLossless (bool l)
- void SetPixelFormat (PixelFormat const &pf) override
- void SetQuality (double q)

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- virtual bool IsStateSuspension () const
- bool IsValid (PhotometricInterpretation const &pi) override
- void SetBitSample (int bit)
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Protected Member Functions inherited from gdcm::ImageCodec

- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes

- int BitSample
- int Quality

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

Friends

- class ImageRegionReader

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

12.174.1 Detailed Description

JPEG codec.

Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: JPEG8Codec, JPEG12Codec & JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case

Note

Things you should know if you ever want to dive into DICOM/JPEG world (among other):

- `http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/625e46919f208`
- `http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/75fdfccc65a62`
- `http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f09`
- `http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c9`

Examples

CompressLossyJPEG.cs, FileChangeTSLossy.cs, and GetJPEGSamplePrecision.cxx.

12.174.2 Constructor & Destructor Documentation

12.174.2.1 JPEGCodec()

```
gdcm::JPEGCodec::JPEGCodec ()
```

12.174.2.2 ~JPEGCodec()

```
gdcm::JPEGCodec::~JPEGCodec ()  [override]
```

12.174.3 Member Function Documentation

12.174.3.1 AppendFrameEncode()

```
bool gdcm::JPEGCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.174.3.2 AppendRowEncode()

```
bool gdcm::JPEGCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.174.3.3 CanCode()

```
bool gdcm::JPEGCodec::CanCode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

Examples

CompressLossyJPEG.cs.

12.174.3.4 CanDecode()

```
bool gdcm::JPEGCodec::CanDecode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.174.3.5 Clone()

```
ImageCodec * gdcm::JPEGCodec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.174.3.6 Code()

```
bool gdcm::JPEGCodec::Code (
            DataElement const & in,
            DataElement & out)  [override], [virtual]
```

Compress into JPEG.

Reimplemented from gdcm::Coder.

12.174.3.7 ComputeOffsetTable()

```
void gdcm::JPEGCodec::ComputeOffsetTable (
            bool b)
```

Compute the offset table:

12.174.3.8 Decode()

```
bool gdcm::JPEGCodec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

12.174.3.9 DecodeByStreams()

```
bool gdcm::JPEGCodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.174.3.10 DecodeExtent()

```
bool gdcm::JPEGCodec::DecodeExtent (
            char * buffer,
            unsigned int xmin,
            unsigned int xmax,
            unsigned int ymin,
            unsigned int ymax,
            unsigned int zmin,
            unsigned int zmax,
            std::istream & is)  [protected]
```

12.174.3.11 EncodeBuffer()

```
virtual bool gdcm::JPEGCodec::EncodeBuffer (
            std::ostream & out,
            const char * inbuffer,
            size_t inlen)  [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

12.174.3.12 GetHeaderInfo()

```
bool gdcm::JPEGCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

Examples

GetJPEGSamplePrecision.cxx.

12.174.3.13 GetLossless()

```
bool gdcm::JPEGCodec::GetLossless () const
```

12.174.3.14 GetQuality()

```
double gdcm::JPEGCodec::GetQuality () const
```

12.174.3.15 IsFrameEncoder()

```
bool gdcm::JPEGCodec::IsFrameEncoder ()  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.174.3.16 IsRowEncoder()

```
bool gdcm::JPEGCodec::IsRowEncoder ()  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.174.3.17 IsStateSuspension()

```
virtual bool gdcm::JPEGCodec::IsStateSuspension () const  [protected], [virtual]
```

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

12.174.3.18 IsValid()

```
bool gdcm::JPEGCodec::IsValid (
            PhotometricInterpretation const & pi)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.174.3.19 SetBitSample()

```
void gdcm::JPEGCodec::SetBitSample (
            int bit)  [protected]
```

12.174.3.20 SetLossless()

```
void gdcm::JPEGCodec::SetLossless (
            bool l)
```

Examples

CompressLossyJPEG.cs, and FileChangeTSLossy.cs.

12.174.3.21 SetPixelFormat()

```
void gdcm::JPEGCodec::SetPixelFormat (
            PixelFormat const & pf)  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

Examples

GetJPEGSamplePrecision.cxx.

12.174.3.22 SetQuality()

```
void gdcm::JPEGCodec::SetQuality (
            double q)
```

Examples

CompressLossyJPEG.cs, and FileChangeTSLossy.cs.

12.174.3.23 StartEncode()

```
bool gdcm::JPEGCodec::StartEncode (
            std::ostream & )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.174.3.24 StopEncode()

```
bool gdcm::JPEGCodec::StopEncode (
            std::ostream & )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.174.4 Friends And Related Symbol Documentation

12.174.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

References ImageRegionReader.

Referenced by ImageRegionReader.

12.174.5 Member Data Documentation

12.174.5.1 BitSample

```
int gdcm::JPEGCodec::BitSample [protected]
```

12.174.5.2 Quality

```
int gdcm::JPEGCodec::Quality [protected]
```

The documentation for this class was generated from the following file:

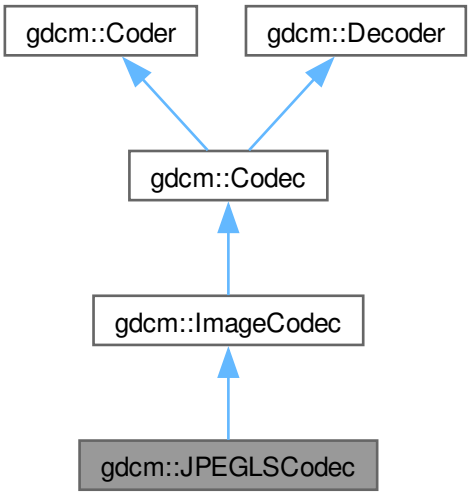
- gdcmJPEGCodec.h

12.175 gdcm::JPEGLSCodec Class Reference

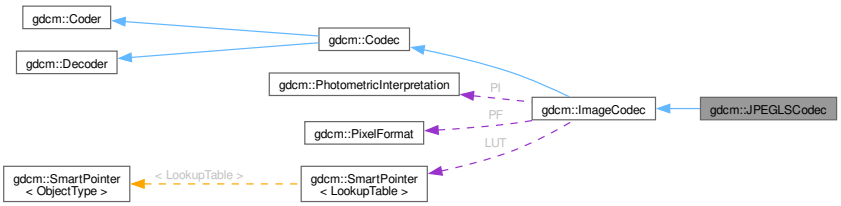
JPEG-LS.

```
#include <gdcmJPEGLSCodec.h>
```

Inheritance diagram for gdcm::JPEGLSCodec:



Collaboration diagram for gdcm::JPEGLSCodec:



Public Member Functions

- JPEGLSCodec ()
- ~JPEGLSCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &in, char *outBuffer, size_t inBufferLength, uint32_t inXMin, uint32_t inXMax, uint32_t inYMin, uint32_t inYMax, uint32_t inZMin, uint32_t inZMax)
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool GetLossless () const
- void SetBufferLength (unsigned long l)
- void SetLossless (bool l)
- void SetLossyError (int error)

  [0-3] generally

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const

- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeExtent (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Protected Member Functions inherited from gdcm::ImageCodec

- bool DecodeByStreams (std::istream &is_, std::ostream &os) override
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsValid (PhotometricInterpretation const &pi)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Friends

- class ImageRegionReader

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.175.1 Detailed Description

JPEG-LS.

Note

codec that implement the JPEG-LS compression this is an implementation of ImageCodec for JPEG-LS

It uses the CharLS JPEG-LS implementation `https://github.com/team-charls/charls`

12.175.2 Constructor & Destructor Documentation

12.175.2.1 JPEGLSCodec()

```
gdcm::JPEGLSCodec::JPEGLSCodec ()
```

12.175.2.2 ~JPEGLSCodec()

```
gdcm::JPEGLSCodec::~JPEGLSCodec () [override]
```

12.175.3 Member Function Documentation

12.175.3.1 AppendFrameEncode()

```
bool gdcm::JPEGLSCodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen) [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.175.3.2 AppendRowEncode()

```
bool gdcm::JPEGLSCodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen) [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.175.3.3 CanCode()

```
bool gdcm::JPEGLSCodec::CanCode (
            TransferSyntax const & ) const [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

12.175.3.4 CanDecode()

```
bool gdcm::JPEGLSCodec::CanDecode (
            TransferSyntax const & ) const [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.175.3.5 Clone()

```
ImageCodec * gdcm::JPEGLSCodec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.175.3.6 Code()

```
bool gdcm::JPEGLSCodec::Code (
            DataElement const & in_,
            DataElement & out_)  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

12.175.3.7 Decode() [1/2]

```
bool gdcm::JPEGLSCodec::Decode (
            DataElement const & in,
            char * outBuffer,
            size_t inBufferLength,
            uint32_t inXMin,
            uint32_t inXMax,
            uint32_t inYMin,
            uint32_t inYMax,
            uint32_t inZMin,
            uint32_t inZMax)
```

12.175.3.8 Decode() [2/2]

```
bool gdcm::JPEGLSCodec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

12.175.3.9 DecodeExtent()

```
bool gdcm::JPEGLSCodec::DecodeExtent (
            char * buffer,
            unsigned int xmin,
            unsigned int xmax,
            unsigned int ymin,
            unsigned int ymax,
            unsigned int zmin,
            unsigned int zmax,
            std::istream & is)  [protected]
```

12.175.3.10 GetBufferLength()

```
unsigned long gdcm::JPEGLSCodec::GetBufferLength () const  [inline]
```

12.175.3.11 GetHeaderInfo()

```
bool gdcm::JPEGLSCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.175.3.12 GetLossless()

```
bool gdcm::JPEGLSCodec::GetLossless () const
```

12.175.3.13 IsFrameEncoder()

```
bool gdcm::JPEGLSCodec::IsFrameEncoder ()  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.175.3.14 IsRowEncoder()

```
bool gdcm::JPEGLSCodec::IsRowEncoder ()  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.175.3.15 SetBufferLength()

```
void gdcm::JPEGLSCodec::SetBufferLength (
            unsigned long l)  [inline]
```

12.175.3.16 SetLossless()

```
void gdcm::JPEGLSCodec::SetLossless (
            bool l)
```

12.175.3.17 SetLossyError()

```
void gdcm::JPEGLSCodec::SetLossyError (
            int error)
```

[0-3] generally

12.175.3.18 StartEncode()

```
bool gdcm::JPEGLSCodec::StartEncode (
            std::ostream & )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.175.3.19 StopEncode()

```
bool gdcm::JPEGLSCodec::StopEncode (
            std::ostream & )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.175.4 Friends And Related Symbol Documentation

12.175.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

References ImageRegionReader.

Referenced by ImageRegionReader.

The documentation for this class was generated from the following file:

- gdcmJPEGLSCodec.h

12.176 gdcm::JSON Class Reference

```
#include <gdcmJSON.h>
```

Public Member Functions

- JSON ()
- ~JSON ()
- bool Code (DataSet const &in, std::ostream &os)
- bool Decode (std::istream &is, DataSet &out)
- bool GetPrettyPrint () const
- void PrettyPrintOff ()
- void PrettyPrintOn ()
- void SetPrettyPrint (bool onoff)

12.176.1 Detailed Description

Examples

QIDO-RS.cxx.

12.176.2 Constructor & Destructor Documentation

12.176.2.1 JSON()

```
gdcm::JSON::JSON ()
```

12.176.2.2 ~JSON()

```
gdcm::JSON::~JSON ()
```

12.176.3 Member Function Documentation

12.176.3.1 Code()

```
bool gdcm::JSON::Code (
            DataSet const & in,
            std::ostream & os)
```

Examples

QIDO-RS.cxx.

12.176.3.2 Decode()

```
bool gdcm::JSON::Decode (
            std::istream & is,
            DataSet & out)
```

Examples

QIDO-RS.cxx.

12.176.3.3 GetPrettyPrint()

```
bool gdcm::JSON::GetPrettyPrint () const
```

12.176.3.4 PrettyPrintOff()

```
void gdcm::JSON::PrettyPrintOff ()
```

12.176.3.5 PrettyPrintOn()

```
void gdcm::JSON::PrettyPrintOn ()
```

Examples

QIDO-RS.cxx.

12.176.3.6 SetPrettyPrint()

```
void gdcm::JSON::SetPrettyPrint (
            bool onoff)
```

The documentation for this class was generated from the following file:

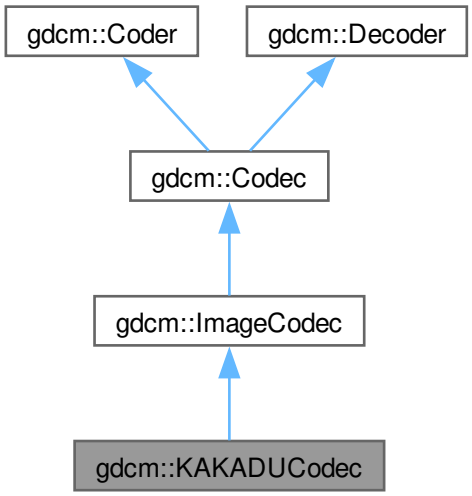
- gdcmJSON.h

12.177 gdcm::KAKADUCodec Class Reference

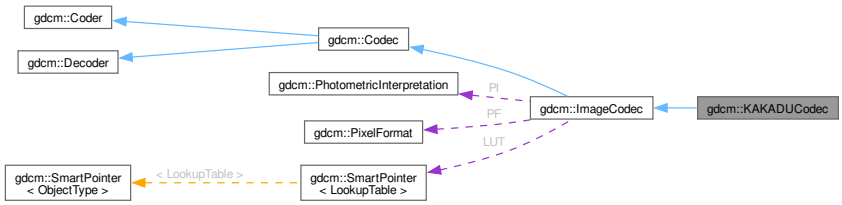
KAKADUCodec.

```
#include <gdcmKAKADUCodec.h>
```

Inheritance diagram for gdcm::KAKADUCodec:



Collaboration diagram for gdcm::KAKADUCodec:



Public Member Functions

- KAKADUCodec ()
- ~KAKADUCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- virtual bool GetHeaderInfo (std::istream &is_, TransferSyntax &ts)
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions inherited from gdcm::ImageCodec

- virtual bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- virtual bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is_, std::ostream &os) override
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsFrameEncoder ()
- virtual bool IsRowEncoder ()
- virtual bool IsValid (PhotometricInterpretation const &pi)
- virtual bool StartEncode (std::ostream &os)
- virtual bool StopEncode (std::ostream &os)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.177.1 Detailed Description

KAKADUCodec.

12.177.2 Constructor & Destructor Documentation

12.177.2.1 KAKADUCodec()

```
gdcm::KAKADUCodec::KAKADUCodec ()
```

12.177.2.2 ~KAKADUCodec()

```
gdcm::KAKADUCodec::~KAKADUCodec ()  [override]
```

12.177.3 Member Function Documentation

12.177.3.1 CanCode()

```
bool gdcm::KAKADUCodec::CanCode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

12.177.3.2 CanDecode()

```
bool gdcm::KAKADUCodec::CanDecode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.177.3.3 Clone()

```
ImageCodec * gdcm::KAKADUCodec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.177.3.4 Code()

```
bool gdcm::KAKADUCodec::Code (
            DataElement const & in_,
            DataElement & out_)  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

12.177.3.5 Decode()

```
bool gdcm::KAKADUCodec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

The documentation for this class was generated from the following file:

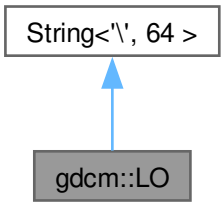
- gdcmKAKADUCodec.h

12.178 gdcm::LO Class Reference

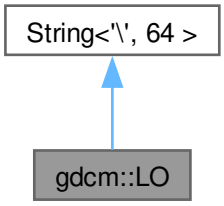
LO.

```
#include <gdcmLO.h>
```

Inheritance diagram for gdcm::LO:



Collaboration diagram for gdcm::LO:



Public Types

- typedef Superclass::const_iterator const_iterator
- typedef Superclass::const_reference const_reference
- typedef Superclass::const_reverse_iterator const_reverse_iterator
- typedef Superclass::difference_type difference_type
- typedef Superclass::iterator iterator
- typedef Superclass::pointer pointer
- typedef Superclass::reference reference
- typedef Superclass::reverse_iterator reverse_iterator
- typedef Superclass::size_type size_type
- typedef String<'\\', 64 > Superclass
- typedef Superclass::value_type value_type

Public Member Functions

- LO ()
- LO (const Superclass &s, size_type pos=0, size_type n=npos)
- LO (const value_type *s)
- LO (const value_type *s, size_type n)
- bool IsValid () const

12.178.1 Detailed Description

LO.

Note

TODO

12.178.2 Member Typedef Documentation

12.178.2.1 const_iterator

```
typedef Superclass::const_iterator gdcm::LO::const_iterator
```

12.178.2.2 const_reference

```
typedef Superclass::const_reference gdcm::LO::const_reference
```

12.178.2.3 const_reverse_iterator

```
typedef Superclass::const_reverse_iterator gdcm::LO::const_reverse_iterator
```

12.178.2.4 difference_type

```
typedef Superclass::difference_type gdcm::LO::difference_type
```

12.178.2.5 iterator

```
typedef Superclass::iterator gdcm::LO::iterator
```

12.178.2.6 pointer

```
typedef Superclass::pointer gdcm::LO::pointer
```

12.178.2.7 reference

```
typedef Superclass::reference gdcm::LO::reference
```

12.178.2.8 reverse_iterator

```
typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator
```

12.178.2.9 size_type

```
typedef Superclass::size_type gdcm::LO::size_type
```

12.178.2.10 Superclass

```
typedef String<'\\',64> gdcm::LO::Superclass
```

12.178.2.11 value_type

```
typedef Superclass::value_type gdcm::LO::value_type
```

12.178.3 Constructor & Destructor Documentation

12.178.3.1 LO() [1/4]

```
gdcm::LO::LO ()  [inline]
```

12.178.3.2 LO() [2/4]

```
gdcm::LO::LO (
            const value_type * s)  [inline]
```

12.178.3.3 LO() [3/4]

```
gdcm::LO::LO (
            const value_type * s,
            size_type n)  [inline]
```

12.178.3.4 LO() [4/4]

```
gdcm::LO::LO (
            const Superclass & s,
            size_type pos = 0,
            size_type n = npos)  [inline]
```

12.178.4 Member Function Documentation

12.178.4.1 IsValid()

```
bool gdcm::LO::IsValid () const  [inline]
```

References gdcm::String<'\\', 64 >::IsValid().

The documentation for this class was generated from the following file:

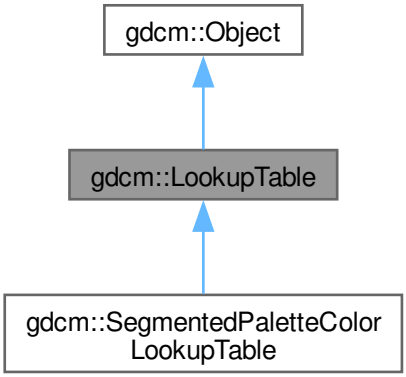
- gdcmLO.h

12.179 gdcm::LookupTable Class Reference

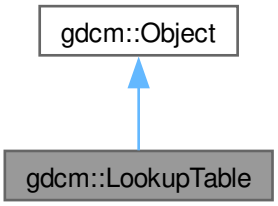
LookupTable class.

```
#include <gdcmLookupTable.h>
```

Inheritance diagram for gdcm::LookupTable:



Collaboration diagram for gdcm::LookupTable:



Public Types

- enum LookupTableType {
  RED = 0 ,
  GREEN ,
  BLUE ,
  GRAY ,
  UNKNOWN }

Public Member Functions

- LookupTable ()
- LookupTable (LookupTable const &lut)
- ~LookupTable () override
- void Allocate (unsigned short bitsample=8)

  Allocate the LUT.
- void Clear ()

  Clear the LUT.
- bool Decode (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
- void Decode (std::istream &is, std::ostream &os) const

  Decode the LUT.
- bool Decode8 (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const

  Decode into RGB 8 bits space.
- unsigned short GetBitSample () const

  return the bit sample
- bool GetBufferAsRGBA (unsigned char *rgba) const

  return the LUT as RGBA buffer
- void GetLUT (LookupTableType type, unsigned char *array, unsigned int &length) const
- void GetLUTDescriptor (LookupTableType type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const
- unsigned int GetLUTLength (LookupTableType type) const
- const unsigned char * GetPointer () const

  return a raw pointer to the LUT
- void InitializeBlueLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)
- bool Initialized () const

  return whether the LUT has been initialized
- void InitializeGreenLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)
- void InitializeLUT (LookupTableType type, unsigned short length, unsigned short subscript, unsigned short bitsize)

  Generic interface:
- void InitializeRedLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)

  RED / GREEN / BLUE specific:
- bool IsRGB8 () const

  Return whether 16 bits LUT is in RGB 8 bits space.
- void Print (std::ostream &) const override
- void SetBlueLUT (const unsigned char *blue, unsigned int length)
- void SetGreenLUT (const unsigned char *green, unsigned int length)
- virtual void SetLUT (LookupTableType type, const unsigned char *array, unsigned int length)
- void SetRedLUT (const unsigned char *red, unsigned int length)
- bool WriteBufferAsRGBA (const unsigned char *rgba)

  Write the LUT as RGBA.

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Protected Attributes

- unsigned short BitSample
- bool IncompleteLUT:1
- LookupTableInternal * Internal

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.179.1 Detailed Description

LookupTable class.

Examples

ExtractImageRegionWithLUT.cs, and PrintLUT.cxx.

12.179.2 Member Enumeration Documentation

12.179.2.1 LookupTableType

```
enum gdcm::LookupTable::LookupTableType
```

Enumerator

RED	
GREEN	
BLUE	
GRAY	
UNKNOWN	

12.179.3 Constructor & Destructor Documentation

12.179.3.1 LookupTable() [1/2]

```
gdcm::LookupTable::LookupTable ()
```

Referenced by LookupTable().

12.179.3.2 ~LookupTable()

```
gdcm::LookupTable::~LookupTable () [override]
```

12.179.3.3 LookupTable() [2/2]

```
gdcm::LookupTable::LookupTable (
            LookupTable const & lut) [inline]
```

References LookupTable(), and gdcm::Object::Object().

12.179.4 Member Function Documentation

12.179.4.1 Allocate()

```
void gdcm::LookupTable::Allocate (
            unsigned short bitsample = 8)
```

Allocate the LUT.

12.179.4.2 Clear()

```
void gdcm::LookupTable::Clear ()
```

Clear the LUT.

12.179.4.3 Decode() [1/2]

```
bool gdcm::LookupTable::Decode (
            char * outputbuffer,
            size_t outlen,
            const char * inputbuffer,
            size_t inlen) const
```

Decode the LUT outputbuffer will contains the RGB decoded PALETTE COLOR input image of size inlen the outputbuffer should be at least 3 times the size of inlen

12.179.4.4 Decode() [2/2]

```
void gdcm::LookupTable::Decode (
            std::istream & is,
            std::ostream & os) const
```

Decode the LUT.

Examples

ExtractImageRegionWithLUT.cs.

12.179.4.5 Decode8()

```
bool gdcm::LookupTable::Decode8 (
            char * outputbuffer,
            size_t outlen,
            const char * inputbuffer,
            size_t inlen) const
```

Decode into RGB 8 bits space.

12.179.4.6 GetBitSample()

```
unsigned short gdcm::LookupTable::GetBitSample () const  [inline]
```

return the bit sample

References BitSample.

12.179.4.7 GetBufferAsRGBA()

```
bool gdcm::LookupTable::GetBufferAsRGBA (
            unsigned char * rgba) const
```

return the LUT as RGBA buffer

12.179.4.8 GetLUT()

```
void gdcm::LookupTable::GetLUT (
            LookupTableType type,
            unsigned char * array,
            unsigned int & length) const
```

12.179.4.9 GetLUTDescriptor()

```
void gdcm::LookupTable::GetLUTDescriptor (
            LookupTableType type,
            unsigned short & length,
            unsigned short & subscript,
            unsigned short & bitsize) const
```

12.179.4.10 GetLUTLength()

```
unsigned int gdcm::LookupTable::GetLUTLength (
            LookupTableType type) const
```

12.179.4.11 GetPointer()

```
const unsigned char * gdcm::LookupTable::GetPointer () const
```

return a raw pointer to the LUT

12.179.4.12 InitializeBlueLUT()

```
void gdcm::LookupTable::InitializeBlueLUT (
            unsigned short length,
            unsigned short subscript,
            unsigned short bitsize)
```

12.179.4.13 Initialized()

```
bool gdcm::LookupTable::Initialized () const
```

return whether the LUT has been initialized

12.179.4.14 InitializeGreenLUT()

```
void gdcm::LookupTable::InitializeGreenLUT (
            unsigned short length,
            unsigned short subscript,
            unsigned short bitsize)
```

12.179.4.15 InitializeLUT()

```
void gdcm::LookupTable::InitializeLUT (
            LookupTableType type,
            unsigned short length,
            unsigned short subscript,
            unsigned short bitsize)
```

Generic interface:

12.179.4.16 InitializeRedLUT()

```
void gdcm::LookupTable::InitializeRedLUT (
            unsigned short length,
            unsigned short subscript,
            unsigned short bitsize)
```

RED / GREEN / BLUE specific:

12.179.4.17 IsRGB8()

```
bool gdcm::LookupTable::IsRGB8 () const
```

Return whether 16 bits LUT is in RGB 8 bits space.

12.179.4.18 Print()

```
void gdcm::LookupTable::Print (
            std::ostream & ) const  [override], [virtual]
```

Reimplemented from gdcm::Object.

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

Examples

PrintLUT.cxx.

12.179.4.19 SetBlueLUT()

```
void gdcm::LookupTable::SetBlueLUT (
            const unsigned char * blue,
            unsigned int length)
```

12.179.4.20 SetGreenLUT()

```
void gdcm::LookupTable::SetGreenLUT (
            const unsigned char * green,
            unsigned int length)
```

12.179.4.21 SetLUT()

```
virtual void gdcm::LookupTable::SetLUT (
            LookupTableType type,
            const unsigned char * array,
            unsigned int length)  [virtual]
```

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

12.179.4.22 SetRedLUT()

```
void gdcm::LookupTable::SetRedLUT (
            const unsigned char * red,
            unsigned int length)
```

12.179.4.23 WriteBufferAsRGBA()

```
bool gdcm::LookupTable::WriteBufferAsRGBA (
            const unsigned char * rgba)
```

Write the LUT as RGBA.

12.179.5 Member Data Documentation

12.179.5.1 BitSample

```
unsigned short gdcm::LookupTable::BitSample [protected]
```

Referenced by GetBitSample().

12.179.5.2 IncompleteLUT

```
bool gdcm::LookupTable::IncompleteLUT [protected]
```

12.179.5.3 Internal

```
LookupTableInternal* gdcm::LookupTable::Internal [protected]
```

The documentation for this class was generated from the following file:

- gdcmLookupTable.h

12.180 gdcm::Scanner2::ltstr Struct Reference

```
#include <gdcmScanner2.h>
```

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

12.180.1 Member Function Documentation

12.180.1.1 operator()()

```
bool gdcm::Scanner2::ltstr::operator() (
            const char * s1,
            const char * s2) const  [inline]
```

The documentation for this struct was generated from the following file:

- gdcmScanner2.h

12.181 gdcm::Scanner::ltstr Struct Reference

```
#include <gdcmScanner.h>
```

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

12.181.1 Member Function Documentation

12.181.1.1 operator()()

```
bool gdcm::Scanner::ltstr::operator() (
            const char * s1,
            const char * s2) const  [inline]
```

The documentation for this struct was generated from the following file:

- gdcmScanner.h

12.182 gdcm::StrictScanner2::ltstr Struct Reference

```
#include <gdcmStrictScanner2.h>
```

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

12.182.1 Member Function Documentation

12.182.1.1 operator()()

```
bool gdcm::StrictScanner2::ltstr::operator() (
            const char * s1,
            const char * s2) const  [inline]
```

The documentation for this struct was generated from the following file:

- gdcmStrictScanner2.h

12.183 gdcm::StrictScanner::ltstr Struct Reference

```
#include <gdcmStrictScanner.h>
```

Public Member Functions

- bool operator() (const char ∗s1, const char ∗s2) const

12.183.1 Member Function Documentation

12.183.1.1 operator()()

```
bool gdcm::StrictScanner::ltstr::operator() (
            const char * s1,
            const char * s2) const  [inline]
```

The documentation for this struct was generated from the following file:

- gdcmStrictScanner.h

12.184 gdcm::Macro Class Reference

Class for representing a Macro.

```
#include <gdcmMacro.h>
```

Public Types

- typedef std::vector< std::string > ArrayIncludeMacrosType
- typedef std::map< Tag, MacroEntry > MapModuleEntry

Public Member Functions

- Macro ()=default
- void AddMacroEntry (const Tag &tag, const MacroEntry &module)

  Will add a ModuleEntry directly at root-level. See Macro for nested-included level.
- void Clear ()
- bool FindMacroEntry (const Tag &tag) const
- const MacroEntry & GetMacroEntry (const Tag &tag) const
- const char * GetName () const
- void SetName (const char *name)
- bool Verify (const DataSet &ds, Usage const &usage) const

Friends

- std::ostream & operator<< (std::ostream &_os, const Macro &_val)

12.184.1 Detailed Description

Class for representing a Macro.

Note

Attribute Macro: a set of Attributes that are described in a single table that is referenced by multiple Module or other tables.

See also

Module

12.184.2 Member Typedef Documentation

12.184.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcm::Macro::ArrayIncludeMacrosType
```

12.184.2.2 MapModuleEntry

```
typedef std::map<Tag, MacroEntry> gdcm::Macro::MapModuleEntry
```

12.184.3 Constructor & Destructor Documentation

12.184.3.1 Macro()

```
gdcm::Macro::Macro () [default]
```

References Macro(), and operator<<.

Referenced by Macro(), and operator<<.

12.184.4 Member Function Documentation

12.184.4.1 AddMacroEntry()

```
void gdcm::Macro::AddMacroEntry (
            const Tag & tag,
            const MacroEntry & module) [inline]
```

Will add a ModuleEntry directly at root-level. See Macro for nested-included level.

12.184.4.2 Clear()

```
void gdcm::Macro::Clear () [inline]
```

12.184.4.3 FindMacroEntry()

```
bool gdcm::Macro::FindMacroEntry (
            const Tag & tag) const
```

Find or Get a ModuleEntry. ModuleEntry are either search are root-level or within nested-macro included in module.

12.184.4.4 GetMacroEntry()

```
const MacroEntry & gdcm::Macro::GetMacroEntry (
            const Tag & tag) const
```

12.184.4.5 GetName()

```
const char * gdcm::Macro::GetName () const [inline]
```

12.184.4.6 SetName()

```
void gdcm::Macro::SetName (
            const char * name)  [inline]
```

12.184.4.7 Verify()

```
bool gdcm::Macro::Verify (
            const DataSet & ds,
            Usage const & usage) const
```

12.184.5 Friends And Related Symbol Documentation

12.184.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Macro & _val)  [friend]
```

References Macro().

Referenced by Macro().

The documentation for this class was generated from the following file:

- gdcmMacro.h

12.185 gdcm::Macros Class Reference

Class for representing a Modules.

```
#include <gdcmMacros.h>
```

Public Types

- typedef std::map< std::string, Macro > ModuleMapType

Public Member Functions

- Macros ()=default
- void AddMacro (const char *ref, const Macro &module)
- void Clear ()
- const Macro & GetMacro (const char *name) const
- bool IsEmpty () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Macros &_val)

12.185.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples

TraverseModules.cxx.

12.185.2 Member Typedef Documentation

12.185.2.1 ModuleMapType

```
typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType
```

12.185.3 Constructor & Destructor Documentation

12.185.3.1 Macros()

```
gdcm::Macros::Macros () [default]
```

References Macros(), and operator<<.

Referenced by Macros(), and operator<<.

12.185.4 Member Function Documentation

12.185.4.1 AddMacro()

```
void gdcm::Macros::AddMacro (
            const char * ref,
            const Macro & module) [inline]
```

12.185.4.2 Clear()

```
void gdcm::Macros::Clear () [inline]
```

12.185.4.3 GetMacro()

```
const Macro & gdcm::Macros::GetMacro (
            const char * name) const [inline]
```

12.185.4.4 IsEmpty()

```
bool gdcm::Macros::IsEmpty () const [inline]
```

12.185.5 Friends And Related Symbol Documentation

12.185.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Macros & _val) [friend]
```

References Macros().

Referenced by Macros().

The documentation for this class was generated from the following file:

- gdcmMacros.h

12.186 gdcm::network::MaximumLengthSub Class Reference

MaximumLengthSub.

```
#include <gdcmMaximumLengthSub.h>
```

Public Member Functions

- MaximumLengthSub ()
- uint32_t GetMaximumLength () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetMaximumLength (uint32_t maximumlength)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.186.1 Detailed Description

MaximumLengthSub.

Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

or

Table D.1-2 Maximum length sub-item fields (A-ASSOCIATE-AC)

12.186.2 Constructor & Destructor Documentation

12.186.2.1 MaximumLengthSub()

```
gdcm::network::MaximumLengthSub::MaximumLengthSub ()
```

12.186.3 Member Function Documentation

12.186.3.1 GetMaximumLength()

```
uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength () const  [inline]
```

12.186.3.2 Print()

```
void gdcm::network::MaximumLengthSub::Print (
            std::ostream & os) const
```

12.186.3.3 Read()

```
std::istream & gdcm::network::MaximumLengthSub::Read (
            std::istream & is)
```

12.186.3.4 SetMaximumLength()

```
void gdcm::network::MaximumLengthSub::SetMaximumLength (
            uint32_t maximumlength)
```

12.186.3.5 Size()

```
size_t gdcm::network::MaximumLengthSub::Size () const
```

12.186.3.6 Write()

```
const std::ostream & gdcm::network::MaximumLengthSub::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

- gdcmMaximumLengthSub.h

12.187 gdcm::MD5 Class Reference

Class for MD5.

```
#include <gdcmMD5.h>
```

Static Public Member Functions

- static bool Compute (const char *buffer, size_t buf_len, char digest_str[33])
- static bool ComputeFile (const char *filename, char digest_str[33])

  Compute md5 from a file `filename`.

12.187.1 Detailed Description

Class for MD5.

Warning

this class is able to pick from two implementations:

1. a lightweight md5 implementation (when GDCM_BUILD_TESTING is turned ON)
2. the one from OpenSSL (when GDCM_USE_SYSTEM_OPENSSL is turned ON)

In all other cases it will return an error

12.187.2 Member Function Documentation

12.187.2.1 Compute()

```
bool gdcm::MD5::Compute (
            const char * buffer,
            size_t buf_len,
            char digest_str[33])  [static]
```

12.187.2.2 ComputeFile()

```
bool gdcm::MD5::ComputeFile (
            const char * filename,
            char digest_str[33])  [static]
```

Compute md5 from a file `filename`.

The documentation for this class was generated from the following file:

- gdcmMD5.h

12.188 gdcm::MEC_MR3 Class Reference

Class for MEC_MR3.

```
#include <gdcmMEC_MR3.h>
```

Static Public Member Functions

- static const PrivateTag & GetCanonMECMR3Tag ()
- static const PrivateTag & GetPMTFInformationDataTag ()
- static const PrivateTag & GetToshibaMECMR3Tag ()
- static bool Print (const char *src, size_t srclen)

12.188.1 Detailed Description

Class for MEC_MR3.

12.188.2 Member Function Documentation

12.188.2.1 GetCanonMECMR3Tag()

```
const PrivateTag & gdcm::MEC_MR3::GetCanonMECMR3Tag ()  [static]
```

Return the private tag used by CANON to store the MEC_MR3 data This is: PrivateTag(0x0029,0x90,"CANON_MEC↩ _MR3");

12.188.2.2 GetPMTFInformationDataTag()

```
const PrivateTag & gdcm::MEC_MR3::GetPMTFInformationDataTag ()  [static]
```

Return the private tag used by PMTF to store the MEC_MR3 data This is: PrivateTag(0x0029,0x90,"PMTF INFORMATION DATA");

12.188.2.3 GetToshibaMECMR3Tag()

```
const PrivateTag & gdcm::MEC_MR3::GetToshibaMECMR3Tag ()  [static]
```

Return the private tag used by TOSHIBA to store the MEC_MR3 data This is: PrivateTag(0x0029,0x90,"TOSHIBA_↩ MEC_MR3");

12.188.2.4 Print()

```
bool gdcm::MEC_MR3::Print (
            const char * src,
            size_t srclen)  [static]
```

The documentation for this class was generated from the following file:

- gdcmMEC_MR3.h

12.189 gdcm::MediaStorage Class Reference

MediaStorage.

```
#include <gdcmMediaStorage.h>
```

Public Types

- enum MSType {
  MediaStorageDirectoryStorage = 0 ,
  ComputedRadiographyImageStorage ,
  DigitalXRayImageStorageForPresentation ,
  DigitalXRayImageStorageForProcessing ,
  DigitalMammographyImageStorageForPresentation ,
  DigitalMammographyImageStorageForProcessing ,
  DigitalIntraoralXrayImageStorageForPresentation ,
  DigitalIntraoralXRayImageStorageForProcessing ,
  CTImageStorage ,
  EnhancedCTImageStorage ,
  UltrasoundImageStorageRetired ,
  UltrasoundImageStorage ,
  UltrasoundMultiFrameImageStorageRetired ,
  UltrasoundMultiFrameImageStorage ,
  MRImageStorage ,
  EnhancedMRImageStorage ,
  MRSpectroscopyStorage ,
  NuclearMedicineImageStorageRetired ,
  SecondaryCaptureImageStorage ,
  MultiframeSingleBitSecondaryCaptureImageStorage ,
  MultiframeGrayscaleByteSecondaryCaptureImageStorage ,
  MultiframeGrayscaleWordSecondaryCaptureImageStorage ,

MultiframeTrueColorSecondaryCaptureImageStorage ,
StandaloneOverlayStorage ,
StandaloneCurveStorage ,
LeadECGWaveformStorage ,
GeneralECGWaveformStorage ,
AmbulatoryECGWaveformStorage ,
HemodynamicWaveformStorage ,
CardiacElectrophysiologyWaveformStorage ,
BasicVoiceAudioWaveformStorage ,
StandaloneModalityLUTStorage ,
StandaloneVOILUTStorage ,
GrayscaleSoftcopyPresentationStateStorageSOPClass ,
XRayAngiographicImageStorage ,
XRayRadiofluoroscopingImageStorage ,
XRayAngiographicBiPlaneImageStorageRetired ,
NuclearMedicineImageStorage ,
RawDataStorage ,
SpacialRegistrationStorage ,
SpacialFiducialsStorage ,
PETImageStorage ,
RTImageStorage ,
RTDoseStorage ,
RTStructureSetStorage ,
RTPlanStorage ,
CSANonImageStorage ,
Philips3D ,
EnhancedSR ,
BasicTextSR ,
HardcopyGrayscaleImageStorage ,
ComprehensiveSR ,
DetachedStudyManagementSOPClass ,
EncapsulatedPDFStorage ,
EncapsulatedCDAStorage ,
StudyComponentManagementSOPClass ,
DetachedVisitManagementSOPClass ,
DetachedPatientManagementSOPClass ,
VideoEndoscopicImageStorage ,
GeneralElectricMagneticResonanceImageStorage ,
GEPrivate3DModelStorage ,
ToshibaPrivateDataStorage ,
MammographyCADSR ,
KeyObjectSelectionDocument ,
HangingProtocolStorage ,
ModalityPerformedProcedureStepSOPClass ,
PhilipsPrivateMRSyntheticImageStorage ,
VLPhotographicImageStorage ,
SegmentationStorage ,
RTIonPlanStorage ,
XRay3DAngiographicImageStorage ,
EnhancedXAImageStorage ,
RTIonBeamsTreatmentRecordStorage ,
SurfaceSegmentationStorage ,
VLWholeSlideMicroscopyImageStorage ,
RTTreatmentSummaryRecordStorage ,

EnhancedUSVolumeStorage ,
XRayRadiationDoseSR ,
VLEndoscopicImageStorage ,
BreastTomosynthesisImageStorage ,
FujiPrivateCRImageStorage ,
OphthalmicPhotography8BitImageStorage ,
OphthalmicTomographyImageStorage ,
VLMicroscopicImageStorage ,
EnhancedPETImageStorage ,
VideoPhotographicImageStorage ,
XRay3DCraniofacialImageStorage ,
IVOCTForPresentation ,
IVOCTForProcessing ,
LegacyConvertedEnhancedCTImageStorage ,
LegacyConvertedEnhancedMRImageStorage ,
LegacyConvertedEnhancedPETImageStorage ,
BreastProjectionXRayImageStorageForPresentation ,
BreastProjectionXRayImageStorageForProcessing ,
HardcopyColorImageStorage ,
EnhancedMRColorImageStorage ,
FujiPrivateMammoCRImageStorage ,
OphthalmicPhotography16BitImageStorage ,
VideoMicroscopicImageStorage ,
MS_END }

- enum ObjectType {
  NoObject = 0 ,
  Video ,
  Waveform ,
  Audio ,
  PDF ,
  URI ,
  Segmentation ,
  ObjectEnd }

Public Member Functions

- MediaStorage (MSType type=MS_END)
- const char ∗ GetModality () const
- unsigned int GetModalityDimension () const
- const char ∗ GetString () const

  Return the Media String of the object.
- void GuessFromModality (const char ∗modality, unsigned int dimension=2)
- bool IsUndefined () const
- operator MSType () const
- bool SetFromDataSet (DataSet const &ds)
- bool SetFromFile (File const &file)
- bool SetFromHeader (FileMetaInformation const &fmi)
- bool SetFromModality (DataSet const &ds)

Static Public Member Functions

- static const char * GetMSString (MSType ts)

  Return the Media String associated. Will return NULL for MS_END.
- static MSType GetMSType (const char *str)
- static unsigned int GetNumberOfModality ()
- static unsigned int GetNumberOfMSString ()
- static unsigned int GetNumberOfMSType ()
- static bool IsImage (MSType ts)

Protected Member Functions

- void SetFromSourceImageSequence (DataSet const &ds)

Friends

- std::ostream & operator<< (std::ostream &os, const MediaStorage &ms)

12.189.1 Detailed Description

MediaStorage.

Note

FIXME There should not be any notion of Image and/or PDF at that point Only the codec can answer yes I support this Media Storage or not... For instance an ImageCodec will answer yes to most of them while a PDFCodec will answer only for the Encapsulated PDF

See also

UIDs

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenerateStandardSOPClasses.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, TestReader.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, and iU22tomultisc.cxx.

12.189.2 Member Enumeration Documentation

12.189.2.1 MSType

enum gdcm::MediaStorage::MSType

Enumerator

MediaStorageDirectoryStorage	
ComputedRadiographyImageStorage	
DigitalXRayImageStorageForPresentation	
DigitalXRayImageStorageForProcessing	
DigitalMammographyImageStorageForPresentation	
DigitalMammographyImageStorageForProcessing	
DigitalIntraoralXrayImageStorageForPresentation	
DigitalIntraoralXRayImageStorageForProcessing	
CTImageStorage	
EnhancedCTImageStorage	
UltrasoundImageStorageRetired	
UltrasoundImageStorage	
UltrasoundMultiFrameImageStorageRetired	
UltrasoundMultiFrameImageStorage	
MRImageStorage	
EnhancedMRImageStorage	
MRSpectroscopyStorage	
NuclearMedicineImageStorageRetired	
SecondaryCaptureImageStorage	
MultiframeSingleBitSecondaryCaptureImageStorage	
MultiframeGrayscaleByteSecondaryCaptureImageStorage	
MultiframeGrayscaleWordSecondaryCaptureImageStorage	
MultiframeTrueColorSecondaryCaptureImageStorage	
StandaloneOverlayStorage	
StandaloneCurveStorage	
LeadECGWaveformStorage	
GeneralECGWaveformStorage	
AmbulatoryECGWaveformStorage	
HemodynamicWaveformStorage	
CardiacElectrophysiologyWaveformStorage	
BasicVoiceAudioWaveformStorage	
StandaloneModalityLUTStorage	
StandaloneVOILUTStorage	

GrayscaleSoftcopyPresentationStateStorageSOPClass	
XRayAngiographicImageStorage	
XRayRadiofluoroscopingImageStorage	
XRayAngiographicBiPlaneImageStorageRetired	
NuclearMedicineImageStorage	
RawDataStorage	
SpacialRegistrationStorage	
SpacialFiducialsStorage	
PETImageStorage	
RTImageStorage	
RTDoseStorage	
RTStructureSetStorage	
RTPlanStorage	
CSANonImageStorage	
Philips3D	
EnhancedSR	
BasicTextSR	
HardcopyGrayscaleImageStorage	
ComprehensiveSR	
DetachedStudyManagementSOPClass	
EncapsulatedPDFStorage	
EncapsulatedCDAStorage	
StudyComponentManagementSOPClass	
DetachedVisitManagementSOPClass	
DetachedPatientManagementSOPClass	
VideoEndoscopicImageStorage	
GeneralElectricMagneticResonanceImageStorage	
GEPrivate3DModelStorage	
ToshibaPrivateDataStorage	
MammographyCADSR	
KeyObjectSelectionDocument	
HangingProtocolStorage	
ModalityPerformedProcedureStepSOPClass	
PhilipsPrivateMRSyntheticImageStorage	
VLPhotographicImageStorage	
SegmentationStorage	
RTIonPlanStorage	
XRay3DAngiographicImageStorage	
EnhancedXAImageStorage	

RTIonBeamsTreatmentRecordStorage	
SurfaceSegmentationStorage	
VLWholeSlideMicroscopyImageStorage	
RTTreatmentSummaryRecordStorage	
EnhancedUSVolumeStorage	
XRayRadiationDoseSR	
VLEndoscopicImageStorage	
BreastTomosynthesisImageStorage	
FujiPrivateCRImageStorage	
OphthalmicPhotography8BitImageStorage	
OphthalmicTomographyImageStorage	
VLMicroscopicImageStorage	
EnhancedPETImageStorage	
VideoPhotographicImageStorage	
XRay3DCraniofacialImageStorage	
IVOCTForPresentation	
IVOCTForProcessing	
LegacyConvertedEnhancedCTImageStorage	
LegacyConvertedEnhancedMRImageStorage	
LegacyConvertedEnhancedPETImageStorage	
BreastProjectionXRayImageStorageForPresentation	
BreastProjectionXRayImageStorageForProcessing	
HardcopyColorImageStorage	
EnhancedMRColorImageStorage	
FujiPrivateMammoCRImageStorage	
OphthalmicPhotography16BitImageStorage	
VideoMicroscopicImageStorage	
MS_END	

Examples

GenerateStandardSOPClasses.cxx, and MpegVideoInfo.cs.

12.189.2.2 ObjectType

```
enum gdcm::MediaStorage::ObjectType
```

Enumerator

NoObject	

Video	
Waveform	
Audio	
PDF	
URI	
Segmentation	
ObjectEnd	

12.189.3 Constructor & Destructor Documentation

12.189.3.1 MediaStorage()

```
gdcm::MediaStorage::MediaStorage (
            MSType type = MS_END)  [inline]
```

References MS_END.

Referenced by GuessFromModality(), and operator<<.

12.189.4 Member Function Documentation

12.189.4.1 GetModality()

```
const char * gdcm::MediaStorage::GetModality () const
```

12.189.4.2 GetModalityDimension()

```
unsigned int gdcm::MediaStorage::GetModalityDimension () const
```

12.189.4.3 GetMSString()

```
const char * gdcm::MediaStorage::GetMSString (
            MSType ts)  [static]
```

Return the Media String associated. Will return NULL for MS_END.

Examples

GenerateStandardSOPClasses.cxx.

Referenced by operator<<.

12.189.4.4 GetMSType()

```
MSType gdcm::MediaStorage::GetMSType (
            const char * str)  [static]
```

Examples

MetaImageMD5Activiz.cs, and TestReader.cxx.

12.189.4.5 GetNumberOfModality()

```
unsigned int gdcm::MediaStorage::GetNumberOfModality ()  [static]
```

12.189.4.6 GetNumberOfMSString()

```
unsigned int gdcm::MediaStorage::GetNumberOfMSString ()  [static]
```

12.189.4.7 GetNumberOfMSType()

```
unsigned int gdcm::MediaStorage::GetNumberOfMSType ()  [static]
```

12.189.4.8 GetString()

```
const char * gdcm::MediaStorage::GetString () const
```

Return the Media String of the object.

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, and iU22tomultisc.cxx.

12.189.4.9 GuessFromModality()

```
void gdcm::MediaStorage::GuessFromModality (
            const char * modality,
            unsigned int dimension = 2)
```

References MediaStorage(), and operator<<.

12.189.4.10 IsImage()

```
bool gdcm::MediaStorage::IsImage (
            MSType ts)  [static]
```

Returns whether DICOM has a Pixel Data element (7fe0,0010)

Warning

MRSpectroscopyStorage could be image but are not

Examples

MetaImageMD5Activiz.cs.

12.189.4.11 IsUndefined()

```
bool gdcm::MediaStorage::IsUndefined () const  [inline]
```

Examples

TestReader.cxx.

References MS_END.

12.189.4.12 operator MSType()

```
gdcm::MediaStorage::operator MSType () const  [inline]
```

12.189.4.13 SetFromDataSet()

```
bool gdcm::MediaStorage::SetFromDataSet (
            DataSet const & ds)
```

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

12.189.4.14 SetFromFile()

```
bool gdcm::MediaStorage::SetFromFile (
            File const & file)
```

Attempt to set the MediaStorage from a file: WARNING: When no MediaStorage & Modality are found BUT a PixelData element is found then MediaStorage is set to the default SecondaryCaptureImageStorage (return value is false in this case)

Examples

ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, TestReader.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.189.4.15 SetFromHeader()

```
bool gdcm::MediaStorage::SetFromHeader (
            FileMetaInformation const & fmi)
```

12.189.4.16 SetFromModality()

```
bool gdcm::MediaStorage::SetFromModality (
            DataSet const & ds)
```

12.189.4.17 SetFromSourceImageSequence()

```
void gdcm::MediaStorage::SetFromSourceImageSequence (
            DataSet const & ds)  [protected]
```

12.189.5 Friends And Related Symbol Documentation

12.189.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const MediaStorage & ms)  [friend]
```

References MediaStorage(), GetMSString(), and operator<<.

Referenced by GuessFromModality(), and operator<<.

The documentation for this class was generated from the following file:

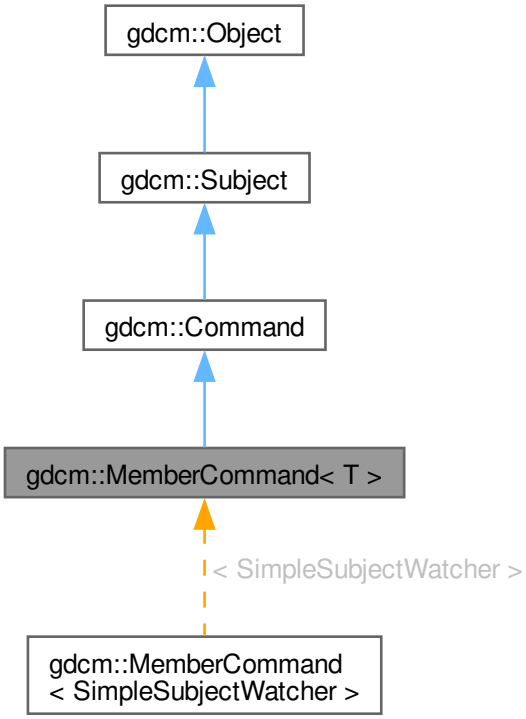
- gdcmMediaStorage.h

12.190 gdcm::MemberCommand< T > Class Template Reference

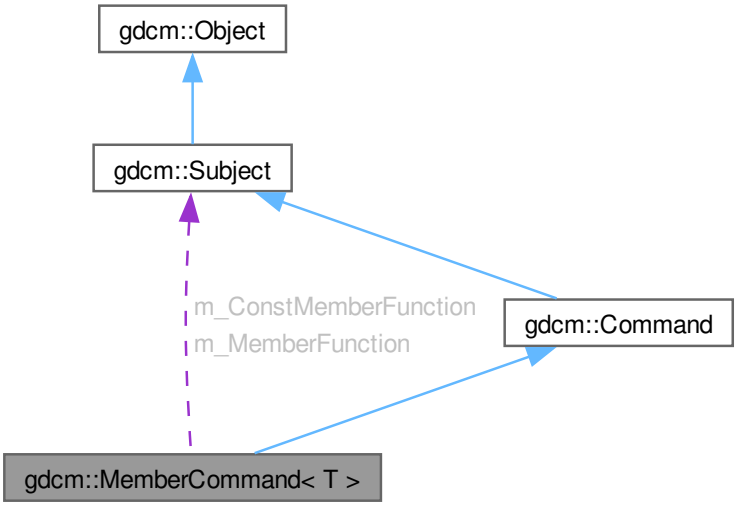
Command subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::MemberCommand< T >:



Collaboration diagram for gdcm::MemberCommand< T >:



Public Types

- typedef MemberCommand Self
- typedef void(T::* TConstMemberFunctionPointer) (const Subject *, const Event &)
- typedef void(T::* TMemberFunctionPointer) (Subject *, const Event &)

Public Member Functions

- MemberCommand (const Self &)=delete
- void Execute (const Subject *caller, const Event &event) override
- void Execute (Subject *caller, const Event &event) override
- void operator= (const Self &)=delete
- void SetCallbackFunction (T *object, TConstMemberFunctionPointer memberFunction)
- void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)

Public Member Functions inherited from gdcm::Command

- Command (const Command &)=delete
- void operator= (const Command &)=delete

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static SmartPointer< MemberCommand > New ()

Protected Member Functions

- MemberCommand ()
- ~MemberCommand () override=default

Protected Member Functions inherited from gdcm::Command

- Command ()
- ~Command () override

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes

- TConstMemberFunctionPointer m_ConstMemberFunction
- TMemberFunctionPointer m_MemberFunction
- T * m_This

12.190.1 Detailed Description

template<class T>
class gdcm::MemberCommand< T >

Command subclass that calls a pointer to a member function.

MemberCommand calls a pointer to a member function with the same arguments as Execute on Command.

12.190.2 Member Typedef Documentation

12.190.2.1 Self

```
template<class T>
typedef MemberCommand gdcm::MemberCommand< T >::Self
```

Standard class typedefs.

12.190.2.2 TConstMemberFunctionPointer

```
template<class T>
typedef void(T::* gdcm::MemberCommand< T >::TConstMemberFunctionPointer) (const Subject *, const
Event &)
```

12.190.2.3 TMemberFunctionPointer

```
template<class T>
typedef void(T::* gdcm::MemberCommand< T >::TMemberFunctionPointer) (Subject *, const Event &)
```

pointer to a member function that takes a Subject∗ and the event

12.190.3 Constructor & Destructor Documentation

12.190.3.1 MemberCommand() [1/2]

```
template<class T>
gdcm::MemberCommand< T >::MemberCommand (
            const Self & )  [delete]
```

12.190.3.2 MemberCommand() [2/2]

```
template<class T>
gdcm::MemberCommand< T >::MemberCommand ()  [inline], [protected]
```

12.190.3.3 ~MemberCommand()

```
template<class T>
gdcm::MemberCommand< T >::~MemberCommand ()  [override], [protected], [default]
```

12.190.4 Member Function Documentation

12.190.4.1 Execute() [1/2]

```
template<class T>
void gdcm::MemberCommand< T >::Execute (
            const Subject * caller,
            const Event & event)  [inline], [override], [virtual]
```

Invoke the member function with a const object.

Implements gdcm::Command.

12.190.4.2 Execute() [2/2]

```
template<class T>
void gdcm::MemberCommand< T >::Execute (
            Subject * caller,
            const Event & event)  [inline], [override], [virtual]
```

Invoke the member function.

Implements gdcm::Command.

12.190.4.3 New()

```
template<class T>
SmartPointer< MemberCommand > gdcm::MemberCommand< T >::New ()  [inline], [static]
```

Method for creation through the object factory.

12.190.4.4 operator=()

```
template<class T>
void gdcm::MemberCommand< T >::operator= (
            const Self & )  [delete]
```

12.190.4.5 SetCallbackFunction() [1/2]

```
template<class T>
void gdcm::MemberCommand< T >::SetCallbackFunction (
            T * object,
            TConstMemberFunctionPointer memberFunction)  [inline]
```

12.190.4.6 SetCallbackFunction() [2/2]

```
template<class T>
void gdcm::MemberCommand< T >::SetCallbackFunction (
            T * object,
            TMemberFunctionPointer memberFunction)  [inline]
```

Run-time type information (and related methods). Set the callback function along with the object that it will be invoked on.

12.190.5 Member Data Documentation

12.190.5.1 m_ConstMemberFunction

```
template<class T>
TConstMemberFunctionPointer gdcm::MemberCommand< T >::m_ConstMemberFunction  [protected]
```

12.190.5.2 m_MemberFunction

```
template<class T>
TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction  [protected]
```

12.190.5.3 m_This

```
template<class T>
T* gdcm::MemberCommand< T >::m_This  [protected]
```

The documentation for this class was generated from the following file:

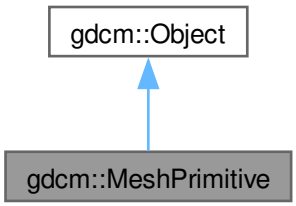
- gdcmCommand.h

12.191 gdcm::MeshPrimitive Class Reference

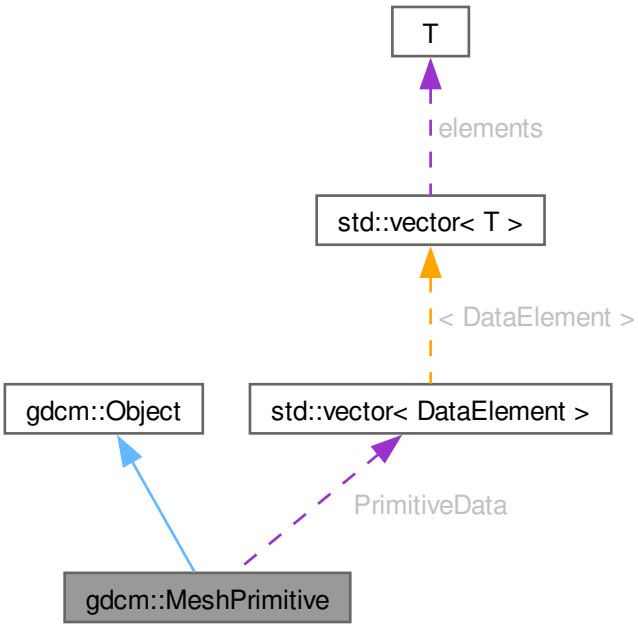
This class defines surface mesh primitives.

```
#include <gdcmMeshPrimitive.h>
```

Inheritance diagram for gdcm::MeshPrimitive:



Collaboration diagram for gdcm::MeshPrimitive:



Public Types

- enum MPType {
  VERTEX = 0 ,
  EDGE ,
  TRIANGLE ,
  TRIANGLE_STRIP ,
  TRIANGLE_FAN ,
  LINE ,
  FACET ,
  MPType_END }

  This enumeration defines primitive types.
- typedef std::vector< DataElement > PrimitivesData

Public Member Functions

- MeshPrimitive ()
- ~MeshPrimitive () override
- void AddPrimitiveData (DataElement const &de)
- unsigned int GetNumberOfPrimitivesData () const
- DataElement & GetPrimitiveData ()
- const DataElement & GetPrimitiveData () const
- DataElement & GetPrimitiveData (const unsigned int idx)
- const DataElement & GetPrimitiveData (const unsigned int idx) const
- PrimitivesData & GetPrimitivesData ()
- const PrimitivesData & GetPrimitivesData () const
- MPType GetPrimitiveType () const
- void SetPrimitiveData (const unsigned int idx, DataElement const &de)
- void SetPrimitiveData (DataElement const &de)
- void SetPrimitivesData (PrimitivesData const &DEs)
- void SetPrimitiveType (const MPType type)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static MPType GetMPType (const char *type)
- static const char * GetMPTypeString (const MPType type)

Protected Attributes

- PrimitivesData PrimitiveData
- MPType PrimitiveType

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.191.1 Detailed Description

This class defines surface mesh primitives.

It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

12.191.2 Member Typedef Documentation

12.191.2.1 PrimitivesData

```
typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData
```

12.191.3 Member Enumeration Documentation

12.191.3.1 MPType

```
enum gdcm::MeshPrimitive::MPType
```

This enumeration defines primitive types.

See also

PS 3.3 C.27.4.1

Enumerator

VERTEX	
EDGE	
TRIANGLE	
TRIANGLE_STRIP	
TRIANGLE_FAN	
LINE	
FACET	
MPType_END	

12.191.4 Constructor & Destructor Documentation

12.191.4.1 MeshPrimitive()

```
gdcm::MeshPrimitive::MeshPrimitive ()
```

12.191.4.2 ~MeshPrimitive()

```
gdcm::MeshPrimitive::~MeshPrimitive ()  [override]
```

12.191.5 Member Function Documentation

12.191.5.1 AddPrimitiveData()

```
void gdcm::MeshPrimitive::AddPrimitiveData (
            DataElement const & de)
```

12.191.5.2 GetMPType()

```
MPType gdcm::MeshPrimitive::GetMPType (
            const char * type)  [static]
```

12.191.5.3 GetMPTypeString()

```
const char * gdcm::MeshPrimitive::GetMPTypeString (
            const MPType type)  [static]
```

12.191.5.4 GetNumberOfPrimitivesData()

```
unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData () const
```

12.191.5.5 GetPrimitiveData() [1/4]

```
DataElement & gdcm::MeshPrimitive::GetPrimitiveData ()
```

12.191.5.6 GetPrimitiveData() [2/4]

```
const DataElement & gdcm::MeshPrimitive::GetPrimitiveData () const
```

12.191.5.7 GetPrimitiveData() [3/4]

```
DataElement & gdcm::MeshPrimitive::GetPrimitiveData (
            const unsigned int idx)
```

12.191.5.8 GetPrimitiveData() [4/4]

```
const DataElement & gdcm::MeshPrimitive::GetPrimitiveData (
            const unsigned int idx) const
```

12.191.5.9 GetPrimitivesData() [1/2]

```
PrimitivesData & gdcm::MeshPrimitive::GetPrimitivesData ()
```

12.191.5.10 GetPrimitivesData() [2/2]

```
const PrimitivesData & gdcm::MeshPrimitive::GetPrimitivesData () const
```

12.191.5.11 GetPrimitiveType()

```
MPType gdcm::MeshPrimitive::GetPrimitiveType () const
```

12.191.5.12 SetPrimitiveData() [1/2]

```
void gdcm::MeshPrimitive::SetPrimitiveData (
            const unsigned int idx,
            DataElement const & de)
```

12.191.5.13 SetPrimitiveData() [2/2]

```
void gdcm::MeshPrimitive::SetPrimitiveData (
            DataElement const & de)
```

12.191.5.14 SetPrimitivesData()

```
void gdcm::MeshPrimitive::SetPrimitivesData (
            PrimitivesData const & DEs)
```

12.191.5.15 SetPrimitiveType()

```
void gdcm::MeshPrimitive::SetPrimitiveType (
            const MPType type)
```

12.191.6 Member Data Documentation

12.191.6.1 PrimitiveData

```
PrimitivesData gdcm::MeshPrimitive::PrimitiveData  [protected]
```

12.191.6.2 PrimitiveType

```
MPType gdcm::MeshPrimitive::PrimitiveType  [protected]
```

The documentation for this class was generated from the following file:

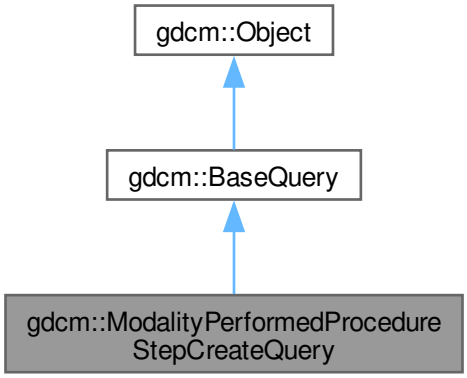
- gdcmMeshPrimitive.h

12.192 gdcm::ModalityPerformedProcedureStepCreateQuery Class Reference

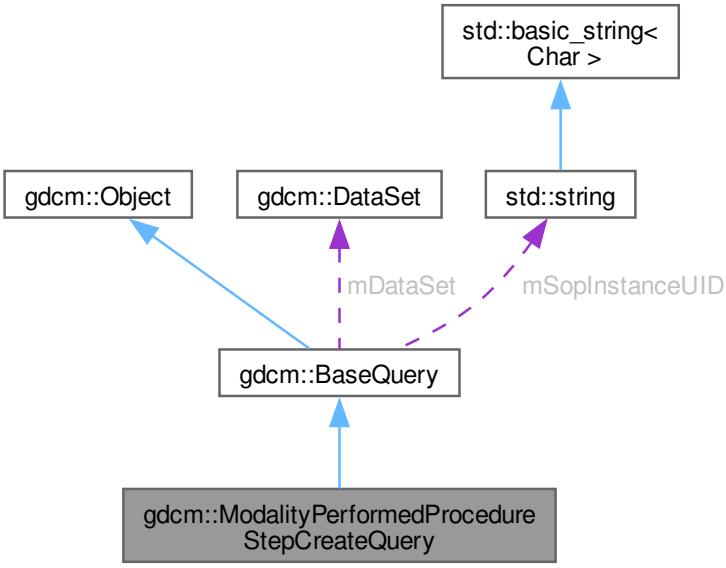
ModalityPerformedProcedureStepCreateQuery.

```
#include <gdcmModalityPerformedProcedureStepCreateQuery.h>
```

Inheritance diagram for gdcm::ModalityPerformedProcedureStepCreateQuery:



Collaboration diagram for gdcm::ModalityPerformedProcedureStepCreateQuery:



Public Member Functions

- ModalityPerformedProcedureStepCreateQuery (const std::string &iSopInstanceUID)

- UIDs::TSName GetAbstractSyntaxUID () const override
- gdcm::DataSet GetRequiredDataSet () const
- bool ValidateQuery (bool inStrict=true) const override

Public Member Functions inherited from gdcm::BaseQuery

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Friends

- class QueryFactory

Additional Inherited Members

Protected Member Functions inherited from gdcm::BaseQuery

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes inherited from gdcm::BaseQuery

- DataSet mDataSet
- std::string mSopInstanceUID

12.192.1 Detailed Description

ModalityPerformedProcedureStepCreateQuery.

contains: the class which will produce a dataset for n-create for Modality Performed Procedure Step sop class

12.192.2 Constructor & Destructor Documentation

12.192.2.1 ModalityPerformedProcedureStepCreateQuery()

```
gdcm::ModalityPerformedProcedureStepCreateQuery::ModalityPerformedProcedureStepCreateQuery (
            const std::string & iSopInstanceUID)
```

12.192.3 Member Function Documentation

12.192.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepCreateQuery::GetAbstractSyntaxUID () const  [override],
[virtual]
```

Implements gdcm::BaseQuery.

12.192.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepCreateQuery::GetRequiredDataSet () const
```

12.192.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepCreateQuery::ValidateQuery (
            bool inStrict = true) const  [override], [virtual]
```

Implements gdcm::BaseQuery.

12.192.4 Friends And Related Symbol Documentation

12.192.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References QueryFactory.

Referenced by QueryFactory.

The documentation for this class was generated from the following file:

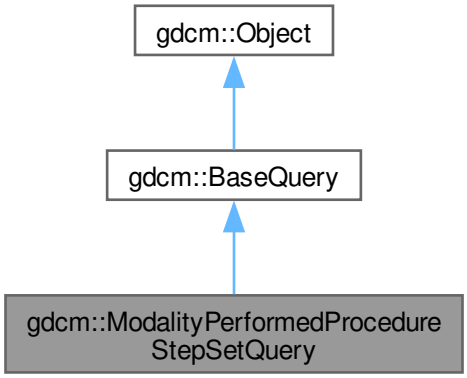
- gdcmModalityPerformedProcedureStepCreateQuery.h

12.193 gdcm::ModalityPerformedProcedureStepSetQuery Class Reference

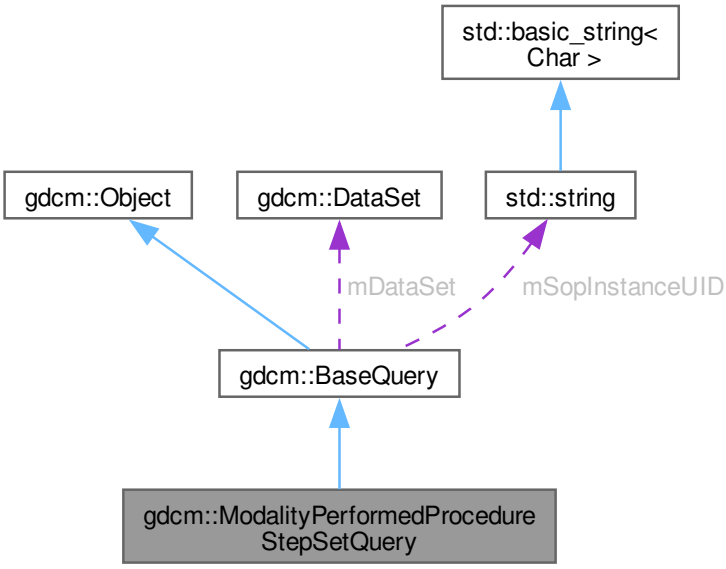
ModalityPerformedProcedureStepSetQuery.

```
#include <gdcmModalityPerformedProcedureStepSetQuery.h>
```

Inheritance diagram for gdcm::ModalityPerformedProcedureStepSetQuery:



Collaboration diagram for gdcm::ModalityPerformedProcedureStepSetQuery:



Public Member Functions

- ModalityPerformedProcedureStepSetQuery (const std::string &iSopInstanceUID)
- UIDs::TSName GetAbstractSyntaxUID () const override
- gdcm::DataSet GetRequiredDataSet () const
- bool ValidateQuery (bool inStrict=true) const override

Public Member Functions inherited from gdcm::BaseQuery

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Friends

- class QueryFactory

Additional Inherited Members

Protected Member Functions inherited from gdcm::BaseQuery

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes inherited from gdcm::BaseQuery

- DataSet mDataSet
- std::string mSopInstanceUID

12.193.1 Detailed Description

ModalityPerformedProcedureStepSetQuery.

contains: the class which will produce a dataset for n-set for Modality Performed Procedure Step sop class

12.193.2 Constructor & Destructor Documentation

12.193.2.1 ModalityPerformedProcedureStepSetQuery()

```
gdcm::ModalityPerformedProcedureStepSetQuery::ModalityPerformedProcedureStepSetQuery (
            const std::string & iSopInstanceUID)
```

12.193.3 Member Function Documentation

12.193.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::ModalityPerformedProcedureStepSetQuery::GetAbstractSyntaxUID () const [override],
[virtual]
```

Implements gdcm::BaseQuery.

12.193.3.2 GetRequiredDataSet()

```
gdcm::DataSet gdcm::ModalityPerformedProcedureStepSetQuery::GetRequiredDataSet () const
```

12.193.3.3 ValidateQuery()

```
bool gdcm::ModalityPerformedProcedureStepSetQuery::ValidateQuery (
            bool inStrict = true) const [override], [virtual]
```

Implements gdcm::BaseQuery.

12.193.4 Friends And Related Symbol Documentation

12.193.4.1 QueryFactory

```
friend class QueryFactory [friend]
```

References QueryFactory.

Referenced by QueryFactory.

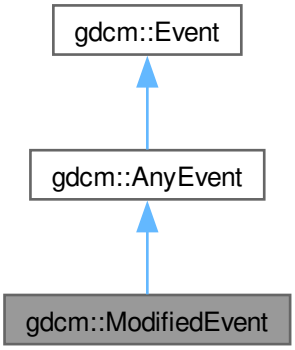
The documentation for this class was generated from the following file:

- gdcmModalityPerformedProcedureStepSetQuery.h

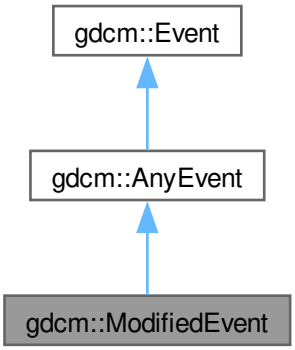
12.194 gdcm::ModifiedEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::ModifiedEvent:



Collaboration diagram for gdcm::ModifiedEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()

- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

- gdcmEvent.h

12.195 gdcm::Module Class Reference

Class for representing a Module.

```
#include <gdcmModule.h>
```

Public Types

- typedef std::vector< std::string > ArrayIncludeMacrosType
- typedef std::map< Tag, ModuleEntry > MapModuleEntry

Public Member Functions

- Module ()=default
- void AddMacro (const char *include)
- void AddModuleEntry (const Tag &tag, const ModuleEntry &module)

  Will add a ModuleEntry directly at root-level. See Macro for nested-included level.
- void Clear ()
- bool FindModuleEntryInMacros (Macros const &macros, const Tag &tag) const
- const ModuleEntry & GetModuleEntryInMacros (Macros const &macros, const Tag &tag) const
- const char * GetName () const
- void SetName (const char *name)
- bool Verify (const DataSet &ds, Usage const &usage) const

Friends

- std::ostream & operator<< (std::ostream &_os, const Module &_val)

12.195.1 Detailed Description

Class for representing a Module.

Note

Module: A set of Attributes within an Information Entity or Normalized IOD which are logically related to each other.

See also

Macro

Examples

TraverseModules.cxx.

12.195.2 Member Typedef Documentation

12.195.2.1 ArrayIncludeMacrosType

```
typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacrosType
```

12.195.2.2 MapModuleEntry

```
typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry
```

12.195.3 Constructor & Destructor Documentation

12.195.3.1 Module()

```
gdcm::Module::Module () [default]
```

References Module(), and operator<<.

Referenced by Module(), and operator<<.

12.195.4 Member Function Documentation

12.195.4.1 AddMacro()

```
void gdcm::Module::AddMacro (
            const char * include) [inline]
```

12.195.4.2 AddModuleEntry()

```
void gdcm::Module::AddModuleEntry (
            const Tag & tag,
            const ModuleEntry & module)  [inline]
```

Will add a ModuleEntry directly at root-level. See Macro for nested-included level.

12.195.4.3 Clear()

```
void gdcm::Module::Clear ()  [inline]
```

12.195.4.4 FindModuleEntryInMacros()

```
bool gdcm::Module::FindModuleEntryInMacros (
            Macros const & macros,
            const Tag & tag) const
```

Find or Get a ModuleEntry. ModuleEntry are either search are root-level or within nested-macro included in module.

Examples

TraverseModules.cxx.

12.195.4.5 GetModuleEntryInMacros()

```
const ModuleEntry & gdcm::Module::GetModuleEntryInMacros (
            Macros const & macros,
            const Tag & tag) const
```

12.195.4.6 GetName()

```
const char * gdcm::Module::GetName () const  [inline]
```

12.195.4.7 SetName()

```
void gdcm::Module::SetName (
            const char * name)  [inline]
```

12.195.4.8 Verify()

```
bool gdcm::Module::Verify (
            const DataSet & ds,
            Usage const & usage) const
```

12.195.5 Friends And Related Symbol Documentation

12.195.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Module & _val)  [friend]
```

References Module().

Referenced by Module().

The documentation for this class was generated from the following file:

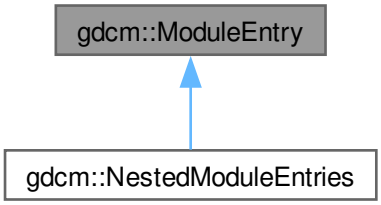
- gdcmModule.h

12.196 gdcm::ModuleEntry Class Reference

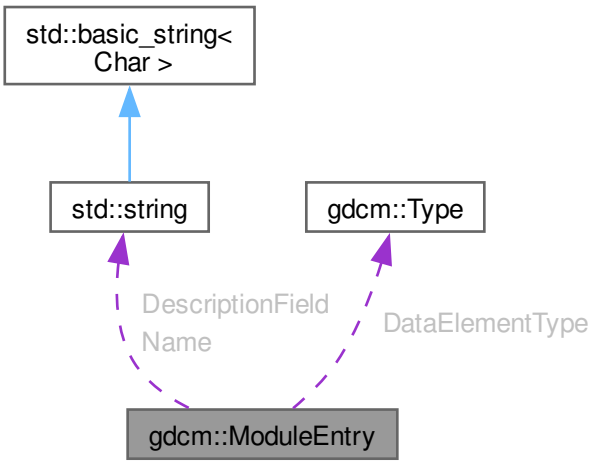
Class for representing a ModuleEntry.

```
#include <gdcmModuleEntry.h>
```

Inheritance diagram for gdcm::ModuleEntry:



Collaboration diagram for gdcm::ModuleEntry:



Public Types

- typedef std::string Description

Public Member Functions

- ModuleEntry (const char *name="", const char *type="3", const char *description="")
- virtual ~ModuleEntry ()=default
- const Description & GetDescription () const
- const char * GetName () const
- const Type & GetType () const
- void SetDescription (const char *d)
- void SetName (const char *name)
- void SetType (const Type &type)

Protected Attributes

- Type DataElementType
- Description DescriptionField
- std::string Name

Friends

- std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)

12.196.1 Detailed Description

Class for representing a ModuleEntry.

Note

bla

See also

DictEntry

Examples

TraverseModules.cxx.

12.196.2 Member Typedef Documentation

12.196.2.1 Description

```
typedef std::string gdcm::ModuleEntry::Description
```

12.196.3 Constructor & Destructor Documentation

12.196.3.1 ModuleEntry()

```
gdcm::ModuleEntry::ModuleEntry (
            const char * name = "",
            const char * type = "3",
            const char * description = "")  [inline]
```

References DataElementType, DescriptionField, gdcm::Type::GetTypeType(), and Name.

Referenced by gdcm::NestedModuleEntries::NestedModuleEntries(), ~ModuleEntry(), gdcm::NestedModuleEntries::AddModuleEntry(), gdcm::NestedModuleEntries::GetModuleEntry(), gdcm::NestedModuleEntries::GetModuleEntry(), and operator<<.

12.196.3.2 ~ModuleEntry()

```
virtual gdcm::ModuleEntry::~ModuleEntry ()  [virtual], [default]
```

References ModuleEntry(), and operator<<.

12.196.4 Member Function Documentation

12.196.4.1 GetDescription()

```
const Description & gdcm::ModuleEntry::GetDescription () const  [inline]
```

References DescriptionField.

12.196.4.2 GetName()

```
const char * gdcm::ModuleEntry::GetName () const  [inline]
```

References Name.

12.196.4.3 GetType()

```
const Type & gdcm::ModuleEntry::GetType () const  [inline]
```

Examples

TraverseModules.cxx.

References DataElementType.

12.196.4.4 SetDescription()

```
void gdcm::ModuleEntry::SetDescription (
            const char * d)  [inline]
```

References DescriptionField.

12.196.4.5 SetName()

```
void gdcm::ModuleEntry::SetName (
            const char * name)  [inline]
```

References Name.

12.196.4.6 SetType()

```
void gdcm::ModuleEntry::SetType (
            const Type & type)  [inline]
```

References DataElementType.

12.196.5 Friends And Related Symbol Documentation

12.196.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const ModuleEntry & _val)  [friend]
```

References ModuleEntry(), DataElementType, DescriptionField, and Name.

Referenced by ~ModuleEntry().

12.196.6 Member Data Documentation

12.196.6.1 DataElementType

```
Type gdcm::ModuleEntry::DataElementType  [protected]
```

Referenced by ModuleEntry(), GetType(), operator<<, gdcm::NestedModuleEntries::operator<<, and SetType().

12.196.6.2 DescriptionField

```
Description gdcm::ModuleEntry::DescriptionField  [protected]
```

Referenced by ModuleEntry(), GetDescription(), operator<<, gdcm::NestedModuleEntries::operator<<, and SetDescription().

12.196.6.3 Name

```
std::string gdcm::ModuleEntry::Name  [protected]
```

Referenced by ModuleEntry(), GetName(), operator<<, gdcm::NestedModuleEntries::operator<<, and SetName().

The documentation for this class was generated from the following file:

- gdcmModuleEntry.h

12.197 gdcm::Modules Class Reference

Class for representing a Modules.

```
#include <gdcmModules.h>
```

Public Types

- typedef std::map< std::string, Module > ModuleMapType

Public Member Functions

- Modules ()=default
- void AddModule (const char *ref, const Module &module)
- void Clear ()
- const Module & GetModule (const char *name) const
- bool IsEmpty () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Modules &_val)

12.197.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples

TraverseModules.cxx.

12.197.2 Member Typedef Documentation

12.197.2.1 ModuleMapType

```
typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType
```

12.197.3 Constructor & Destructor Documentation

12.197.3.1 Modules()

```
gdcm::Modules::Modules () [default]
```

References Modules(), and operator<<.

Referenced by Modules(), and operator<<.

12.197.4 Member Function Documentation

12.197.4.1 AddModule()

```
void gdcm::Modules::AddModule (
            const char * ref,
            const Module & module)  [inline]
```

12.197.4.2 Clear()

```
void gdcm::Modules::Clear ()  [inline]
```

12.197.4.3 GetModule()

```
const Module & gdcm::Modules::GetModule (
            const char * name) const  [inline]
```

Examples

TraverseModules.cxx.

12.197.4.4 IsEmpty()

```
bool gdcm::Modules::IsEmpty () const  [inline]
```

12.197.5 Friends And Related Symbol Documentation

12.197.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Modules & _val)  [friend]
```

References Modules().

Referenced by Modules().

The documentation for this class was generated from the following file:

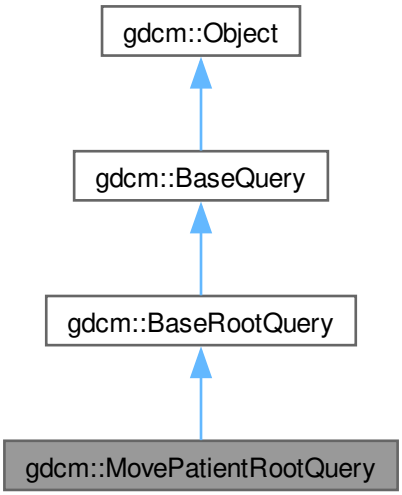
- gdcmModules.h

12.198 gdcm::MovePatientRootQuery Class Reference

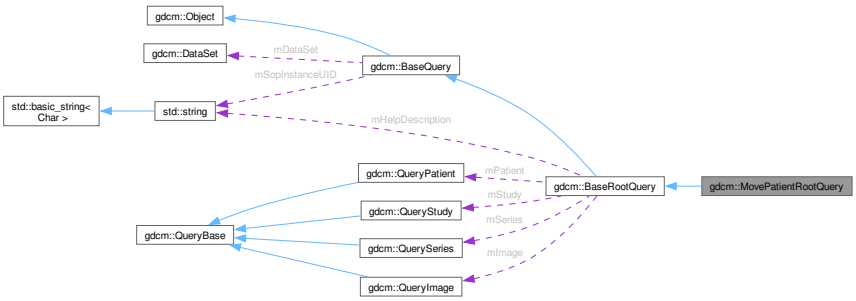
MovePatientRootQuery.

```
#include <gdcmMovePatientRootQuery.h>
```

Inheritance diagram for gdcm::MovePatientRootQuery:



Collaboration diagram for gdcm::MovePatientRootQuery:



Public Member Functions

- MovePatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Public Member Functions inherited from gdcm::BaseRootQuery

- ~BaseRootQuery () override=default
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)

Public Member Functions inherited from gdcm::BaseQuery

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Friends

- class QueryFactory

Additional Inherited Members

Static Public Member Functions inherited from gdcm::BaseRootQuery

- static QueryBase * Construct (ERootType inRootType, EQueryLevel qlevel)
- static int GetQueryLevelFromString (const char *str)
- static const char * GetQueryLevelString (EQueryLevel ql)

Protected Member Functions inherited from gdcm::BaseRootQuery

- BaseRootQuery ()

Protected Member Functions inherited from gdcm::BaseQuery

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes inherited from gdcm::BaseRootQuery

- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Protected Attributes inherited from gdcm::BaseQuery

- DataSet mDataSet
- std::string mSopInstanceUID

12.198.1 Detailed Description

MovePatientRootQuery.

contains: the class which will produce a dataset for c-move with patient root

12.198.2 Constructor & Destructor Documentation

12.198.2.1 MovePatientRootQuery()

```
gdcm::MovePatientRootQuery::MovePatientRootQuery ()
```

12.198.3 Member Function Documentation

12.198.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID () const  [override], [virtual]
```

Implements gdcm::BaseQuery.

12.198.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::MovePatientRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

12.198.3.3 InitializeDataSet()

```
void gdcm::MovePatientRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

12.198.3.4 ValidateQuery()

```
bool gdcm::MovePatientRootQuery::ValidateQuery (
            bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

12.198.4 Friends And Related Symbol Documentation

12.198.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References QueryFactory.

Referenced by QueryFactory.

The documentation for this class was generated from the following file:

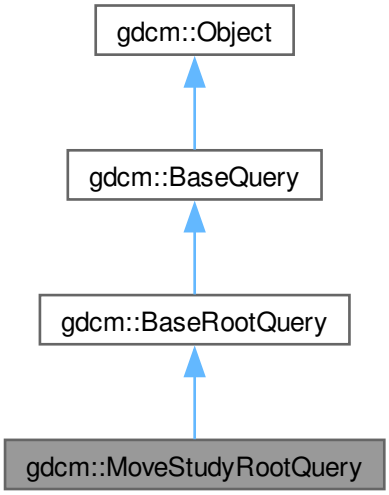
- gdcmMovePatientRootQuery.h

12.199 gdcm::MoveStudyRootQuery Class Reference

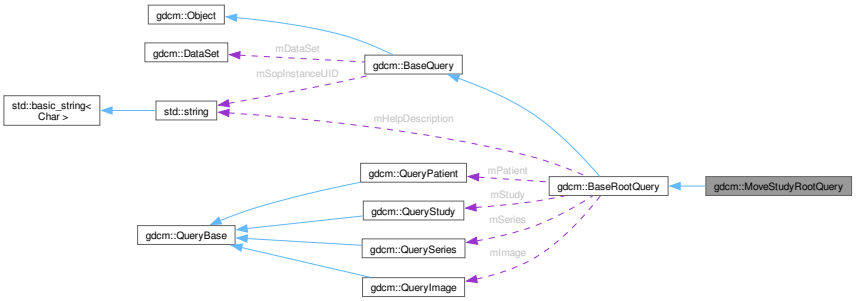
MoveStudyRootQuery.

```
#include <gdcmMoveStudyRootQuery.h>
```

Inheritance diagram for gdcm::MoveStudyRootQuery:



Collaboration diagram for gdcm::MoveStudyRootQuery:



Public Member Functions

- MoveStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Public Member Functions inherited from gdcm::BaseRootQuery

- ~BaseRootQuery () override=default
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)

Public Member Functions inherited from gdcm::BaseQuery

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Friends

- class QueryFactory

Additional Inherited Members

Static Public Member Functions inherited from gdcm::BaseRootQuery

- static QueryBase * Construct (ERootType inRootType, EQueryLevel qlevel)
- static int GetQueryLevelFromString (const char *str)
- static const char * GetQueryLevelString (EQueryLevel ql)

Protected Member Functions inherited from gdcm::BaseRootQuery

- BaseRootQuery ()

Protected Member Functions inherited from gdcm::BaseQuery

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes inherited from gdcm::BaseRootQuery

- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Protected Attributes inherited from gdcm::BaseQuery

- DataSet mDataSet
- std::string mSopInstanceUID

12.199.1 Detailed Description

MoveStudyRootQuery.

contains: the class which will produce a dataset for C-MOVE with study root

12.199.2 Constructor & Destructor Documentation

12.199.2.1 MoveStudyRootQuery()

```
gdcm::MoveStudyRootQuery::MoveStudyRootQuery ()
```

12.199.3 Member Function Documentation

12.199.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID () const  [override], [virtual]
```

Implements gdcm::BaseQuery.

12.199.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::MoveStudyRootQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

12.199.3.3 InitializeDataSet()

```
void gdcm::MoveStudyRootQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

12.199.3.4 ValidateQuery()

```
bool gdcm::MoveStudyRootQuery::ValidateQuery (
            bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

12.199.4 Friends And Related Symbol Documentation

12.199.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References QueryFactory.

Referenced by QueryFactory.

The documentation for this class was generated from the following file:

- gdcmMoveStudyRootQuery.h

12.200 gdcm::MrProtocol Class Reference

Class for MrProtocol.

```
#include <gdcmMrProtocol.h>
```

Classes

- struct Slice
- struct SliceArray
- struct Vector3

Public Member Functions

- MrProtocol ()
- ~MrProtocol ()
- bool FindMrProtocolByName (const char *name) const
- const char * GetMrProtocolByName (const char *name) const
- bool GetSliceArray (MrProtocol::SliceArray &sa) const
- int GetVersion () const
- bool Load (const ByteValue *bv, const char *str, int version)
- void Print (std::ostream &os) const

Friends

- std::ostream & operator<< (std::ostream &_os, const MrProtocol &d)

12.200.1 Detailed Description

Class for MrProtocol.

Examples

MrProtocol.cxx.

12.200.2 Constructor & Destructor Documentation

12.200.2.1 MrProtocol()

```
gdcm::MrProtocol::MrProtocol ()
```

Referenced by operator<<.

12.200.2.2 ~MrProtocol()

```
gdcm::MrProtocol::~MrProtocol ()
```

12.200.3 Member Function Documentation

12.200.3.1 FindMrProtocolByName()

```
bool gdcm::MrProtocol::FindMrProtocolByName (
            const char * name) const
```

12.200.3.2 GetMrProtocolByName()

```
const char * gdcm::MrProtocol::GetMrProtocolByName (
            const char * name) const
```

12.200.3.3 GetSliceArray()

```
bool gdcm::MrProtocol::GetSliceArray (
            MrProtocol::SliceArray & sa) const
```

12.200.3.4 GetVersion()

```
int gdcm::MrProtocol::GetVersion () const
```

12.200.3.5 Load()

```
bool gdcm::MrProtocol::Load (
            const ByteValue * bv,
            const char * str,
            int version)
```

12.200.3.6 Print()

```
void gdcm::MrProtocol::Print (
            std::ostream & os) const
```

Referenced by operator<<.

12.200.4 Friends And Related Symbol Documentation

12.200.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const MrProtocol & d)  [friend]
```

References MrProtocol(), and Print().

The documentation for this class was generated from the following file:

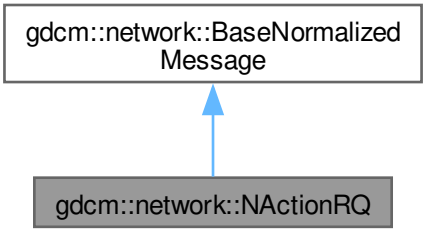
- gdcmMrProtocol.h

12.201 gdcm::network::NActionRQ Class Reference

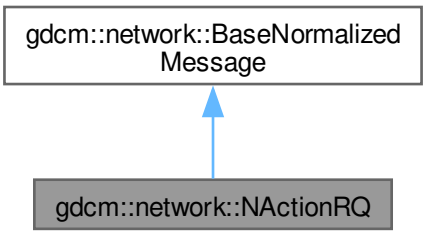
NActionRQ.

```
#include <gdcmNActionMessages.h>
```

Inheritance diagram for gdcm::network::NActionRQ:



Collaboration diagram for gdcm::network::NActionRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default

12.201.1 Detailed Description

NActionRQ.

this file defines the messages for the NAction action

12.201.2 Member Function Documentation

12.201.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NActionRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

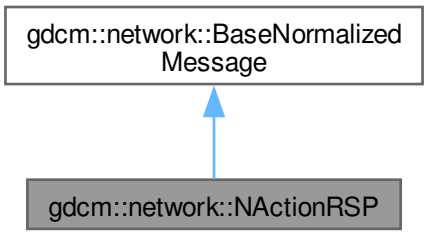
- gdcmNActionMessages.h

12.202 gdcm::network::NActionRSP Class Reference

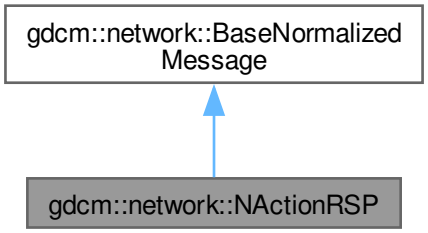
NActionRSP this file defines the messages for the NAction action.

```
#include <gdcmNActionMessages.h>
```

Inheritance diagram for gdcm::network::NActionRSP:



Collaboration diagram for gdcm::network::NActionRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

12.202.1 Detailed Description

NActionRSP this file defines the messages for the NAction action.

12.202.2 Member Function Documentation

12.202.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NActionRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

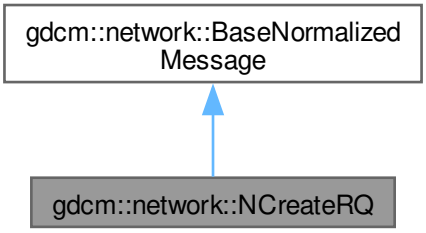
- gdcmNActionMessages.h

12.203 gdcm::network::NCreateRQ Class Reference

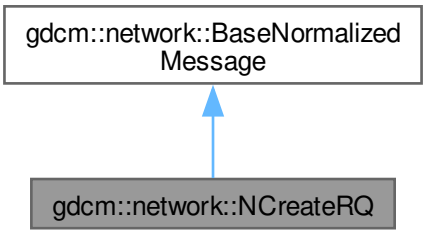
NCreateRQ.

```
#include <gdcmNCreateMessages.h>
```

Inheritance diagram for gdcm::network::NCreateRQ:



Collaboration diagram for gdcm::network::NCreateRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default

12.203.1 Detailed Description

NCreateRQ.

this file defines the messages for the ncreate action

12.203.2 Member Function Documentation

12.203.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NCreateRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

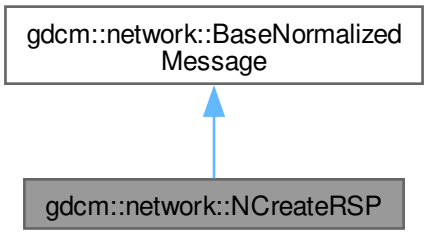
- gdcmNCreateMessages.h

12.204 gdcm::network::NCreateRSP Class Reference

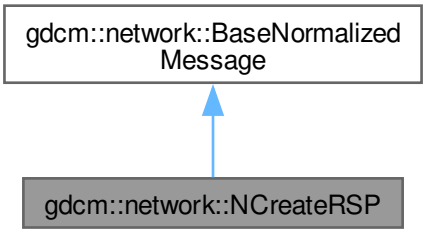
NCreateRSP this file defines the messages for the ncreate action.

```
#include <gdcmNCreateMessages.h>
```

Inheritance diagram for gdcm::network::NCreateRSP:



Collaboration diagram for gdcm::network::NCreateRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

12.204.1 Detailed Description

NCreateRSP this file defines the messages for the ncreate action.

12.204.2 Member Function Documentation

12.204.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NCreateRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

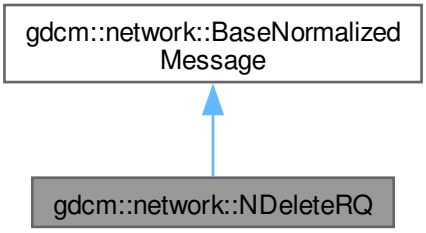
- gdcmNCreateMessages.h

12.205 gdcm::network::NDeleteRQ Class Reference

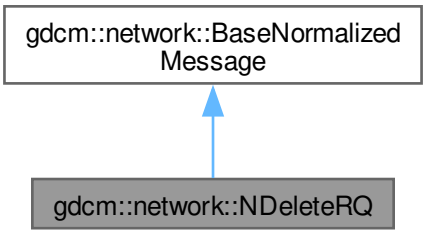
NDeleteRQ.

```
#include <gdcmNDeleteMessages.h>
```

Inheritance diagram for gdcm::network::NDeleteRQ:



Collaboration diagram for gdcm::network::NDeleteRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default

12.205.1 Detailed Description

NDeleteRQ.

this file defines the messages for the ndelete action

12.205.2 Member Function Documentation

12.205.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NDeleteRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

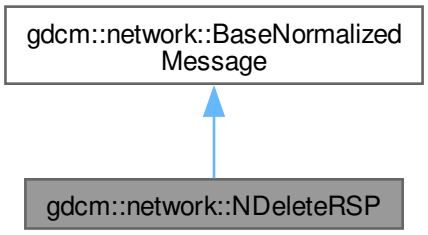
- gdcmNDeleteMessages.h

12.206 gdcm::network::NDeleteRSP Class Reference

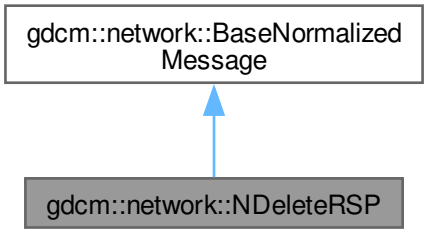
NDeleteRSP this file defines the messages for the ndelete action.

```
#include <gdcmNDeleteMessages.h>
```

Inheritance diagram for gdcm::network::NDeleteRSP:



Collaboration diagram for gdcm::network::NDeleteRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

12.206.1 Detailed Description

NDeleteRSP this file defines the messages for the ndelete action.

12.206.2 Member Function Documentation

12.206.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NDeleteRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

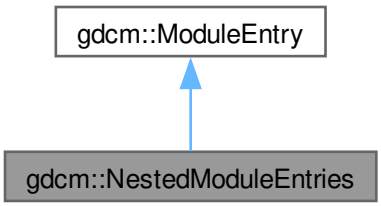
- gdcmNDeleteMessages.h

12.207 gdcm::NestedModuleEntries Class Reference

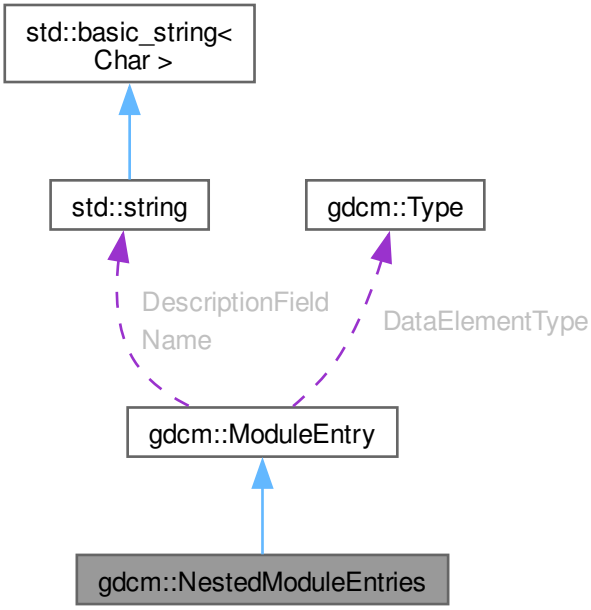
Class for representing a NestedModuleEntries.

```
#include <gdcmNestedModuleEntries.h>
```

Inheritance diagram for gdcm::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



Public Types

- typedef std::vector< ModuleEntry >::size_type SizeType

Public Types inherited from gdcm::ModuleEntry

- typedef std::string Description

Public Member Functions

- NestedModuleEntries (const char *name="", const char *type="3", const char *description="")
- void AddModuleEntry (const ModuleEntry &me)
- ModuleEntry & GetModuleEntry (SizeType idx)
- const ModuleEntry & GetModuleEntry (SizeType idx) const
- SizeType GetNumberOfModuleEntries ()

Public Member Functions inherited from gdcm::ModuleEntry

- ModuleEntry (const char *name="", const char *type="3", const char *description="")
- virtual ~ModuleEntry ()=default
- const Description & GetDescription () const
- const char * GetName () const
- const Type & GetType () const
- void SetDescription (const char *d)
- void SetName (const char *name)
- void SetType (const Type &type)

Friends

- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)

Additional Inherited Members

Protected Attributes inherited from gdcm::ModuleEntry

- Type DataElementType
- Description DescriptionField
- std::string Name

12.207.1 Detailed Description

Class for representing a NestedModuleEntries.

Note

bla

See also

ModuleEntry

12.207.2 Member Typedef Documentation

12.207.2.1 SizeType

```
typedef std::vector<ModuleEntry>::size_type gdcm::NestedModuleEntries::SizeType
```

12.207.3 Constructor & Destructor Documentation

12.207.3.1 NestedModuleEntries()

```
gdcm::NestedModuleEntries::NestedModuleEntries (
            const char * name = "",
            const char * type = "3",
            const char * description = "")  [inline]
```

References gdcm::ModuleEntry::ModuleEntry().

Referenced by operator<<.

12.207.4 Member Function Documentation

12.207.4.1 AddModuleEntry()

```
void gdcm::NestedModuleEntries::AddModuleEntry (
            const ModuleEntry & me)  [inline]
```

References gdcm::ModuleEntry::ModuleEntry().

12.207.4.2 GetModuleEntry() [1/2]

```
ModuleEntry & gdcm::NestedModuleEntries::GetModuleEntry (
            SizeType idx)  [inline]
```

References gdcm::ModuleEntry::ModuleEntry().

12.207.4.3 GetModuleEntry() [2/2]

```
const ModuleEntry & gdcm::NestedModuleEntries::GetModuleEntry (
            SizeType idx) const  [inline]
```

References gdcm::ModuleEntry::ModuleEntry().

12.207.4.4 GetNumberOfModuleEntries()

```
SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ()  [inline]
```

12.207.5 Friends And Related Symbol Documentation

12.207.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const NestedModuleEntries & _val)  [friend]
```

References NestedModuleEntries(), gdcm::ModuleEntry::DataElementType, gdcm::ModuleEntry::DescriptionField, and gdcm::ModuleEntry::Name.

The documentation for this class was generated from the following file:

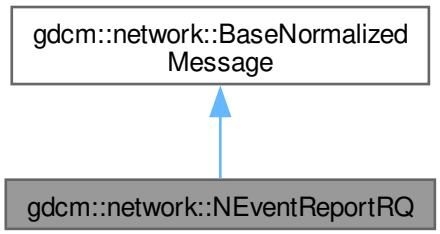
- gdcmNestedModuleEntries.h

12.208 gdcm::network::NEventReportRQ Class Reference

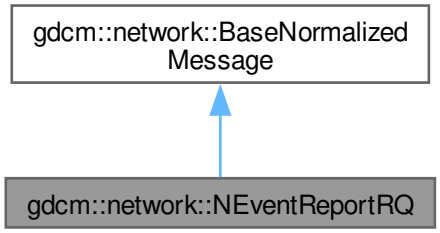
NEventReportRQ.

```
#include <gdcmNEventReportMessages.h>
```

Inheritance diagram for gdcm::network::NEventReportRQ:



Collaboration diagram for gdcm::network::NEventReportRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default

12.208.1 Detailed Description

NEventReportRQ.

this file defines the messages for the neventreport action

12.208.2 Member Function Documentation

12.208.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NEventReportRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

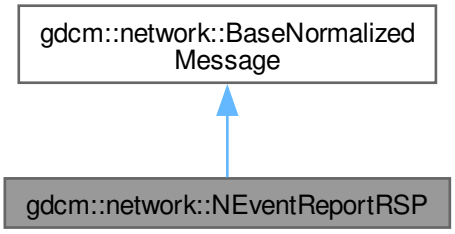
- gdcmNEventReportMessages.h

12.209 gdcm::network::NEventReportRSP Class Reference

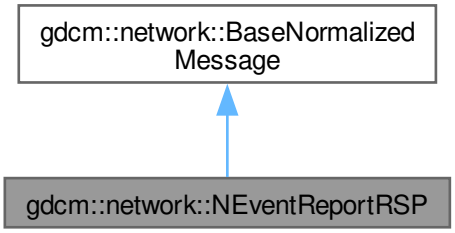
NEventReportRSP this file defines the messages for the neventreport action.

```
#include <gdcmNEventReportMessages.h>
```

Inheritance diagram for gdcm::network::NEventReportRSP:



Collaboration diagram for gdcm::network::NEventReportRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

12.209.1 Detailed Description

NEventReportRSP this file defines the messages for the neventreport action.

12.209.2 Member Function Documentation

12.209.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NEventReportRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

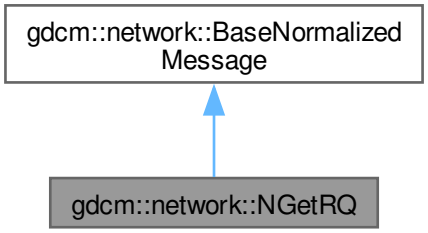
- gdcmNEventReportMessages.h

12.210 gdcm::network::NGetRQ Class Reference

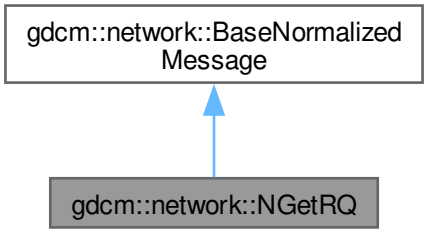
NGetRQ.

```
#include <gdcmNGetMessages.h>
```

Inheritance diagram for gdcm::network::NGetRQ:



Collaboration diagram for gdcm::network::NGetRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default

12.210.1 Detailed Description

NGetRQ.

this file defines the messages for the nget action

12.210.2 Member Function Documentation

12.210.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NGetRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

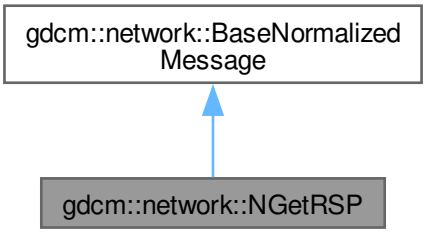
- gdcmNGetMessages.h

12.211 gdcm::network::NGetRSP Class Reference

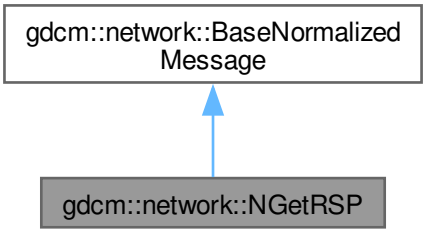
NGetRSP this file defines the messages for the nget action.

```
#include <gdcmNGetMessages.h>
```

Inheritance diagram for gdcm::network::NGetRSP:



Collaboration diagram for gdcm::network::NGetRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

12.211.1 Detailed Description

NGetRSP this file defines the messages for the nget action.

12.211.2 Member Function Documentation

12.211.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NGetRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

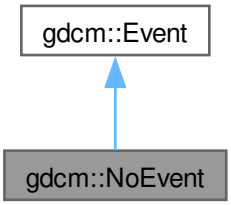
The documentation for this class was generated from the following file:

- gdcmNGetMessages.h

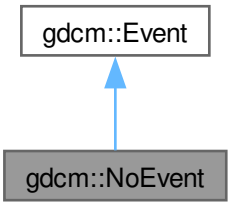
12.212 gdcm::NoEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

12.212.1 Detailed Description

Define some common GDCM events

The documentation for this class was generated from the following file:

- gdcmEvent.h

12.213 gdcm::network::NormalizedMessageFactory Class Reference

```
#include <gdcmNormalizedMessageFactory.h>
```

Static Public Member Functions

- static std::vector< PresentationDataValue > ConstructNAction (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNCreate (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNDelete (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNEventReport (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNGet (const ULConnection &inConnection, const BaseQuery *inQuery)
- static std::vector< PresentationDataValue > ConstructNSet (const ULConnection &inConnection, const BaseQuery *inQuery)

12.213.1 Member Function Documentation

12.213.1.1 ConstructNAction()

```
std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNAction (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.213.1.2 ConstructNCreate()

```
std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNCreate (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.213.1.3 ConstructNDelete()

```
std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNDelete (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.213.1.4 ConstructNEventReport()

```
std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNEvent↩
Report (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.213.1.5 ConstructNGet()

```
std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNGet (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.213.1.6 ConstructNSet()

```
std::vector< PresentationDataValue > gdcm::network::NormalizedMessageFactory::ConstructNSet (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

The documentation for this class was generated from the following file:

- gdcmNormalizedMessageFactory.h

12.214 gdcm::NormalizedNetworkFunctions Class Reference

Normalized Network Functions.

```
#include <gdcmNormalizedNetworkFunctions.h>
```

Static Public Member Functions

- static BaseQuery * ConstructQuery (const std::string &sopInstanceUID, const DataSet &queryds, ENQueryType queryType=eCreateMMPS)
- static bool NAction (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &ret↩ DataSets, const char *aetitle, const char *call)
- static bool NCreate (const char *remote, uint16_t portno, BaseQuery *query, std::vector< DataSet > &retData↩ Sets, const char *aetitle, const char *call)
- static bool NDelete (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &ret↩ DataSets, const char *aetitle, const char *call)
- static bool NEventReport (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle, const char *call)
- static bool NGet (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &ret↩ DataSets, const char *aetitle, const char *call)
- static bool NSet (const char *remote, uint16_t portno, const BaseQuery *query, std::vector< DataSet > &ret↩ DataSets, const char *aetitle, const char *call)

12.214.1 Detailed Description

Normalized Network Functions.

These functions provide a generic API to the DICOM functions implemented in GDCM. Advanced users can use this code as a template for building their own versions of these functions (for instance, to provide progress bars or some other way of handling returned query information), but for most users, these functions should be sufficient to interface with a PACS to a local machine. Note that these functions are not contained within a static class or some other class-style interface, because multiple connections can be instantiated in the same program. The DICOM standard is much more function oriented rather than class oriented in this instance, so the design of this API reflects that functional approach. These functions implements the following SCU operations:

- N-EVENT-REPORT
- N-GET
- N-SET
- N-ACTION
- N-CREATE
- N-DELETE

12.214.2 Member Function Documentation

12.214.2.1 ConstructQuery()

```
BaseQuery * gdcm::NormalizedNetworkFunctions::ConstructQuery (
            const std::string & sopInstanceUID,
            const DataSet & queryds,
            ENQueryType queryType = eCreateMMPS)  [static]
```

References gdcm::eCreateMMPS.

12.214.2.2 NAction()

```
bool gdcm::NormalizedNetworkFunctions::NAction (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call)  [static]
```

12.214.2.3 NCreate()

```
bool gdcm::NormalizedNetworkFunctions::NCreate (
            const char * remote,
            uint16_t portno,
            BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call)  [static]
```

12.214.2.4 NDelete()

```
bool gdcm::NormalizedNetworkFunctions::NDelete (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call)  [static]
```

12.214.2.5 NEventReport()

```
bool gdcm::NormalizedNetworkFunctions::NEventReport (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call)  [static]
```

12.214.2.6 NGet()

```
bool gdcm::NormalizedNetworkFunctions::NGet (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call)  [static]
```

12.214.2.7 NSet()

```
bool gdcm::NormalizedNetworkFunctions::NSet (
            const char * remote,
            uint16_t portno,
            const BaseQuery * query,
            std::vector< DataSet > & retDataSets,
            const char * aetitle,
            const char * call)  [static]
```

The documentation for this class was generated from the following file:

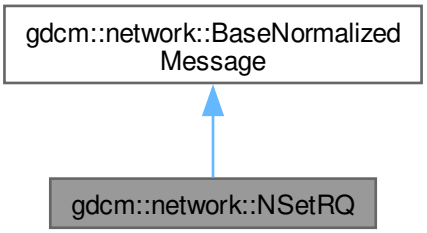
- gdcmNormalizedNetworkFunctions.h

12.215 gdcm::network::NSetRQ Class Reference

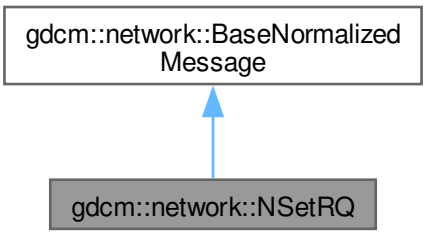
NSetRQ.

```
#include <gdcmNSetMessages.h>
```

Inheritance diagram for gdcm::network::NSetRQ:



Collaboration diagram for gdcm::network::NSetRQ:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery) override

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default

12.215.1 Detailed Description

NSetRQ.

this file defines the messages for the nset action

12.215.2 Member Function Documentation

12.215.2.1 ConstructPDV()

```
std::vector< PresentationDataValue > gdcm::network::NSetRQ::ConstructPDV (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [override], [virtual]
```

Implements gdcm::network::BaseNormalizedMessage.

The documentation for this class was generated from the following file:

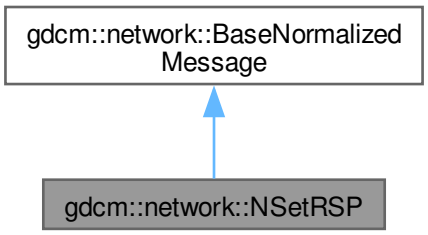
- gdcmNSetMessages.h

12.216 gdcm::network::NSetRSP Class Reference

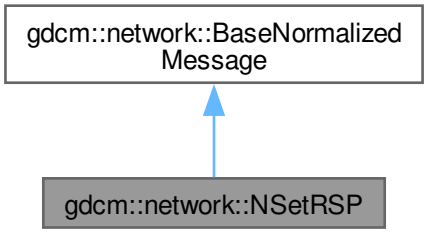
NSetRSP this file defines the messages for the nset action.

```
#include <gdcmNSetMessages.h>
```

Inheritance diagram for gdcm::network::NSetRSP:



Collaboration diagram for gdcm::network::NSetRSP:



Public Member Functions

- std::vector< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)

Public Member Functions inherited from gdcm::network::BaseNormalizedMessage

- virtual ~BaseNormalizedMessage ()=default
- virtual std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseQuery *inQuery)=0

12.216.1 Detailed Description

NSetRSP this file defines the messages for the nset action.

12.216.2 Member Function Documentation

12.216.2.1 ConstructPDVByDataSet()

```
std::vector< PresentationDataValue > gdcm::network::NSetRSP::ConstructPDVByDataSet (
            const DataSet * inDataSet)
```

The documentation for this class was generated from the following file:

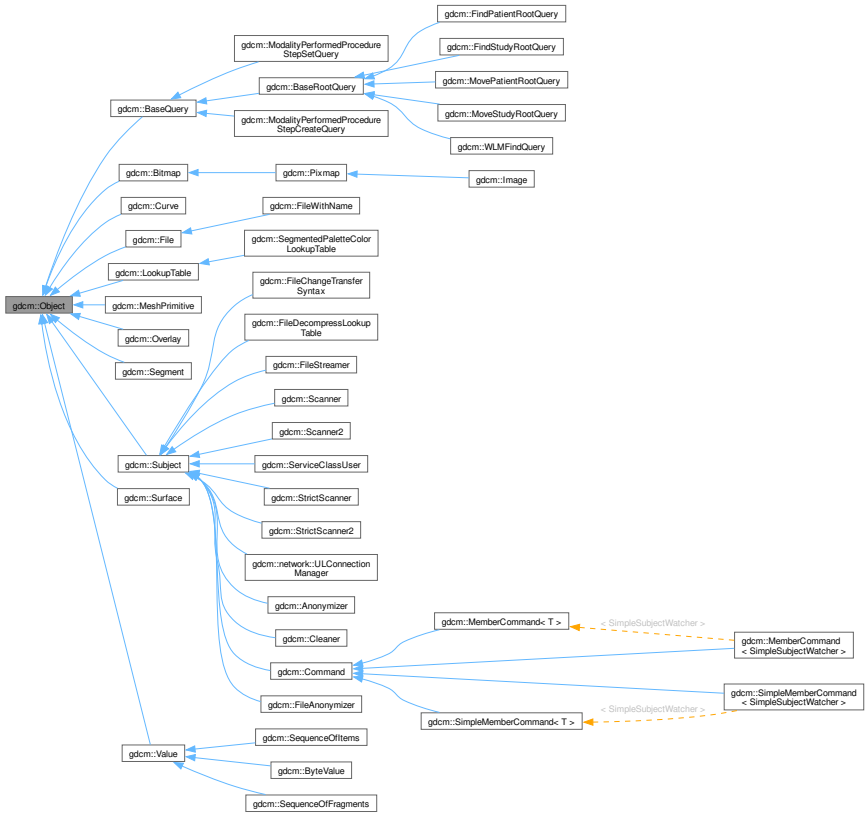
- gdcmNSetMessages.h

12.217 gdcm::Object Class Reference

Object.

```
#include <gdcmObject.h>
```

Inheritance diagram for gdcm::Object:



Public Member Functions

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Protected Member Functions

- void Register ()
- void UnRegister ()

Friends

- std::ostream & operator<< (std::ostream &os, const Object &obj)
- template<class ObjectType>
  class SmartPointer

12.217.1 Detailed Description

Object.

Note

main superclass for object that want to use SmartPointer invasive ref counting system

See also

SmartPointer

12.217.2 Constructor & Destructor Documentation

12.217.2.1 Object() [1/2]

```
gdcm::Object::Object () [inline]
```

Referenced by gdcm::LookupTable::LookupTable(), Object(), operator<<, operator=(), and SmartPointer.

12.217.2.2 ~Object()

```
virtual gdcm::Object::~Object () [inline], [virtual]
```

12.217.2.3 Object() [2/2]

```
gdcm::Object::Object (
            const Object & ) [inline]
```

Special requirement for copy/cstor, assignment operator.

References Object().

12.217.3 Member Function Documentation

12.217.3.1 operator=()

```
void gdcm::Object::operator= (
            const Object & )  [inline]
```

References Object().

12.217.3.2 Print()

```
virtual void gdcm::Object::Print (
            std::ostream & ) const  [inline], [virtual]
```

Reimplemented in gdcm::BaseQuery, gdcm::Bitmap, gdcm::ByteValue, gdcm::Curve, gdcm::Image, gdcm::LookupTable, gdcm::Overlay, gdcm::Pixmap, gdcm::Scanner2, gdcm::Scanner, gdcm::SegmentedPaletteColorLookupTable, gdcm::SequenceOfFragments, gdcm::SequenceOfItems, gdcm::StrictScanner2, and gdcm::StrictScanner.

Examples

ReadAndDumpDICOMDIR.cxx.

Referenced by gdcm::DataElement::operator<<, and operator<<.

12.217.3.3 Register()

```
void gdcm::Object::Register ()  [inline], [protected]
```

12.217.3.4 UnRegister()

```
void gdcm::Object::UnRegister ()  [inline], [protected]
```

12.217.4 Friends And Related Symbol Documentation

12.217.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Object & obj)  [friend]
```

References Object(), and Print().

Referenced by SmartPointer.

12.217.4.2 SmartPointer

```
template<class ObjectType>
friend class SmartPointer  [friend]
```

References Object(), operator<<, and SmartPointer.

Referenced by gdcm::Segment::AddSurface(), gdcm::Segment::GetSurface(), gdcm::Bitmap::SetLUT(), gdcm::Surface::SetMeshPrimitive and SmartPointer.

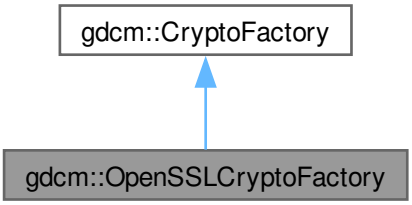
The documentation for this class was generated from the following file:

- gdcmObject.h

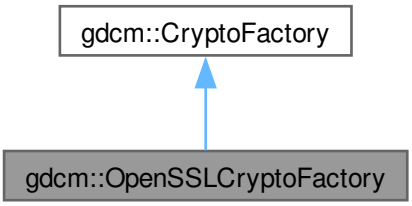
12.218 gdcm::OpenSSLCryptoFactory Class Reference

```
#include <gdcmOpenSSLCryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLCryptoFactory:



Collaboration diagram for gdcm::OpenSSLCryptoFactory:



Public Member Functions

- OpenSSLCryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Protected Member Functions

- void InitOpenSSL ()

Protected Member Functions inherited from gdcm::CryptoFactory

- CryptoFactory ()=default
- CryptoFactory (CryptoLib id)
- ~CryptoFactory ()=default

Additional Inherited Members

Public Types inherited from gdcm::CryptoFactory

- enum CryptoLib {
  DEFAULT = 0 ,
  OPENSSL = 1 ,
  CAPI = 2 ,
  OPENSSLP7 = 3 }

Static Public Member Functions inherited from gdcm::CryptoFactory

- static CryptoFactory * GetFactoryInstance (CryptoLib id=DEFAULT)

12.218.1 Constructor & Destructor Documentation

12.218.1.1 OpenSSLCryptoFactory()

```
gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (
            CryptoLib id)  [inline]
```

References gdcm::CryptoFactory::CryptoFactory(), and gdcmDebugMacro.

Referenced by InitOpenSSL().

12.218.2 Member Function Documentation

12.218.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax * gdcm::OpenSSLCryptoFactory::CreateCMSProvider () [inline], [virtual]
```

Implements gdcm::CryptoFactory.

References InitOpenSSL().

12.218.2.2 InitOpenSSL()

```
void gdcm::OpenSSLCryptoFactory::InitOpenSSL () [protected]
```

References OpenSSLCryptoFactory().

Referenced by CreateCMSProvider().

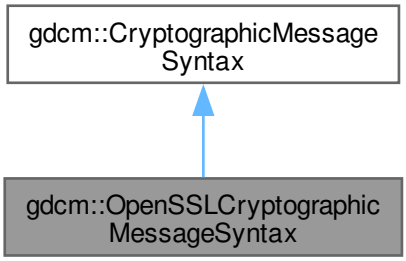
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptoFactory.h

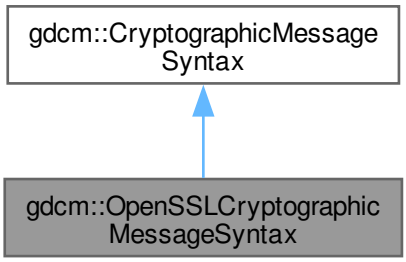
12.219 gdcm::OpenSSLCryptographicMessageSyntax Class Reference

```
#include <gdcmOpenSSLCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcm::OpenSSLCryptographicMessageSyntax:



Collaboration diagram for gdcm::OpenSSLCryptographicMessageSyntax:



Public Member Functions

- OpenSSLCryptographicMessageSyntax ()
- ~OpenSSLCryptographicMessageSyntax ()
- bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const

  decrypt content from a PKCS#7 envelopedData structure
- bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const

  create a CMS envelopedData structure
- CipherTypes GetCipherType () const
- bool ParseCertificateFile (const char *filename)
- bool ParseKeyFile (const char *filename)
- void SetCipherType (CipherTypes type)
- bool SetPassword (const char *pass, size_t passLen)

Public Member Functions inherited from gdcm::CryptographicMessageSyntax

- CryptographicMessageSyntax ()=default
- CryptographicMessageSyntax (const CryptographicMessageSyntax &)=delete
- virtual ~CryptographicMessageSyntax ()=default
- void operator= (const CryptographicMessageSyntax &)=delete

Additional Inherited Members

Public Types inherited from gdcm::CryptographicMessageSyntax

- enum CipherTypes {
  DES3_CIPHER ,
  AES128_CIPHER ,
  AES192_CIPHER ,
  AES256_CIPHER }

12.219.1 Constructor & Destructor Documentation

12.219.1.1 OpenSSLCryptographicMessageSyntax()

```
gdcm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ()
```

Referenced by Decrypt().

12.219.1.2 ~OpenSSLCryptographicMessageSyntax()

```
gdcm::OpenSSLCryptographicMessageSyntax::~OpenSSLCryptographicMessageSyntax ()
```

12.219.2 Member Function Documentation

12.219.2.1 Decrypt()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::Decrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len) const  [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

References OpenSSLCryptographicMessageSyntax().

12.219.2.2 Encrypt()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::Encrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len) const  [virtual]
```

create a CMS envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

12.219.2.3 GetCipherType()

```
CipherTypes gdcm::OpenSSLCryptographicMessageSyntax::GetCipherType () const  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.219.2.4 ParseCertificateFile()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (
            const char * filename)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.219.2.5 ParseKeyFile()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::ParseKeyFile (
            const char * filename)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.219.2.6 SetCipherType()

```
void gdcm::OpenSSLCryptographicMessageSyntax::SetCipherType (
            CipherTypes type)  [virtual]
```

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

12.219.2.7 SetPassword()

```
bool gdcm::OpenSSLCryptographicMessageSyntax::SetPassword (
            const char * pass,
            size_t passLen)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

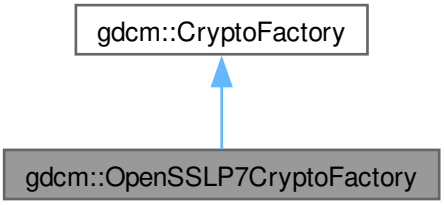
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptographicMessageSyntax.h

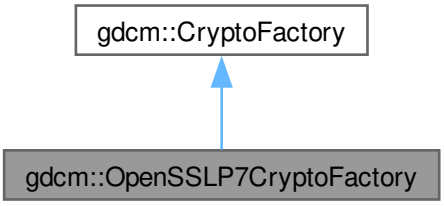
12.220 gdcm::OpenSSLP7CryptoFactory Class Reference

```
#include <gdcmOpenSSLP7CryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcm::OpenSSLP7CryptoFactory:



Public Member Functions

- OpenSSLP7CryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Additional Inherited Members

Public Types inherited from gdcm::CryptoFactory

- enum CryptoLib {
  DEFAULT = 0 ,
  OPENSSL = 1 ,
  CAPI = 2 ,
  OPENSSLP7 = 3 }

Static Public Member Functions inherited from gdcm::CryptoFactory

- static CryptoFactory * GetFactoryInstance (CryptoLib id=DEFAULT)

Protected Member Functions inherited from gdcm::CryptoFactory

- CryptoFactory ()=default
- CryptoFactory (CryptoLib id)
- ~CryptoFactory ()=default

12.220.1 Constructor & Destructor Documentation

12.220.1.1 OpenSSLP7CryptoFactory()

```
gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory (
            CryptoLib id)  [inline]
```

References gdcm::CryptoFactory::CryptoFactory(), and gdcmDebugMacro.

12.220.2 Member Function Documentation

12.220.2.1 CreateCMSProvider()

```
CryptographicMessageSyntax * gdcm::OpenSSLP7CryptoFactory::CreateCMSProvider ()  [inline], [virtual]
```

Implements gdcm::CryptoFactory.

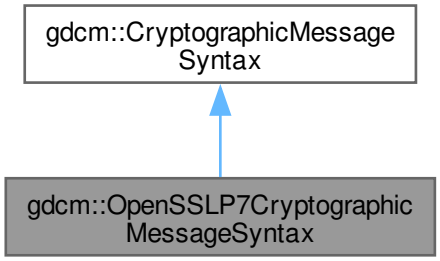
The documentation for this class was generated from the following file:

- gdcmOpenSSLP7CryptoFactory.h

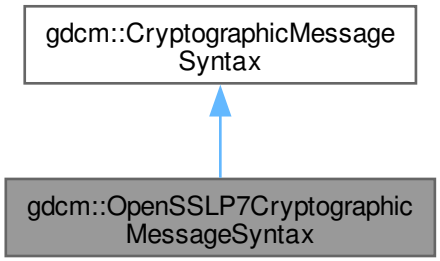
12.221 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference

```
#include <gdcmOpenSSLP7CryptographicMessageSyntax.h>
```

Inheritance diagram for gdcm::OpenSSLP7CryptographicMessageSyntax:



Collaboration diagram for gdcm::OpenSSLP7CryptographicMessageSyntax:



Public Member Functions

- OpenSSLP7CryptographicMessageSyntax ()
- ~OpenSSLP7CryptographicMessageSyntax ()
- bool Decrypt (char *output, size_t &outlen, const char *array, size_t len) const

  decrypt content from a PKCS#7 envelopedData structure
- bool Encrypt (char *output, size_t &outlen, const char *array, size_t len) const

  create a PKCS#7 envelopedData structure
- CipherTypes GetCipherType () const
- bool ParseCertificateFile (const char *filename)
- bool ParseKeyFile (const char *filename)
- void SetCipherType (CipherTypes type)
- bool SetPassword (const char *, size_t)

Public Member Functions inherited from gdcm::CryptographicMessageSyntax

- CryptographicMessageSyntax ()=default
- CryptographicMessageSyntax (const CryptographicMessageSyntax &)=delete
- virtual ~CryptographicMessageSyntax ()=default
- void operator= (const CryptographicMessageSyntax &)=delete

Additional Inherited Members

Public Types inherited from gdcm::CryptographicMessageSyntax

- enum CipherTypes {
  DES3_CIPHER ,
  AES128_CIPHER ,
  AES192_CIPHER ,
  AES256_CIPHER }

12.221.1 Detailed Description

Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities

See online documentation `http://www.openssl.org/docs/crypto/PKCS7_encrypt.html`

12.221.2 Constructor & Destructor Documentation

12.221.2.1 OpenSSLP7CryptographicMessageSyntax()

```
gdcm::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ()
```

Referenced by Decrypt().

12.221.2.2 ~OpenSSLP7CryptographicMessageSyntax()

```
gdcm::OpenSSLP7CryptographicMessageSyntax::~OpenSSLP7CryptographicMessageSyntax ()
```

12.221.3 Member Function Documentation

12.221.3.1 Decrypt()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::Decrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len) const  [virtual]
```

decrypt content from a PKCS#7 envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

References OpenSSLP7CryptographicMessageSyntax().

12.221.3.2 Encrypt()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::Encrypt (
            char * output,
            size_t & outlen,
            const char * array,
            size_t len) const  [virtual]
```

create a PKCS#7 envelopedData structure

Implements gdcm::CryptographicMessageSyntax.

12.221.3.3 GetCipherType()

```
CipherTypes gdcm::OpenSSLP7CryptographicMessageSyntax::GetCipherType () const  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.221.3.4 ParseCertificateFile()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (
            const char * filename)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.221.3.5 ParseKeyFile()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (
            const char * filename)  [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

12.221.3.6 SetCipherType()

```
void gdcm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (
            CipherTypes type)  [virtual]
```

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

12.221.3.7 SetPassword()

```
bool gdcm::OpenSSLP7CryptographicMessageSyntax::SetPassword (
            const char * ,
            size_t )  [inline], [virtual]
```

Implements gdcm::CryptographicMessageSyntax.

References gdcmWarningMacro.

The documentation for this class was generated from the following file:

- gdcmOpenSSLP7CryptographicMessageSyntax.h

12.222 gdcm::Orientation Class Reference

class to handle Orientation

```
#include <gdcmOrientation.h>
```

Public Types

- enum OrientationType {
  UNKNOWN ,
  AXIAL ,
  CORONAL ,
  SAGITTAL ,
  OBLIQUE }

Public Member Functions

- Orientation ()
- ~Orientation ()=default
- void Print (std::ostream &) const

  Print.

Static Public Member Functions

- static const char * GetLabel (OrientationType type)

  Return the label of an Orientation.
- static double GetObliquityThresholdCosineValue ()
- static OrientationType GetType (const double dircos[6])
- static void SetObliquityThresholdCosineValue (double val)

  ObliquityThresholdCosineValue stuff.

Static Protected Member Functions

- static char GetMajorAxisFromPatientRelativeDirectionCosine (double x, double y, double z)

Friends

- std::ostream & operator<< (std::ostream &_os, const Orientation &o)

12.222.1 Detailed Description

class to handle Orientation

12.222.2 Member Enumeration Documentation

12.222.2.1 OrientationType

```
enum gdcm::Orientation::OrientationType
```

Enumerator

UNKNOWN	
AXIAL	
CORONAL	
SAGITTAL	
OBLIQUE	

Examples

FixOrientation.cxx.

12.222.3 Constructor & Destructor Documentation

12.222.3.1 Orientation()

```
gdcm::Orientation::Orientation ()
```

Referenced by operator<<.

12.222.3.2 ~Orientation()

```
gdcm::Orientation::~Orientation ()  [default]
```

12.222.4 Member Function Documentation

12.222.4.1 GetLabel()

```
const char * gdcm::Orientation::GetLabel (
            OrientationType type)  [static]
```

Return the label of an Orientation.

Examples

FixOrientation.cxx.

12.222.4.2 GetMajorAxisFromPatientRelativeDirectionCosine()

```
char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (
            double x,
            double y,
            double z)  [static], [protected]
```

12.222.4.3 GetObliquityThresholdCosineValue()

```
double gdcm::Orientation::GetObliquityThresholdCosineValue ()  [static]
```

12.222.4.4 GetType()

```
OrientationType gdcm::Orientation::GetType (
            const double dircos[6])  [static]
```

Return the type of orientation from a direction cosines Input is an array of 6 double

Examples

FixOrientation.cxx.

12.222.4.5 Print()

```
void gdcm::Orientation::Print (
            std::ostream & ) const
```

Print.

Referenced by operator<<.

12.222.4.6 SetObliquityThresholdCosineValue()

```
void gdcm::Orientation::SetObliquityThresholdCosineValue (
            double val)  [static]
```

ObliquityThresholdCosineValue stuff.

12.222.5 Friends And Related Symbol Documentation

12.222.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Orientation & o)  [friend]
```

References Orientation(), and Print().

The documentation for this class was generated from the following file:

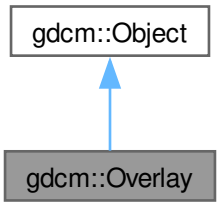
- gdcmOrientation.h

12.223 gdcm::Overlay Class Reference

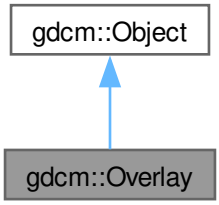
Overlay class.

```
#include <gdcmOverlay.h>
```

Inheritance diagram for gdcm::Overlay:



Collaboration diagram for gdcm::Overlay:



Public Types

- enum OverlayType {
  Invalid = 0 ,
  Graphics = 1 ,
  ROI = 2 }

Public Member Functions

- Overlay ()
- Overlay (Overlay const &ov)
- ~Overlay () override
- void Decompress (std::ostream &os) const

  Decode the internal OverlayData (packed bits) into unpacked representation.
- unsigned short GetBitPosition () const

  return bit position
- unsigned short GetBitsAllocated () const

  return bits allocated
- unsigned short GetColumns () const

  get columns
- const char * GetDescription () const

  get description
- unsigned short GetGroup () const

  Get Group number.
- const signed short * GetOrigin () const

  get origin
- const ByteValue & GetOverlayData () const
- unsigned short GetRows () const

  get rows
- const char * GetType () const

  get type
- OverlayType GetTypeAsEnum () const

- bool GetUnpackBuffer (char ∗buffer, size_t len) const
- size_t GetUnpackBufferLength () const
- bool GrabOverlayFromPixelData (DataSet const &ds)
- bool IsEmpty () const

  Return whether or not the Overlay is empty:
- bool IsInPixelData () const

  return if the Overlay is stored in the pixel data or not
- void IsInPixelData (bool b)

  Set whether or no the OverlayData is in the Pixel Data:
- bool IsZero () const

  return true if all bits are set to 0
- Overlay & operator= (Overlay const &ov)
- void Print (std::ostream &) const override

  Print.
- void SetBitPosition (unsigned short bitposition)

  set bit position
- void SetBitsAllocated (unsigned short bitsallocated)

  set bits allocated
- void SetColumns (unsigned short columns)

  set columns
- void SetDescription (const char ∗description)

  set description
- void SetFrameOrigin (unsigned short frameorigin)

  set frame origin
- void SetGroup (unsigned short group)

  Set Group number.
- void SetNumberOfFrames (unsigned int numberoframes)

  set number of frames
- void SetOrigin (const signed short origin[2])

  set origin
- void SetOverlay (const char ∗array, size_t length)

  set overlay from byte array + length
- void SetRows (unsigned short rows)

  set rows
- void SetType (const char ∗type)

  set type
- void Update (const DataElement &de)

  Update overlay from data element de:

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ∼Object ()
- void operator= (const Object &)

Static Public Member Functions

- static const char * GetOverlayTypeAsString (OverlayType ot)
- static OverlayType GetOverlayTypeFromString (const char *)

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.223.1 Detailed Description

Overlay class.

Note

see AreOverlaysInPixelData

Todo Is there actually any way to recognize an overlay ? On images with multiple overlay I do not see any way to differentiate them (other than the group tag).

Example:

12.223.2 Member Enumeration Documentation

12.223.2.1 OverlayType

```
enum gdcm::Overlay::OverlayType
```

Enumerator

Invalid	
Graphics	
ROI	

12.223.3 Constructor & Destructor Documentation

12.223.3.1 Overlay() [1/2]

```
gdcm::Overlay::Overlay ()
```

Referenced by Overlay(), and operator=().

12.223.3.2 ~Overlay()

```
gdcm::Overlay::~Overlay () [override]
```

12.223.3.3 Overlay() [2/2]

```
gdcm::Overlay::Overlay (
            Overlay const & ov)
```

References Overlay().

12.223.4 Member Function Documentation

12.223.4.1 Decompress()

```
void gdcm::Overlay::Decompress (
            std::ostream & os) const
```

Decode the internal OverlayData (packed bits) into unpacked representation.

12.223.4.2 GetBitPosition()

```
unsigned short gdcm::Overlay::GetBitPosition () const
```

return bit position

12.223.4.3 GetBitsAllocated()

```
unsigned short gdcm::Overlay::GetBitsAllocated () const
```

return bits allocated

12.223.4.4 GetColumns()

```
unsigned short gdcm::Overlay::GetColumns () const
```

get columns

12.223.4.5 GetDescription()

```
const char * gdcm::Overlay::GetDescription () const
```

get description

12.223.4.6 GetGroup()

```
unsigned short gdcm::Overlay::GetGroup () const
```

Get Group number.

12.223.4.7 GetOrigin()

```
const signed short * gdcm::Overlay::GetOrigin () const
```

get origin

12.223.4.8 GetOverlayData()

```
const ByteValue & gdcm::Overlay::GetOverlayData () const
```

Return the Overlay Data as ByteValue: Not thread safe

12.223.4.9 GetOverlayTypeAsString()

```
const char * gdcm::Overlay::GetOverlayTypeAsString (
            OverlayType ot)  [static]
```

12.223.4.10 GetOverlayTypeFromString()

```
OverlayType gdcm::Overlay::GetOverlayTypeFromString (
            const char * )  [static]
```

12.223.4.11 GetRows()

```
unsigned short gdcm::Overlay::GetRows () const
```

get rows

12.223.4.12 GetType()

```
const char * gdcm::Overlay::GetType () const
```

get type

12.223.4.13 GetTypeAsEnum()

```
OverlayType gdcm::Overlay::GetTypeAsEnum () const
```

12.223.4.14 GetUnpackBuffer()

```
bool gdcm::Overlay::GetUnpackBuffer (
            char * buffer,
            size_t len) const
```

Retrieve the unpack buffer for Overlay. This is an error if the size if below GetUnpackBufferLength()

12.223.4.15 GetUnpackBufferLength()

```
size_t gdcm::Overlay::GetUnpackBufferLength () const
```

Retrieve the size of the buffer needed to hold the Overlay as specified by Col & Row parameters

12.223.4.16 GrabOverlayFromPixelData()

```
bool gdcm::Overlay::GrabOverlayFromPixelData (
            DataSet const & ds)
```

Warning

Before calling this method, you must verify the consistency between the image metadata (Image PixelFormat, Rows, Columns) and the overlay parameters. This pre-verification is required to ensure that the bit-depth is compatible and that the overlay data fits within the allocated pixel storage.

12.223.4.17 IsEmpty()

```
bool gdcm::Overlay::IsEmpty () const
```

Return whether or not the Overlay is empty:

12.223.4.18 IsInPixelData() [1/2]

```
bool gdcm::Overlay::IsInPixelData () const
```

return if the Overlay is stored in the pixel data or not

12.223.4.19 IsInPixelData() [2/2]

```
void gdcm::Overlay::IsInPixelData (
            bool b)
```

Set whether or no the OverlayData is in the Pixel Data:

12.223.4.20 IsZero()

```
bool gdcm::Overlay::IsZero () const
```

return true if all bits are set to 0

12.223.4.21 operator=()

```
Overlay & gdcm::Overlay::operator= (
            Overlay const & ov)
```

References Overlay().

12.223.4.22 Print()

```
void gdcm::Overlay::Print (
            std::ostream & ) const  [override], [virtual]
```

Print.

Reimplemented from gdcm::Object.

12.223.4.23 SetBitPosition()

```
void gdcm::Overlay::SetBitPosition (
            unsigned short bitposition)
```

set bit position

12.223.4.24 SetBitsAllocated()

```
void gdcm::Overlay::SetBitsAllocated (
            unsigned short bitsallocated)
```

set bits allocated

12.223.4.25 SetColumns()

```
void gdcm::Overlay::SetColumns (
            unsigned short columns)
```

set columns

12.223.4.26 SetDescription()

```
void gdcm::Overlay::SetDescription (
            const char * description)
```

set description

12.223.4.27 SetFrameOrigin()

```
void gdcm::Overlay::SetFrameOrigin (
            unsigned short frameorigin)
```

set frame origin

12.223.4.28 SetGroup()

```
void gdcm::Overlay::SetGroup (
            unsigned short group)
```

Set Group number.

12.223.4.29 SetNumberOfFrames()

```
void gdcm::Overlay::SetNumberOfFrames (
            unsigned int numberofframes)
```

set number of frames

12.223.4.30 SetOrigin()

```
void gdcm::Overlay::SetOrigin (
            const signed short origin[2])
```

set origin

12.223.4.31 SetOverlay()

```
void gdcm::Overlay::SetOverlay (
            const char * array,
            size_t length)
```

set overlay from byte array + length

12.223.4.32 SetRows()

```
void gdcm::Overlay::SetRows (
            unsigned short rows)
```

set rows

12.223.4.33 SetType()

```
void gdcm::Overlay::SetType (
            const char * type)
```

set type

12.223.4.34 Update()

```
void gdcm::Overlay::Update (
            const DataElement & de)
```

Update overlay from data element de:

The documentation for this class was generated from the following file:

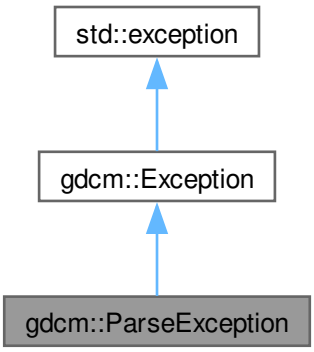
- gdcmOverlay.h

12.224 gdcm::ParseException Class Reference

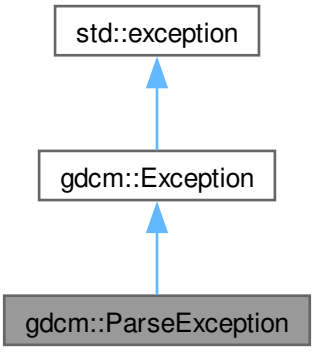
ParseException Standard exception handling object.

```
#include <gdcmParseException.h>
```

Inheritance diagram for gdcm::ParseException:



Collaboration diagram for gdcm::ParseException:



Public Member Functions

- ParseException ()=default
- ParseException (const ParseException &orig)
- ~ParseException () override throw ()
- const DataElement & GetLastElement () const
- ParseException & operator= (const ParseException &orig)
- void SetLastElement (DataElement &de)

Public Member Functions inherited from gdcm::Exception

- Exception (const char *desc="None", const char *file=__FILE__, unsigned int lineNumber=__LINE__, const char *func="")
- ~Exception () override throw ()
- const char * GetDescription () const

  Return the Description.
- const char * what () const override throw ()

  what implementation

12.224.1 Detailed Description

ParseException Standard exception handling object.

12.224.2 Constructor & Destructor Documentation

12.224.2.1 ParseException() [1/2]

```
gdcm::ParseException::ParseException () [default]
```

Referenced by ParseException(), and operator=().

12.224.2.2 ~ParseException()

```
gdcm::ParseException::~ParseException () throw ( ) [inline], [override]
```

12.224.2.3 ParseException() [2/2]

```
gdcm::ParseException::ParseException (
            const ParseException & orig) [inline]
```

References gdcm::Exception::Exception(), and ParseException().

12.224.3 Member Function Documentation

12.224.3.1 GetLastElement()

```
const DataElement & gdcm::ParseException::GetLastElement () const  [inline]
```

12.224.3.2 operator=()

```
ParseException & gdcm::ParseException::operator= (
            const ParseException & orig)  [inline]
```

Assignment operator.

References ParseException().

12.224.3.3 SetLastElement()

```
void gdcm::ParseException::SetLastElement (
            DataElement & de)  [inline]
```

Equivalence operator.

Referenced by gdcm::BasicOffsetTable::Read(), gdcm::Fragment::ReadBacktrack(), and gdcm::Fragment::ReadValue().

The documentation for this class was generated from the following file:

- gdcmParseException.h

12.225 gdcm::Parser Class Reference

Parser ala XML_Parser from expat (SAX).

```
#include <gdcmParser.h>
```

Public Types

- typedef void(∗ EndElementHandler) (void ∗userData, const Tag &name)
- enum ErrorType {
  NoError ,
  NoMemoryError ,
  SyntaxError ,
  NoElementsError ,
  TagMismatchError ,
  DuplicateAttributeError ,
  JunkAfterDocElementError ,
  UndefinedEntityError ,
  UnexpectedStateError }
- typedef void(∗ StartElementHandler) (void ∗userData, const Tag &tag, const char ∗atts[ ])

Public Member Functions

- Parser ()
- ~Parser ()
- unsigned long GetCurrentByteIndex () const
- ErrorType GetErrorCode () const
- void * GetUserData () const
- bool Parse (const char *s, int len, bool isFinal)
- void SetElementHandler (StartElementHandler start, EndElementHandler end)
- void SetUserData (void *userData)

Static Public Member Functions

- static const char * GetErrorString (ErrorType const &err)

Protected Member Functions

- char * GetBuffer (int len)
- bool ParseBuffer (int len, bool isFinal)
- ErrorType Process ()

12.225.1 Detailed Description

Parser ala XML_Parser from expat (SAX).

Detailed description here

Note

Simple API for DICOM

12.225.2 Member Typedef Documentation

12.225.2.1 EndElementHandler

```
typedef void(* gdcm::Parser::EndElementHandler) (void *userData, const Tag &name)
```

12.225.2.2 StartElementHandler

```
typedef void(* gdcm::Parser::StartElementHandler) (void *userData, const Tag &tag, const char
*atts[])
```

12.225.3 Member Enumeration Documentation

12.225.3.1 ErrorType

```
enum gdcm::Parser::ErrorType
```

Enumerator

NoError	
NoMemoryError	
SyntaxError	
NoElementsError	
TagMismatchError	
DuplicateAttributeError	
JunkAfterDocElementError	
UndefinedEntityError	
UnexpectedStateError	

12.225.4 Constructor & Destructor Documentation

12.225.4.1 Parser()

```
gdcm::Parser::Parser () [inline]
```

References NoError.

12.225.4.2 ~Parser()

```
gdcm::Parser::~Parser () [inline]
```

12.225.5 Member Function Documentation

12.225.5.1 GetBuffer()

```
char * gdcm::Parser::GetBuffer (
            int len) [protected]
```

12.225.5.2 GetCurrentByteIndex()

```
unsigned long gdcm::Parser::GetCurrentByteIndex () const
```

12.225.5.3 GetErrorCode()

```
ErrorType gdcm::Parser::GetErrorCode () const
```

12.225.5.4 GetErrorString()

```
const char * gdcm::Parser::GetErrorString (
            ErrorType const & err)  [static]
```

12.225.5.5 GetUserData()

```
void * gdcm::Parser::GetUserData () const
```

12.225.5.6 Parse()

```
bool gdcm::Parser::Parse (
            const char * s,
            int len,
            bool isFinal)
```

12.225.5.7 ParseBuffer()

```
bool gdcm::Parser::ParseBuffer (
            int len,
            bool isFinal)  [protected]
```

12.225.5.8 Process()

```
ErrorType gdcm::Parser::Process ()  [protected]
```

12.225.5.9 SetElementHandler()

```
void gdcm::Parser::SetElementHandler (
            StartElementHandler start,
            EndElementHandler end)
```

12.225.5.10 SetUserData()

```
void gdcm::Parser::SetUserData (
            void * userData)
```

The documentation for this class was generated from the following file:

- gdcmParser.h

12.226 gdcm::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

```
#include <gdcmPatient.h>
```

Public Member Functions

- Patient ()=default

12.226.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

12.226.2 Constructor & Destructor Documentation

12.226.2.1 Patient()

```
gdcm::Patient::Patient () [default]
```

The documentation for this class was generated from the following file:

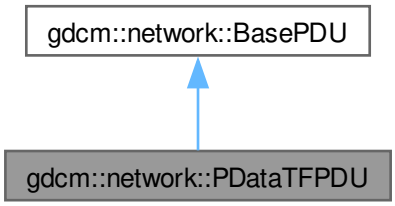
- gdcmPatient.h

12.227 gdcm::network::PDataTFPDU Class Reference

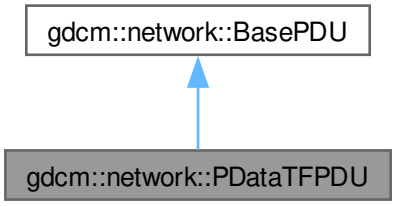
PDataTFPDU.

```
#include <gdcmPDataTFPDU.h>
```

Inheritance diagram for gdcm::network::PDataTFPDU:



Collaboration diagram for gdcm::network::PDataTFPDU:



Public Types

- typedef std::vector< PresentationDataValue >::size_type SizeType

Public Member Functions

- PDataTFPDU ()
- void AddPresentationDataValue (PresentationDataValue const &pdv)
- SizeType GetNumberOfPresentationDataValues () const
- PresentationDataValue const & GetPresentationDataValue (SizeType i) const
- bool IsLastFragment () const override
- void Print (std::ostream &os) const override
- std::istream & Read (std::istream &is) override
- size_t Size () const override
- const std::ostream & Write (std::ostream &os) const override

Public Member Functions inherited from gdcm::network::BasePDU

- virtual ~BasePDU ()=default

Protected Member Functions

- std::istream & ReadInto (std::istream &is, std::ostream &os)

12.227.1 Detailed Description

PDataTFPDU.

Table 9-22 P-DATA-TF PDU FIELDS

12.227.2 Member Typedef Documentation

12.227.2.1 SizeType

```
typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType
```

12.227.3 Constructor & Destructor Documentation

12.227.3.1 PDataTFPDU()

```
gdcm::network::PDataTFPDU::PDataTFPDU ()
```

12.227.4 Member Function Documentation

12.227.4.1 AddPresentationDataValue()

```
void gdcm::network::PDataTFPDU::AddPresentationDataValue (
            PresentationDataValue const & pdv)  [inline]
```

References Size().

12.227.4.2 GetNumberOfPresentationDataValues()

```
SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues () const  [inline]
```

12.227.4.3 GetPresentationDataValue()

```
PresentationDataValue const & gdcm::network::PDataTFPDU::GetPresentationDataValue (
            SizeType i) const  [inline]
```

12.227.4.4 IsLastFragment()

```
bool gdcm::network::PDataTFPDU::IsLastFragment () const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.227.4.5 Print()

```
void gdcm::network::PDataTFPDU::Print (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.227.4.6 Read()

```
std::istream & gdcm::network::PDataTFPDU::Read (
            std::istream & is)  [override], [virtual]
```

Implements gdcm::network::BasePDU.

12.227.4.7 ReadInto()

```
std::istream & gdcm::network::PDataTFPDU::ReadInto (
            std::istream & is,
            std::ostream & os)  [protected]
```

12.227.4.8 Size()

```
size_t gdcm::network::PDataTFPDU::Size () const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

Referenced by AddPresentationDataValue().

12.227.4.9 Write()

```
const std::ostream & gdcm::network::PDataTFPDU::Write (
            std::ostream & os) const  [override], [virtual]
```

Implements gdcm::network::BasePDU.

The documentation for this class was generated from the following file:

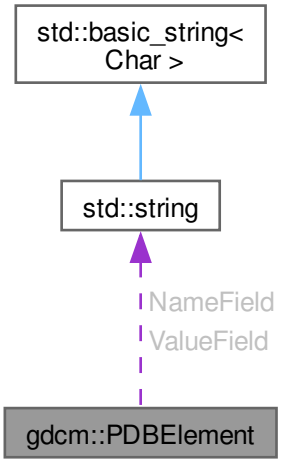
- gdcmPDataTFPDU.h

12.228 gdcm::PDBElement Class Reference

Class to represent a PDB Element.

```
#include <gdcmPDBElement.h>
```

Collaboration diagram for gdcm::PDBElement:



Public Member Functions

- PDBElement ()=default
- const char ∗ GetName () const

  Set/Get Name.
- const char ∗ GetValue () const

  Set/Get Value.
- bool operator== (const PDBElement &de) const
- void SetName (const char ∗name)
- void SetValue (const char ∗value)

Protected Attributes

- std::string NameField
- std::string ValueField

Friends

- std::ostream & operator<< (std::ostream &os, const PDBElement &val)

12.228.1 Detailed Description

Class to represent a PDB Element.

See also

PDBHeader

12.228.2 Constructor & Destructor Documentation

12.228.2.1 PDBElement()

```
gdcm::PDBElement::PDBElement () [default]
```

References PDBElement(), and operator<<.

Referenced by PDBElement(), operator<<, and operator==().

12.228.3 Member Function Documentation

12.228.3.1 GetName()

```
const char * gdcm::PDBElement::GetName () const [inline]
```

Set/Get Name.

References NameField.

12.228.3.2 GetValue()

```
const char * gdcm::PDBElement::GetValue () const [inline]
```

Set/Get Value.

References ValueField.

12.228.3.3 operator==()

```
bool gdcm::PDBElement::operator== (
            const PDBElement & de) const [inline]
```

References PDBElement(), NameField, and ValueField.

12.228.3.4 SetName()

```
void gdcm::PDBElement::SetName (
            const char * name)  [inline]
```

References NameField.

12.228.3.5 SetValue()

```
void gdcm::PDBElement::SetValue (
            const char * value)  [inline]
```

References ValueField.

12.228.4 Friends And Related Symbol Documentation

12.228.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const PDBElement & val)  [friend]
```

References PDBElement(), NameField, and ValueField.

Referenced by PDBElement().

12.228.5 Member Data Documentation

12.228.5.1 NameField

```
std::string gdcm::PDBElement::NameField  [protected]
```

Referenced by GetName(), operator<<, operator==(), and SetName().

12.228.5.2 ValueField

```
std::string gdcm::PDBElement::ValueField  [protected]
```

Referenced by GetValue(), operator<<, operator==(), and SetValue().

The documentation for this class was generated from the following file:

- gdcmPDBElement.h

12.229 gdcm::PDBHeader Class Reference

Class for PDBHeader.

```
#include <gdcmPDBHeader.h>
```

Public Member Functions

- PDBHeader ()=default
- ~PDBHeader ()=default
- bool FindPDBElementByName (const char *name)

  Return true if the PDB element matching name is found or not.
- const PDBElement & GetPDBElementByName (const char *name)
- bool LoadFromDataElement (DataElement const &de)

  Load the PDB Header from a DataElement of a DataSet.
- void Print (std::ostream &os) const

  Print.

Static Public Member Functions

- static const PrivateTag & GetPDBInfoTag ()

  Return the Private Tag where the PDB header is stored within a DICOM DataSet.

Protected Member Functions

- const PDBElement & GetPDBEEnd () const

Friends

- std::ostream & operator<< (std::ostream &_os, const PDBHeader &d)

12.229.1 Detailed Description

Class for PDBHeader.

GEMS MR Image have an Attribute (0025,1b,GEMS_SERS_01) which store the Acquisition parameter of the MR Image. It is compressed and can therefore not be used as is. This class de-encapsulated the Protocol Data Block and allow users to query element by name.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.

: the API of this class might change.

: SEDESC is not always pure ASCII it can contains latin1

See also

CSAHeader

12.229.2 Constructor & Destructor Documentation

12.229.2.1 PDBHeader()

```
gdcm::PDBHeader::PDBHeader () [default]
```

Referenced by operator<<.

12.229.2.2 ~PDBHeader()

```
gdcm::PDBHeader::~PDBHeader () [default]
```

12.229.3 Member Function Documentation

12.229.3.1 FindPDBElementByName()

```
bool gdcm::PDBHeader::FindPDBElementByName (
            const char * name)
```

Return true if the PDB element matching name is found or not.

12.229.3.2 GetPDBEEnd()

```
const PDBElement & gdcm::PDBHeader::GetPDBEEnd () const [protected]
```

12.229.3.3 GetPDBElementByName()

```
const PDBElement & gdcm::PDBHeader::GetPDBElementByName (
            const char * name)
```

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

12.229.3.4 GetPDBInfoTag()

```
const PrivateTag & gdcm::PDBHeader::GetPDBInfoTag () [static]
```

Return the Private Tag where the PDB header is stored within a DICOM DataSet.

12.229.3.5 LoadFromDataElement()

```
bool gdcm::PDBHeader::LoadFromDataElement (
            DataElement const & de)
```

Load the PDB Header from a DataElement of a DataSet.

12.229.3.6 Print()

```
void gdcm::PDBHeader::Print (
            std::ostream & os) const
```

Print.

Referenced by operator<<.

12.229.4 Friends And Related Symbol Documentation

12.229.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const PDBHeader & d)  [friend]
```

References PDBHeader(), and Print().

The documentation for this class was generated from the following file:

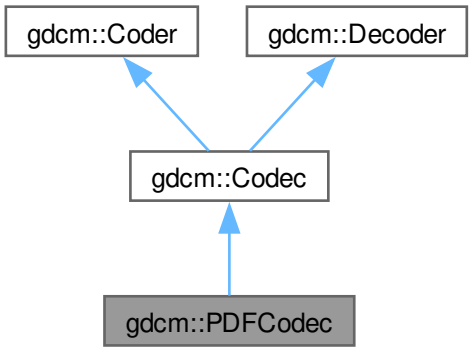
- gdcmPDBHeader.h

12.230 gdcm::PDFCodec Class Reference

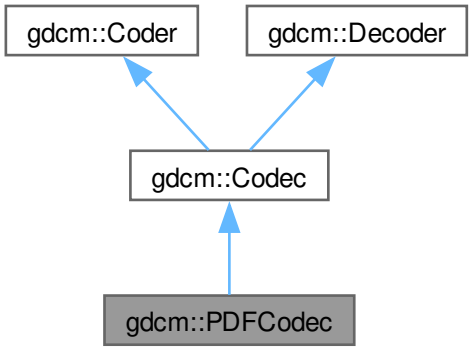
PDFCodec class.

```
#include <gdcmPDFCodec.h>
```

Inheritance diagram for gdcm::PDFCodec:



Collaboration diagram for gdcm::PDFCodec:



Public Member Functions

- PDFCodec ()
- ~PDFCodec () override
- bool CanCode (TransferSyntax const &) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &) const override

  Return whether this decoder support this transfer syntax (can decode it).
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Additional Inherited Members

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Protected Member Functions inherited from gdcm::Decoder

- virtual bool DecodeByStreams (std::istream &, std::ostream &)

12.230.1 Detailed Description

PDFCodec class.

12.230.2 Constructor & Destructor Documentation

12.230.2.1 PDFCodec()

```
gdcm::PDFCodec::PDFCodec ()
```

12.230.2.2 ~PDFCodec()

```
gdcm::PDFCodec::~PDFCodec ()  [override]
```

12.230.3 Member Function Documentation

12.230.3.1 CanCode()

```
bool gdcm::PDFCodec::CanCode (
            TransferSyntax const & ) const  [inline], [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Implements gdcm::Coder.

12.230.3.2 CanDecode()

```
bool gdcm::PDFCodec::CanDecode (
            TransferSyntax const & ) const  [inline], [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Implements gdcm::Decoder.

12.230.3.3 Decode()

```
bool gdcm::PDFCodec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::Decoder.

The documentation for this class was generated from the following file:

- gdcmPDFCodec.h

12.231 gdcm::network::PDUFactory Class Reference

PDUFactory basically, given an initial byte, construct the.

```
#include <gdcmPDUFactory.h>
```

Static Public Member Functions

- static BasePDU ∗ ConstructAbortPDU ()
- static BasePDU ∗ ConstructPDU (uint8_t itemtype)
- static BasePDU ∗ ConstructReleasePDU ()
- static std::vector< BasePDU ∗ > CreateCEchoPDU (const ULConnection &inConnection)
- static std::vector< BasePDU ∗ > CreateCFindPDU (const ULConnection &inConnection, const BaseRootQuery ∗inRootQuery)
- static std::vector< BasePDU ∗ > CreateCMovePDU (const ULConnection &inConnection, const BaseRootQuery ∗inRootQuery)
- static std::vector< BasePDU ∗ > CreateCStoreRQPDU (const ULConnection &inConnection, const File &file, bool writeDataSet=true)
- static std::vector< BasePDU ∗ > CreateCStoreRSPPDU (const DataSet ∗inDataSet, const BasePDU ∗inPC)
- static std::vector< BasePDU ∗ > CreateNActionPDU (const ULConnection &inConnection, const BaseQuery ∗inQuery)
- static std::vector< BasePDU ∗ > CreateNCreatePDU (const ULConnection &inConnection, const BaseQuery ∗inQuery)
- static std::vector< BasePDU ∗ > CreateNDeletePDU (const ULConnection &inConnection, const BaseQuery ∗inQuery)
- static std::vector< BasePDU ∗ > CreateNEventReportPDU (const ULConnection &inConnection, const BaseQuery ∗inQuery)
- static std::vector< BasePDU ∗ > CreateNGetPDU (const ULConnection &inConnection, const BaseQuery ∗in↩ Query)
- static std::vector< BasePDU ∗ > CreateNSetPDU (const ULConnection &inConnection, const BaseQuery ∗in↩ Query)
- static EEventID DetermineEventByPDU (const BasePDU ∗inPDU)
- static std::vector< PresentationDataValue > GetPDVs (const std::vector< BasePDU ∗ > &inDataPDUs)

12.231.1 Detailed Description

PDUFactory basically, given an initial byte, construct the.

appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

12.231.2 Member Function Documentation

12.231.2.1 ConstructAbortPDU()

```
BasePDU * gdcm::network::PDUFactory::ConstructAbortPDU ()  [static]
```

12.231.2.2 ConstructPDU()

```
BasePDU * gdcm::network::PDUFactory::ConstructPDU (
            uint8_t itemtype)  [static]
```

12.231.2.3 ConstructReleasePDU()

```
BasePDU * gdcm::network::PDUFactory::ConstructReleasePDU ()  [static]
```

12.231.2.4 CreateCEchoPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCEchoPDU (
            const ULConnection & inConnection)  [static]
```

12.231.2.5 CreateCFindPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCFindPDU (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery)  [static]
```

12.231.2.6 CreateCMovePDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCMovePDU (
            const ULConnection & inConnection,
            const BaseRootQuery * inRootQuery)  [static]
```

12.231.2.7 CreateCStoreRQPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCStoreRQPDU (
            const ULConnection & inConnection,
            const File & file,
            bool writeDataSet = true)  [static]
```

12.231.2.8 CreateCStoreRSPPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateCStoreRSPPDU (
            const DataSet * inDataSet,
            const BasePDU * inPC)  [static]
```

12.231.2.9 CreateNActionPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNActionPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.231.2.10 CreateNCreatePDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNCreatePDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.231.2.11 CreateNDeletePDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNDeletePDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.231.2.12 CreateNEventReportPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNEventReportPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.231.2.13 CreateNGetPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNGetPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.231.2.14 CreateNSetPDU()

```
std::vector< BasePDU * > gdcm::network::PDUFactory::CreateNSetPDU (
            const ULConnection & inConnection,
            const BaseQuery * inQuery)  [static]
```

12.231.2.15 DetermineEventByPDU()

```
EEventID gdcm::network::PDUFactory::DetermineEventByPDU (
            const BasePDU * inPDU)  [static]
```

12.231.2.16 GetPDVs()

```
std::vector< PresentationDataValue > gdcm::network::PDUFactory::GetPDVs (
            const std::vector< BasePDU * > & inDataPDUs)  [static]
```

The documentation for this class was generated from the following file:

- gdcmPDUFactory.h

12.232 gdcm::PersonName Class Reference

PersonName class.

```
#include <gdcmPersonName.h>
```

Public Member Functions

- unsigned int GetMaxLength () const
- unsigned int GetNumberOfComponents () const
- void Print (std::ostream &os) const
- void SetBlob (const std::vector< char > &v)
- void SetComponents (const char *comp1="", const char *comp2="", const char *comp3="", const char *comp4="", const char *comp5="")
- void SetComponents (const char *components[ ])

Public Attributes

- char Component [MaxNumberOfComponents][MaxLength+1]

Static Public Attributes

- static const unsigned int MaxLength = 64
- static const unsigned int MaxNumberOfComponents = 5
- static const char Padding = ' '
- static const char Separator = '^'

12.232.1 Detailed Description

PersonName class.

12.232.2 Member Function Documentation

12.232.2.1 GetMaxLength()

```
unsigned int gdcm::PersonName::GetMaxLength () const  [inline]
```

References MaxLength.

Referenced by SetComponents().

12.232.2.2 GetNumberOfComponents()

```
unsigned int gdcm::PersonName::GetNumberOfComponents () const  [inline]
```

References Component.

12.232.2.3 Print()

```
void gdcm::PersonName::Print (
            std::ostream & os) const  [inline]
```

References Component.

12.232.2.4 SetBlob()

```
void gdcm::PersonName::SetBlob (
            const std::vector< char > & v)  [inline]
```

12.232.2.5 SetComponents() [1/2]

```
void gdcm::PersonName::SetComponents (
            const char * comp1 = "",
            const char * comp2 = "",
            const char * comp3 = "",
            const char * comp4 = "",
            const char * comp5 = "")  [inline]
```

References SetComponents().

Referenced by SetComponents().

12.232.2.6 SetComponents() [2/2]

```
void gdcm::PersonName::SetComponents (
            const char * components[])  [inline]
```

References Component, and GetMaxLength().

12.232.3 Member Data Documentation

12.232.3.1 Component

```
char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]
```

Referenced by GetNumberOfComponents(), Print(), and SetComponents().

12.232.3.2 MaxLength

```
const unsigned int gdcm::PersonName::MaxLength = 64  [static]
```

Referenced by GetMaxLength().

12.232.3.3 MaxNumberOfComponents

```
const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5  [static]
```

12.232.3.4 Padding

```
const char gdcm::PersonName::Padding = ' '  [static]
```

12.232.3.5 Separator

```
const char gdcm::PersonName::Separator = '^'  [static]
```

The documentation for this class was generated from the following file:

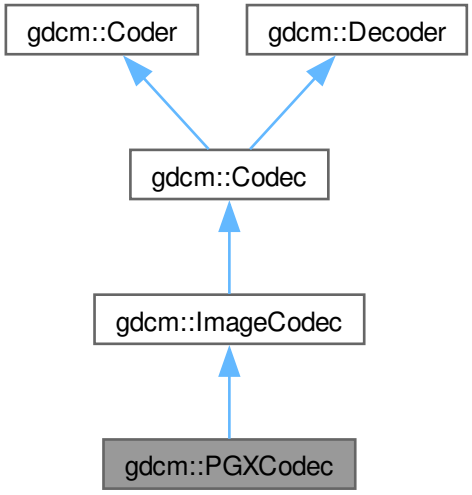
- gdcmPersonName.h

12.233 gdcm::PGXCodec Class Reference

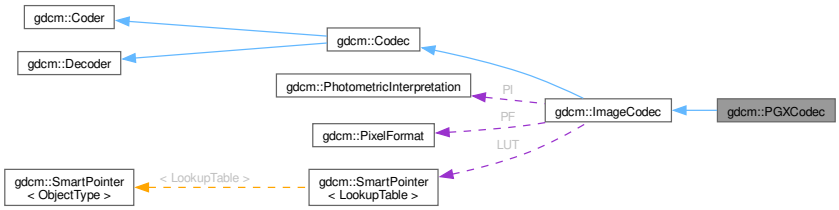
Class to do PGX.

```
#include <gdcmPGXCodec.h>
```

Inheritance diagram for gdcm::PGXCodec:



Collaboration diagram for gdcm::PGXCodec:



Public Member Functions

- PGXCodec ()
- ~PGXCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool Read (const char *filename, DataElement &out) const
- bool Write (const char *filename, const DataElement &out) const

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- bool Decode (DataElement const &is_, DataElement &os) override

  Decode.
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions inherited from gdcm::ImageCodec

- virtual bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- virtual bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is_, std::ostream &os) override
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsFrameEncoder ()
- virtual bool IsRowEncoder ()
- virtual bool IsValid (PhotometricInterpretation const &pi)
- virtual bool StartEncode (std::ostream &os)
- virtual bool StopEncode (std::ostream &os)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.233.1 Detailed Description

Class to do PGX.

See PGX as used in JPEG 2000 implementation and reference images

12.233.2 Constructor & Destructor Documentation

12.233.2.1 PGXCodec()

```
gdcm::PGXCodec::PGXCodec ()
```

12.233.2.2 ~PGXCodec()

```
gdcm::PGXCodec::~PGXCodec ()  [override]
```

12.233.3 Member Function Documentation

12.233.3.1 CanCode()

```
bool gdcm::PGXCodec::CanCode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

12.233.3.2 CanDecode()

```
bool gdcm::PGXCodec::CanDecode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.233.3.3 Clone()

```
ImageCodec * gdcm::PGXCodec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.233.3.4 GetHeaderInfo()

```
bool gdcm::PGXCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.233.3.5 Read()

```
bool gdcm::PGXCodec::Read (
            const char * filename,
            DataElement & out) const
```

12.233.3.6 Write()

```
bool gdcm::PGXCodec::Write (
            const char * filename,
            const DataElement & out) const
```

The documentation for this class was generated from the following file:

- gdcmPGXCodec.h

12.234 gdcm::PhotometricInterpretation Class Reference

Class to represent an PhotometricInterpretation.

```
#include <gdcmPhotometricInterpretation.h>
```

Public Types

- enum PIType {
  UNKNOWN = 0 ,
  MONOCHROME1 ,
  MONOCHROME2 ,
  PALETTE_COLOR ,
  RGB ,
  HSV ,
  ARGB ,
  CMYK ,
  YBR_FULL ,
  YBR_FULL_422 ,
  YBR_PARTIAL_422 ,
  YBR_PARTIAL_420 ,
  YBR_ICT ,
  YBR_RCT ,
  PI_END }

Public Member Functions

- PhotometricInterpretation (PIType pi=UNKNOWN)
- unsigned short GetSamplesPerPixel () const

  return the value for Sample Per Pixel associated with a particular Photometric Interpretation
- const char * GetString () const
- PIType GetType () const
- bool IsLossless () const
- bool IsLossy () const
- bool IsSameColorSpace (PhotometricInterpretation const &pi) const
- operator PIType () const

Static Public Member Functions

- static const char * GetPIString (PIType pi)
- static PIType GetPIType (const char *pi)
- static bool IsRetired (PIType pi)

Friends

- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &pi)

12.234.1 Detailed Description

Class to represent an PhotometricInterpretation.

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, ExtractImageRegion.cs, FileChangeTS.cs, FileChangeTSLossy.cs, HelloVizWorld.cxx, MpegVideoInfo.cs, csa2img.cxx, and iU22tomultisc.cxx.

12.234.2 Member Enumeration Documentation

12.234.2.1 PIType

```
enum gdcm::PhotometricInterpretation::PIType
```

Enumerator

UNKNOWN	
MONOCHROME1	
MONOCHROME2	

PALETTE_COLOR	
RGB	
HSV	
ARGB	
CMYK	
YBR_FULL	
YBR_FULL_422	
YBR_PARTIAL_422	
YBR_PARTIAL_420	
YBR_ICT	
YBR_RCT	
PI_END	

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, and MpegVideoInfo.cs.

12.234.3 Constructor & Destructor Documentation

12.234.3.1 PhotometricInterpretation()

```
gdcm::PhotometricInterpretation::PhotometricInterpretation (
            PIType pi = UNKNOWN)  [inline]
```

References UNKNOWN.

Referenced by GetSamplesPerPixel(), IsSameColorSpace(), and operator<<.

12.234.4 Member Function Documentation

12.234.4.1 GetPIString()

```
const char * gdcm::PhotometricInterpretation::GetPIString (
            PIType pi)  [static]
```

Referenced by operator<<.

12.234.4.2 GetPIType()

```
PIType gdcm::PhotometricInterpretation::GetPIType (
            const char * pi)  [static]
```

12.234.4.3 GetSamplesPerPixel()

```
unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel () const
```

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

References PhotometricInterpretation(), and operator<<.

12.234.4.4 GetString()

```
const char * gdcm::PhotometricInterpretation::GetString () const
```

12.234.4.5 GetType()

```
PIType gdcm::PhotometricInterpretation::GetType () const  [inline]
```

12.234.4.6 IsLossless()

```
bool gdcm::PhotometricInterpretation::IsLossless () const
```

12.234.4.7 IsLossy()

```
bool gdcm::PhotometricInterpretation::IsLossy () const
```

12.234.4.8 IsRetired()

```
bool gdcm::PhotometricInterpretation::IsRetired (
            PIType pi)  [static]
```

12.234.4.9 IsSameColorSpace()

```
bool gdcm::PhotometricInterpretation::IsSameColorSpace (
            PhotometricInterpretation const & pi) const
```

References PhotometricInterpretation().

12.234.4.10 operator PIType()

```
gdcm::PhotometricInterpretation::operator PIType () const  [inline]
```

12.234.5 Friends And Related Symbol Documentation

12.234.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const PhotometricInterpretation & pi)  [friend]
```

References PhotometricInterpretation(), and GetPIString().

Referenced by GetSamplesPerPixel().

The documentation for this class was generated from the following file:

- gdcmPhotometricInterpretation.h

12.235 gdcm::PixelFormat Class Reference

PixelFormat.

```
#include <gdcmPixelFormat.h>
```

Public Types

- enum ScalarType {
  UINT8 ,
  INT8 ,
  UINT12 ,
  INT12 ,
  UINT16 ,
  INT16 ,
  UINT32 ,
  INT32 ,
  UINT64 ,
  INT64 ,
  FLOAT16 ,
  FLOAT32 ,
  FLOAT64 ,
  SINGLEBIT ,
  UNKNOWN }

Public Member Functions

- PixelFormat ()
- PixelFormat (ScalarType st)
- PixelFormat (unsigned short samplesperpixel, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- unsigned short GetBitsAllocated () const

  BitsAllocated see Tag (0028,0100) US Bits Allocated.
- unsigned short GetBitsStored () const

  BitsStored see Tag (0028,0101) US Bits Stored.
- unsigned short GetHighBit () const

  HighBit see Tag (0028,0102) US High Bit.
- int64_t GetMax () const

  return the max possible of the pixel
- int64_t GetMin () const

  return the min possible of the pixel
- unsigned short GetPixelRepresentation () const

  PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.
- uint8_t GetPixelSize () const
- unsigned short GetSamplesPerPixel () const
- ScalarType GetScalarType () const

  ScalarType does not take into account the sample per pixel.
- const char * GetScalarTypeAsString () const
- bool IsCompatible (const TransferSyntax &ts) const
- bool IsValid () const

  return IsValid
- operator ScalarType () const
- bool operator!= (const PixelFormat &pf) const
- bool operator!= (ScalarType st) const
- bool operator== (const PixelFormat &pf) const
- bool operator== (ScalarType st) const
- void Print (std::ostream &os) const

  Print.
- void SetBitsAllocated (unsigned short ba)
- void SetBitsStored (unsigned short bs)
- void SetHighBit (unsigned short hb)
- void SetPixelRepresentation (unsigned short pr)
- void SetSamplesPerPixel (unsigned short spp)
- void SetScalarType (ScalarType st)

Protected Member Functions

- bool Validate ()

  When image with 24/24/23 was read, need to validate.

Friends

- class Bitmap
- std::ostream & operator<< (std::ostream &_os, const PixelFormat &pf)

12.235.1 Detailed Description

PixelFormat.

By default the Pixel Type will be instantiated with the following parameters:

- SamplesPerPixel : 1
- BitsAllocated : 8
- BitsStored : 8
- HighBit : 7
- PixelRepresentation : 0

Fundamentally PixelFormat is very close to what DICOM allows. It will be very hard to extend this class for the upcoming DICOM standard where Floating 32 and 64bits will be allowed.

It is also very hard for this class to fully support 64bits integer type (see GetMin / GetMax signature restricted to 64bits signed).

Examples

CreateARGBImage.cxx, CreateCMYKImage.cxx, DecompressImage.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, FixJAIBugJPEGLS.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, MpegVideoInfo.cs, RescaleImage.cs, TemplateEmptyImage.cxx, csa2img.cxx, iU22tomultisc.cxx, and threadgdcm.cxx.

12.235.2 Member Enumeration Documentation

12.235.2.1 ScalarType

`enum gdcm::PixelFormat::ScalarType`

Enumerator

UINT8	
INT8	
UINT12	
INT12	
UINT16	
INT16	
UINT32	
INT32	
UINT64	

INT64	
FLOAT16	
FLOAT32	
FLOAT64	
SINGLEBIT	
UNKNOWN	

Examples

GetArray.cs.

12.235.3 Constructor & Destructor Documentation

12.235.3.1 PixelFormat() [1/3]

```
gdcm::PixelFormat::PixelFormat ()  [inline]
```

References PixelFormat().

Referenced by PixelFormat(), Bitmap, operator!=(), operator<<, and operator==().

12.235.3.2 PixelFormat() [2/3]

```
gdcm::PixelFormat::PixelFormat (
            unsigned short samplesperpixel,
            unsigned short bitsallocated = 8,
            unsigned short bitsstored = 8,
            unsigned short highbit = 7,
            unsigned short pixelrepresentation = 0)  [inline], [explicit]
```

12.235.3.3 PixelFormat() [3/3]

```
gdcm::PixelFormat::PixelFormat (
            ScalarType st)
```

12.235.4 Member Function Documentation

12.235.4.1 GetBitsAllocated()

```
unsigned short gdcm::PixelFormat::GetBitsAllocated () const  [inline]
```

BitsAllocated see Tag (0028,0100) US Bits Allocated.

Examples

GetJPEGSamplePrecision.cxx.

12.235.4.2 GetBitsStored()

```
unsigned short gdcm::PixelFormat::GetBitsStored () const [inline]
```

BitsStored see Tag (0028,0101) US Bits Stored.

Examples

GetJPEGSamplePrecision.cxx.

12.235.4.3 GetHighBit()

```
unsigned short gdcm::PixelFormat::GetHighBit () const [inline]
```

HighBit see Tag (0028,0102) US High Bit.

12.235.4.4 GetMax()

```
int64_t gdcm::PixelFormat::GetMax () const
```

return the max possible of the pixel

12.235.4.5 GetMin()

```
int64_t gdcm::PixelFormat::GetMin () const
```

return the min possible of the pixel

12.235.4.6 GetPixelRepresentation()

```
unsigned short gdcm::PixelFormat::GetPixelRepresentation () const [inline]
```

PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.

12.235.4.7 GetPixelSize()

```
uint8_t gdcm::PixelFormat::GetPixelSize () const
```

return the size of the pixel This is the number of words it would take to store one pixel

Warning

the return value takes into account the SamplesPerPixel

in the rare case when BitsAllocated == 12, the function assume word padding and value returned will be identical as if BitsAllocated == 16

Examples

ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, and threadgdcm.cxx.

12.235.4.8 GetSamplesPerPixel()

```
unsigned short gdcm::PixelFormat::GetSamplesPerPixel () const
```

Samples Per Pixel see (0028,0002) US Samples Per Pixel DICOM - only allows 1, 3 and 4 as valid value. Other value are undefined behavior.

Examples

threadgdcm.cxx.

12.235.4.9 GetScalarType()

```
ScalarType gdcm::PixelFormat::GetScalarType () const
```

ScalarType does not take into account the sample per pixel.

Examples

GetArray.cs.

Referenced by operator ScalarType(), operator!=(), and operator==().

12.235.4.10 GetScalarTypeAsString()

```
const char * gdcm::PixelFormat::GetScalarTypeAsString () const
```

Examples

GetArray.cs.

12.235.4.11 IsCompatible()

```
bool gdcm::PixelFormat::IsCompatible (
            const TransferSyntax & ts) const
```

12.235.4.12 IsValid()

```
bool gdcm::PixelFormat::IsValid () const
```

return IsValid

12.235.4.13 operator ScalarType()

```
gdcm::PixelFormat::operator ScalarType () const  [inline]
```

References GetScalarType().

12.235.4.14 operator"!=() [1/2]

```
bool gdcm::PixelFormat::operator!= (
            const PixelFormat & pf) const  [inline]
```

References PixelFormat().

12.235.4.15 operator"!=() [2/2]

```
bool gdcm::PixelFormat::operator!= (
            ScalarType st) const  [inline]
```

References GetScalarType().

12.235.4.16 operator==() [1/2]

```
bool gdcm::PixelFormat::operator== (
            const PixelFormat & pf) const  [inline]
```

References PixelFormat().

12.235.4.17 operator==() [2/2]

```
bool gdcm::PixelFormat::operator== (
            ScalarType st) const  [inline]
```

References GetScalarType().

12.235.4.18 Print()

```
void gdcm::PixelFormat::Print (
            std::ostream & os) const
```

Print.

Referenced by operator<<.

12.235.4.19 SetBitsAllocated()

```
void gdcm::PixelFormat::SetBitsAllocated (
            unsigned short ba)  [inline]
```

12.235.4.20 SetBitsStored()

```
void gdcm::PixelFormat::SetBitsStored (
            unsigned short bs)  [inline]
```

References SetHighBit().

12.235.4.21 SetHighBit()

```
void gdcm::PixelFormat::SetHighBit (
            unsigned short hb)  [inline]
```

Referenced by SetBitsStored().

12.235.4.22 SetPixelRepresentation()

```
void gdcm::PixelFormat::SetPixelRepresentation (
            unsigned short pr)  [inline]
```

Examples

TemplateEmptyImage.cxx.

12.235.4.23 SetSamplesPerPixel()

```
void gdcm::PixelFormat::SetSamplesPerPixel (
            unsigned short spp)  [inline]
```

Examples

CreateARGBImage.cxx, and CreateCMYKImage.cxx.

References gdcmAssertMacro.

12.235.4.24 SetScalarType()

```
void gdcm::PixelFormat::SetScalarType (
            ScalarType st)
```

Set PixelFormat based only on the ScalarType

Warning

: You need to call SetScalarType *before* SetSamplesPerPixel

12.235.4.25 Validate()

```
bool gdcm::PixelFormat::Validate () [protected]
```

When image with 24/24/23 was read, need to validate.

12.235.5 Friends And Related Symbol Documentation

12.235.5.1 Bitmap

```
friend class Bitmap [friend]
```

References PixelFormat(), Bitmap, and operator<<.

Referenced by Bitmap.

12.235.5.2 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const PixelFormat & pf) [friend]
```

References PixelFormat(), and Print().

Referenced by Bitmap.

The documentation for this class was generated from the following file:

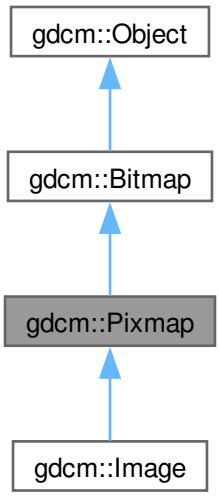
- gdcmPixelFormat.h

12.236 gdcm::Pixmap Class Reference

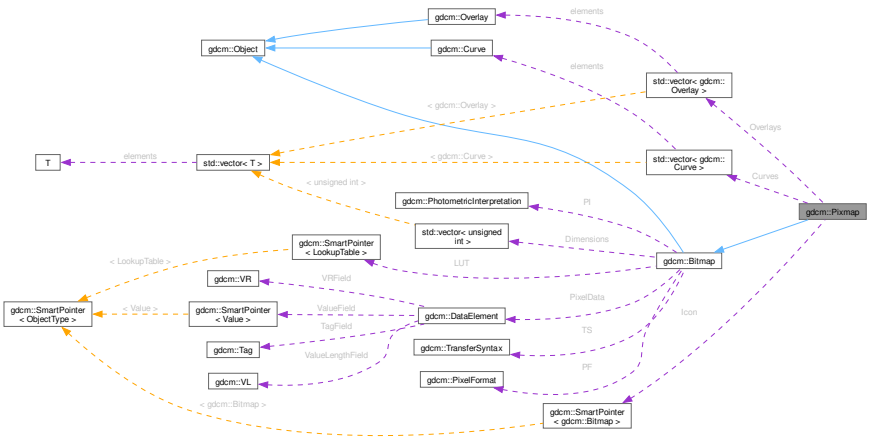
Pixmap class.

```
#include <gdcmPixmap.h>
```

Inheritance diagram for gdcm::Pixmap:



Collaboration diagram for gdcm::Pixmap:



Public Member Functions

- Pixmap ()
- ~Pixmap () override
- bool AreOverlaysInPixelData () const override

  returns if Overlays are stored in the unused bit of the pixel data:
- Curve & GetCurve (size_t i=0)

  Curve: group 50xx.
- const Curve & GetCurve (size_t i=0) const
- IconImage & GetIconImage ()
- const IconImage & GetIconImage () const

  Set/Get Icon Image.
- size_t GetNumberOfCurves () const
- size_t GetNumberOfOverlays () const
- Overlay & GetOverlay (size_t i=0)

  Overlay: group 60xx.
- const Overlay & GetOverlay (size_t i=0) const
- void Print (std::ostream &) const override
- void RemoveOverlay (size_t i)
- void SetIconImage (IconImage const &ii)
- void SetNumberOfCurves (size_t n)
- void SetNumberOfOverlays (size_t n)
- bool UnusedBitsPresentInPixelData () const override

  returns if there are unused bits in the pixel data

Public Member Functions inherited from gdcm::Bitmap

- Bitmap ()
- ~Bitmap () override
- void Clear ()
- bool GetBuffer (char *buffer) const

  Access the raw data.
- unsigned long GetBufferLength () const
- unsigned int GetColumns () const
- DataElement & GetDataElement ()
- const DataElement & GetDataElement () const
- unsigned int GetDimension (unsigned int idx) const
- const unsigned int * GetDimensions () const

  Return the dimension of the pixel data, first dimension (x), then 2nd (y), then 3rd (z)...
- LookupTable & GetLUT ()
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const

  INTERNAL do not use.
- unsigned int GetNumberOfDimensions () const

  Return the number of dimension of the pixel data bytes; for example 2 for a 2D matrices of values.
- const PhotometricInterpretation & GetPhotometricInterpretation () const

  return the photometric interpretation
- PixelFormat & GetPixelFormat ()

- const PixelFormat & GetPixelFormat () const

  Get/Set PixelFormat.
- unsigned int GetPlanarConfiguration () const

  return the planar configuration
- unsigned int GetRows () const
- const TransferSyntax & GetTransferSyntax () const
- bool IsEmpty () const
- bool IsLossy () const

  Return whether or not the image was compressed using a lossy compressor or not.
- bool IsTransferSyntaxCompatible (TransferSyntax const &ts) const
- void SetColumns (unsigned int col)
- void SetDataElement (DataElement const &de)
- void SetDimension (unsigned int idx, unsigned int dim)
- void SetDimensions (const unsigned int dims[3])
- void SetLossyFlag (bool f)

  Specifically set that the image was compressed using a lossy compression mechanism.
- void SetLUT (LookupTable const &lut)

  Set/Get LUT.
- void SetNeedByteSwap (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)
- void SetRows (unsigned int rows)
- void SetTransferSyntax (TransferSyntax const &ts)

  Transfer syntax.

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Protected Attributes

- std::vector< Curve > Curves
- SmartPointer< IconImage > Icon
- std::vector< Overlay > Overlays

Protected Attributes inherited from gdcm::Bitmap

- std::vector< unsigned int > Dimensions
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- DataElement PixelData
- unsigned int PlanarConfiguration
- TransferSyntax TS

Additional Inherited Members

Protected Types inherited from gdcm::Bitmap

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions inherited from gdcm::Bitmap

- bool ComputeLossyFlag ()
- bool GetBuffer2 (std::ostream &os) const
- bool TryJPEG2000Codec (char *buffer, bool &lossyflag) const
- bool TryJPEG2000Codec2 (std::ostream &os) const
- bool TryJPEGCodec (char *buffer, bool &lossyflag) const
- bool TryJPEGCodec2 (std::ostream &os) const
- bool TryJPEGLSCodec (char *buffer, bool &lossyflag) const
- bool TryKAKADUCodec (char *buffer, bool &lossyflag) const
- bool TryPVRGCodec (char *buffer, bool &lossyflag) const
- bool TryRAWCodec (char *buffer, bool &lossyflag) const
- bool TryRLECodec (char *buffer, bool &lossyflag) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.236.1 Detailed Description

Pixmap class.

A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

See also

PixmapReader

Examples

FileChangeTS.cs, FileChangeTSLossy.cs, and StandardizeFiles.cs.

12.236.2 Constructor & Destructor Documentation

12.236.2.1 Pixmap()

```
gdcm::Pixmap::Pixmap ()
```

12.236.2.2 ~Pixmap()

```
gdcm::Pixmap::~Pixmap ()  [override]
```

12.236.3 Member Function Documentation

12.236.3.1 AreOverlaysInPixelData()

```
bool gdcm::Pixmap::AreOverlaysInPixelData () const  [override], [virtual]
```

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from gdcm::Bitmap.

12.236.3.2 GetCurve() [1/2]

```
Curve & gdcm::Pixmap::GetCurve (
            size_t i = 0)  [inline]
```

Curve: group 50xx.

References Curves.

12.236.3.3 GetCurve() [2/2]

```
const Curve & gdcm::Pixmap::GetCurve (
            size_t i = 0) const  [inline]
```

References Curves.

12.236.3.4 GetIconImage() [1/2]

```
IconImage & gdcm::Pixmap::GetIconImage ()  [inline]
```

References Icon.

12.236.3.5 GetIconImage() [2/2]

```
const IconImage & gdcm::Pixmap::GetIconImage () const  [inline]
```

Set/Get Icon Image.

References Icon.

12.236.3.6 GetNumberOfCurves()

```
size_t gdcm::Pixmap::GetNumberOfCurves () const  [inline]
```

References Curves.

12.236.3.7 GetNumberOfOverlays()

```
size_t gdcm::Pixmap::GetNumberOfOverlays () const  [inline]
```

References Overlays.

12.236.3.8 GetOverlay() [1/2]

```
Overlay & gdcm::Pixmap::GetOverlay (
            size_t i = 0)  [inline]
```

Overlay: group 60xx.

References Overlays.

12.236.3.9 GetOverlay() [2/2]

```
const Overlay & gdcm::Pixmap::GetOverlay (
            size_t i = 0) const  [inline]
```

References Overlays.

12.236.3.10 Print()

```
void gdcm::Pixmap::Print (
            std::ostream & ) const  [override], [virtual]
```

Reimplemented from gdcm::Bitmap.

12.236.3.11 RemoveOverlay()

```
void gdcm::Pixmap::RemoveOverlay (
            size_t i)  [inline]
```

References Overlays.

12.236.3.12 SetIconImage()

```
void gdcm::Pixmap::SetIconImage (
            IconImage const & ii)  [inline]
```

References Icon.

12.236.3.13 SetNumberOfCurves()

```
void gdcm::Pixmap::SetNumberOfCurves (
            size_t n)  [inline]
```

References Curves.

12.236.3.14 SetNumberOfOverlays()

```
void gdcm::Pixmap::SetNumberOfOverlays (
            size_t n)  [inline]
```

References Overlays.

12.236.3.15 UnusedBitsPresentInPixelData()

```
bool gdcm::Pixmap::UnusedBitsPresentInPixelData () const  [override], [virtual]
```

returns if there are unused bits in the pixel data

Reimplemented from gdcm::Bitmap.

12.236.4 Member Data Documentation

12.236.4.1 Curves

```
std::vector<Curve> gdcm::Pixmap::Curves  [protected]
```

Referenced by GetCurve(), GetCurve(), GetNumberOfCurves(), and SetNumberOfCurves().

12.236.4.2 Icon

```
SmartPointer<IconImage> gdcm::Pixmap::Icon [protected]
```

Referenced by GetIconImage(), GetIconImage(), and SetIconImage().

12.236.4.3 Overlays

```
std::vector<Overlay> gdcm::Pixmap::Overlays [protected]
```

Referenced by GetNumberOfOverlays(), GetOverlay(), GetOverlay(), RemoveOverlay(), and SetNumberOfOverlays().

The documentation for this class was generated from the following file:

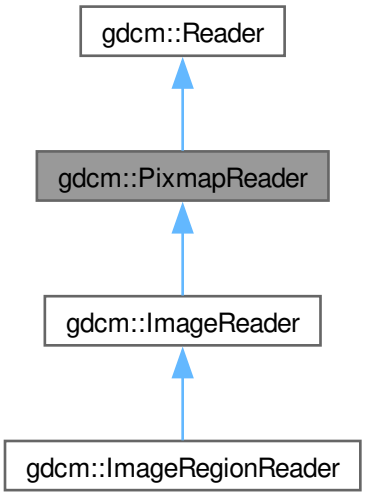
- gdcmPixmap.h

12.237 gdcm::PixmapReader Class Reference

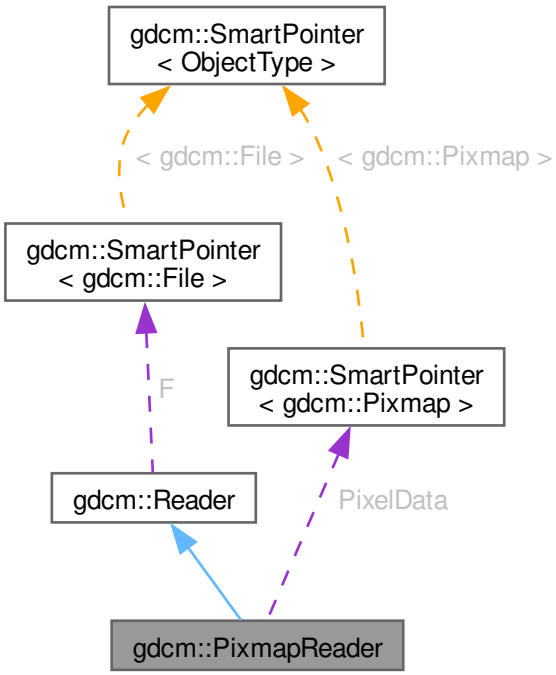
PixmapReader.

```
#include <gdcmPixmapReader.h>
```

Inheritance diagram for gdcm::PixmapReader:



Collaboration diagram for gdcm::PixmapReader:



Public Member Functions

- PixmapReader ()
- ~PixmapReader () override
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const

  Return the read image (need to call Read() first).
- bool Read () override

Public Member Functions inherited from gdcm::Reader

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- File & GetFile ()

  Set/Get File.
- const File & GetFile () const

  Set/Get File.
- size_t GetStreamCurrentPosition () const

- bool ReadSelectedPrivateTags (std::set< PrivateTag > const &ptags, bool readvalues=true)

  Will only read the specified selected private tags.
- bool ReadSelectedTags (std::set< Tag > const &tags, bool readvalues=true)

  Will only read the specified selected tags.
- bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())
- void SetFile (File &file)

  Set/Get File.
- void SetFileName (const char *filename_native)
- void SetStream (std::istream &input_stream)

  Set the open-ed stream directly.

Protected Member Functions

- virtual bool ReadACRNEMAImage ()
- virtual bool ReadImage (MediaStorage const &ms)
- bool ReadImageInternal (MediaStorage const &ms, bool handlepixeldata=true)

Protected Member Functions inherited from gdcm::Reader

- std::istream * GetStreamPtr () const
- bool ReadDataSet ()
- bool ReadMetaInformation ()
- bool ReadPreamble ()

Protected Attributes

- SmartPointer< Pixmap > PixelData

Protected Attributes inherited from gdcm::Reader

- SmartPointer< File > F

12.237.1 Detailed Description

PixmapReader.

Note

its role is to convert the DICOM DataSet into a Pixmap representation By default it is also loading the lookup table and overlay when found as they impact the rendering or the image

See PS 3.3-2008, Table C.7-11b IMAGE PIXEL MACRO ATTRIBUTES for the list of attribute that belong to what gdcm calls a 'Pixmap'

Warning

the API ReadUpToTag and ReadSelectedTag

See also

Pixmap

Examples

StandardizeFiles.cs.

12.237.2 Constructor & Destructor Documentation

12.237.2.1 PixmapReader()

```
gdcm::PixmapReader::PixmapReader ()
```

12.237.2.2 ~PixmapReader()

```
gdcm::PixmapReader::~PixmapReader () [override]
```

12.237.3 Member Function Documentation

12.237.3.1 GetPixmap() [1/2]

```
Pixmap & gdcm::PixmapReader::GetPixmap ()
```

12.237.3.2 GetPixmap() [2/2]

```
const Pixmap & gdcm::PixmapReader::GetPixmap () const
```

Return the read image (need to call Read() first).

Examples

StandardizeFiles.cs.

12.237.3.3 Read()

```
bool gdcm::PixmapReader::Read () [override], [virtual]
```

Read the DICOM image. There are two reason for failure:

1. The input filename is not DICOM
2. The input DICOM file does not contains an Pixmap.

Reimplemented from gdcm::Reader.

Examples

StandardizeFiles.cs.

12.237.3.4 ReadACRNEMAImage()

```
virtual bool gdcm::PixmapReader::ReadACRNEMAImage ()  [protected], [virtual]
```

Reimplemented in gdcm::ImageReader.

12.237.3.5 ReadImage()

```
virtual bool gdcm::PixmapReader::ReadImage (
            MediaStorage const & ms)  [protected], [virtual]
```

Reimplemented in gdcm::ImageReader.

12.237.3.6 ReadImageInternal()

```
bool gdcm::PixmapReader::ReadImageInternal (
            MediaStorage const & ms,
            bool handlepixeldata = true)  [protected]
```

12.237.4 Member Data Documentation

12.237.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapReader::PixelData  [protected]
```

The documentation for this class was generated from the following file:

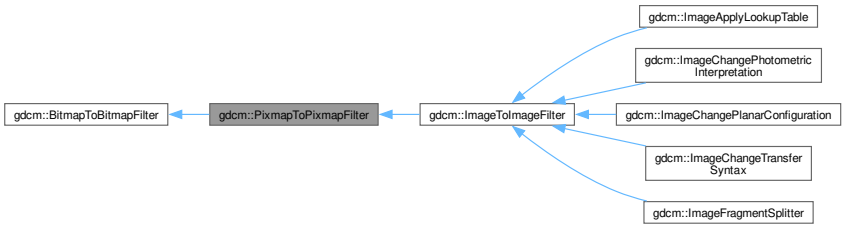
- gdcmPixmapReader.h

12.238 gdcm::PixmapToPixmapFilter Class Reference

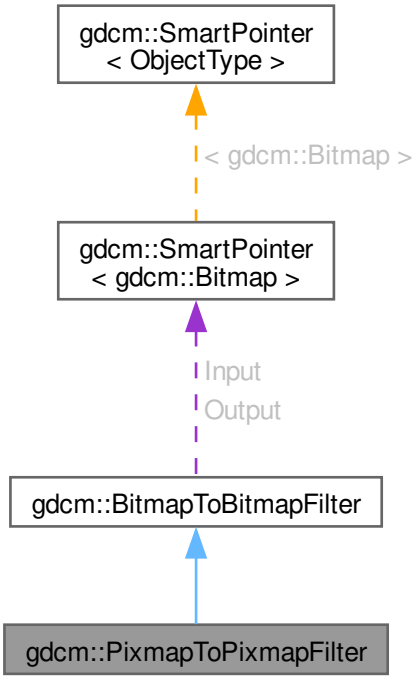
PixmapToPixmapFilter class.

```
#include <gdcmPixmapToPixmapFilter.h>
```

Inheritance diagram for gdcm::PixmapToPixmapFilter:



Collaboration diagram for gdcm::PixmapToPixmapFilter:



Public Member Functions

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()=default
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Public Member Functions inherited from gdcm::BitmapToBitmapFilter

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()=default
- const Bitmap & GetOutput () const

  Get Output image.
- const Bitmap & GetOutputAsBitmap () const
- void SetInput (const Bitmap &image)

  Set input image.

Additional Inherited Members

Protected Attributes inherited from gdcm::BitmapToBitmapFilter

- SmartPointer< Bitmap > Input
- SmartPointer< Bitmap > Output

12.238.1 Detailed Description

PixmapToPixmapFilter class.

Super class for all filter taking an image and producing an output image

Examples

StandardizeFiles.cs.

12.238.2 Constructor & Destructor Documentation

12.238.2.1 PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ()
```

12.238.2.2 ~PixmapToPixmapFilter()

```
gdcm::PixmapToPixmapFilter::~PixmapToPixmapFilter () [default]
```

12.238.3 Member Function Documentation

12.238.3.1 GetInput()

```
Pixmap & gdcm::PixmapToPixmapFilter::GetInput ()
```

12.238.3.2 GetOutput()

```
const Pixmap & gdcm::PixmapToPixmapFilter::GetOutput () const
```

Get Output image.

12.238.3.3 GetOutputAsPixmap()

```
const Pixmap & gdcm::PixmapToPixmapFilter::GetOutputAsPixmap () const
```

The documentation for this class was generated from the following file:

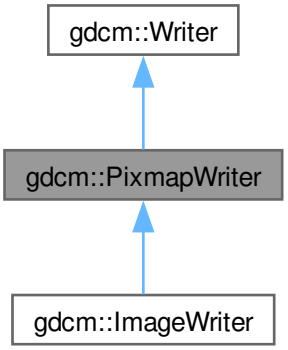
- gdcmPixmapToPixmapFilter.h

12.239 gdcm::PixmapWriter Class Reference

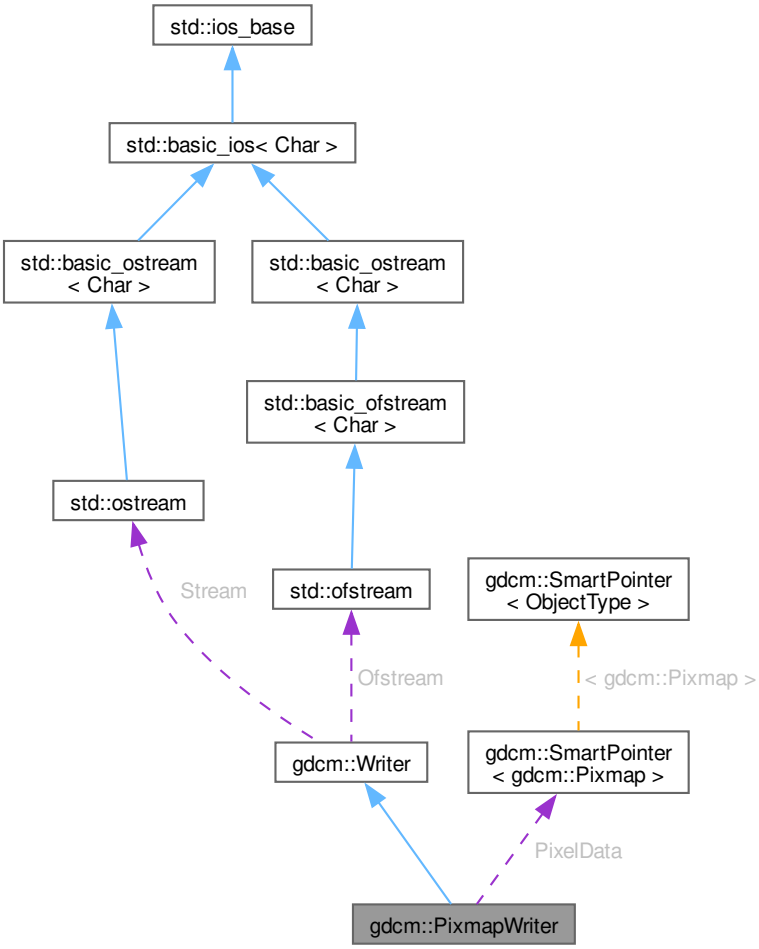
PixmapWriter.

```
#include <gdcmPixmapWriter.h>
```

Inheritance diagram for gdcm::PixmapWriter:



Collaboration diagram for gdcm::PixmapWriter:



Public Member Functions

- PixmapWriter ()
- ~PixmapWriter () override
- virtual Pixmap & GetImage ()
- virtual const Pixmap & GetImage () const
- Pixmap & GetPixmap ()
- const Pixmap & GetPixmap () const
- virtual void SetImage (Pixmap const &img)
- void SetPixmap (Pixmap const &img)
- bool Write () override

  Write.

Public Member Functions inherited from gdcm::Writer

- Writer ()
- virtual ~Writer ()
- void CheckFileMetaInformationOff ()
- void CheckFileMetaInformationOn ()
- File & GetFile ()
- void SetCheckFileMetaInformation (bool b)

  Undocumented function, do not use (= leave default).
- void SetFile (const File &f)

  Set/Get the DICOM file (DataSet + Header).
- void SetFileName (const char *filename_native)

  Set the filename of DICOM file to write:
- void SetStream (std::ostream &output_stream)

  Set user ostream buffer.

Protected Member Functions

- void DoIconImage (DataSet &ds, Pixmap const &image)
- bool PrepareWrite (MediaStorage const &refms)

Protected Member Functions inherited from gdcm::Writer

- bool GetCheckFileMetaInformation () const
- std::ostream * GetStreamPtr () const
- void SetWriteDataSetOnly (bool b)

Protected Attributes

- SmartPointer< Pixmap > PixelData

Protected Attributes inherited from gdcm::Writer

- std::ofstream * Ofstream
- std::ostream * Stream

12.239.1 Detailed Description

PixmapWriter.

This class will takes two inputs:

1. The DICOM DataSet
2. The Image input It will override any info from the Image over the DataSet.

For instance when one read in a lossy compressed image and write out as unencapsulated (ie implicitly lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

Examples

StandardizeFiles.cs.

12.239.2 Constructor & Destructor Documentation

12.239.2.1 PixmapWriter()

```
gdcm::PixmapWriter::PixmapWriter ()
```

12.239.2.2 ~PixmapWriter()

```
gdcm::PixmapWriter::~PixmapWriter ()  [override]
```

12.239.3 Member Function Documentation

12.239.3.1 DoIconImage()

```
void gdcm::PixmapWriter::DoIconImage (
            DataSet & ds,
            Pixmap const & image)  [protected]
```

12.239.3.2 GetImage() [1/2]

```
virtual Pixmap & gdcm::PixmapWriter::GetImage ()  [inline], [virtual]
```

Reimplemented in gdcm::ImageWriter.

References PixelData.

12.239.3.3 GetImage() [2/2]

```
virtual const Pixmap & gdcm::PixmapWriter::GetImage () const  [inline], [virtual]
```

Set/Get Pixmap to be written It will overwrite anything Pixmap infos found in DataSet (see parent class to see how to pass dataset)

Reimplemented in gdcm::ImageWriter.

References PixelData.

12.239.3.4 GetPixmap() [1/2]

```
Pixmap & gdcm::PixmapWriter::GetPixmap ()  [inline]
```

References PixelData.

12.239.3.5 GetPixmap() [2/2]

```
const Pixmap & gdcm::PixmapWriter::GetPixmap () const  [inline]
```

References PixelData.

12.239.3.6 PrepareWrite()

```
bool gdcm::PixmapWriter::PrepareWrite (
            MediaStorage const & refms)  [protected]
```

12.239.3.7 SetImage()

```
virtual void gdcm::PixmapWriter::SetImage (
            Pixmap const & img)  [virtual]
```

Examples

BasicImageAnonymizer.cs, CompressImage.cxx, CompressLossyJPEG.cs, DecompressImage.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, and TemplateEmptyImage.cxx.

12.239.3.8 SetPixmap()

```
void gdcm::PixmapWriter::SetPixmap (
            Pixmap const & img)
```

Examples

StandardizeFiles.cs.

12.239.3.9 Write()

```
bool gdcm::PixmapWriter::Write ()  [override], [virtual]
```

Write.

Reimplemented from gdcm::Writer.

Examples

StandardizeFiles.cs.

12.239.4 Member Data Documentation

12.239.4.1 PixelData

```
SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData [protected]
```

Referenced by GetImage(), GetImage(), GetPixmap(), and GetPixmap().

The documentation for this class was generated from the following file:

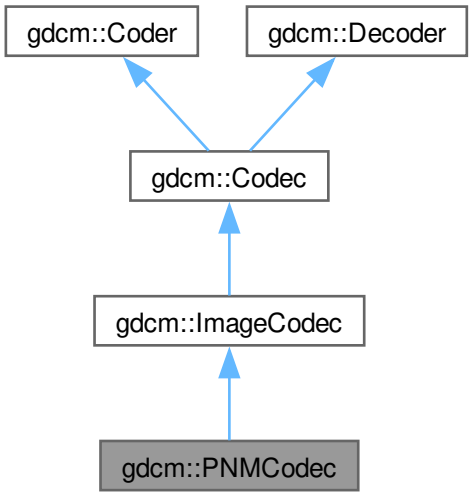
- gdcmPixmapWriter.h

12.240 gdcm::PNMCodec Class Reference

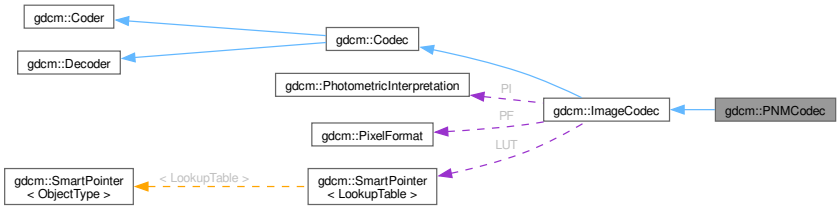
Class to do PNM.

```
#include <gdcmPNMCodec.h>
```

Inheritance diagram for gdcm::PNMCodec:



Collaboration diagram for gdcm::PNMCodec:



Public Member Functions

- PNMCodec ()
- ~PNMCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- bool Read (const char *filename, DataElement &out) const
- void SetBufferLength (unsigned long l)
- bool Write (const char *filename, const DataElement &out) const

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- bool Decode (DataElement const &is_, DataElement &os) override

  Decode.
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)

- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default
- virtual bool Code (DataElement const &in_, DataElement &out_)

  Code.

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions inherited from gdcm::ImageCodec

- virtual bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- virtual bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is_, std::ostream &os) override
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsFrameEncoder ()
- virtual bool IsRowEncoder ()
- virtual bool IsValid (PhotometricInterpretation const &pi)
- virtual bool StartEncode (std::ostream &os)
- virtual bool StopEncode (std::ostream &os)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.240.1 Detailed Description

Class to do PNM.

PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/`

Note

Only support P5 & P6 PNM file (binary grayscale and binary rgb)

Examples

ExtractIconFromFile.cxx.

12.240.2 Constructor & Destructor Documentation

12.240.2.1 PNMCodec()

```
gdcm::PNMCodec::PNMCodec ()
```

12.240.2.2 ~PNMCodec()

```
gdcm::PNMCodec::~PNMCodec ()  [override]
```

12.240.3 Member Function Documentation

12.240.3.1 CanCode()

```
bool gdcm::PNMCodec::CanCode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

12.240.3.2 CanDecode()

```
bool gdcm::PNMCodec::CanDecode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.240.3.3 Clone()

```
ImageCodec * gdcm::PNMCodec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.240.3.4 GetBufferLength()

```
unsigned long gdcm::PNMCodec::GetBufferLength () const  [inline]
```

12.240.3.5 GetHeaderInfo()

```
bool gdcm::PNMCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.240.3.6 Read()

```
bool gdcm::PNMCodec::Read (
            const char * filename,
            DataElement & out) const
```

12.240.3.7 SetBufferLength()

```
void gdcm::PNMCodec::SetBufferLength (
            unsigned long l)  [inline]
```

12.240.3.8 Write()

```
bool gdcm::PNMCodec::Write (
            const char * filename,
            const DataElement & out) const
```

Examples

ExtractIconFromFile.cxx.

The documentation for this class was generated from the following file:

- gdcmPNMCodec.h

12.241 gdcm::Preamble Class Reference

DICOM Preamble (Part 10).

```
#include <gdcmPreamble.h>
```

Public Member Functions

- Preamble ()
- Preamble (Preamble const &)
- ~Preamble ()
- void Clear ()

  Clear.
- void Create ()
- const char * GetInternal () const

  Get internal pointer to preamble.
- VL GetLength () const

  Return size of Preamble.
- bool IsEmpty () const

  Check if Preamble is empty.
- Preamble & operator= (Preamble const &)
- void Print (std::ostream &os) const

  Print Preamble.
- std::istream & Read (std::istream &is)

  Read Preamble.
- void Remove ()
- void Valid ()

  Set Preamble to the default one.
- std::ostream const & Write (std::ostream &os) const

  Write Preamble.

Protected Member Functions

- bool IsValid () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Preamble &_val)

12.241.1 Detailed Description

DICOM Preamble (Part 10).

12.241.2 Constructor & Destructor Documentation

12.241.2.1 Preamble() [1/2]

```
gdcm::Preamble::Preamble ()
```

Referenced by Preamble(), ~Preamble(), operator<<, and operator=().

12.241.2.2 ~Preamble()

```
gdcm::Preamble::~Preamble ()
```

References Preamble(), and operator<<.

12.241.2.3 Preamble() [2/2]

```
gdcm::Preamble::Preamble (
            Preamble const & )  [inline]
```

References Preamble(), and Create().

12.241.3 Member Function Documentation

12.241.3.1 Clear()

```
void gdcm::Preamble::Clear ()
```

Clear.

12.241.3.2 Create()

```
void gdcm::Preamble::Create ()
```

Referenced by Preamble(), and operator=().

12.241.3.3 GetInternal()

```
const char * gdcm::Preamble::GetInternal () const  [inline]
```

Get internal pointer to preamble.

12.241.3.4 GetLength()

```
VL gdcm::Preamble::GetLength () const  [inline]
```

Return size of Preamble.

12.241.3.5 IsEmpty()

```
bool gdcm::Preamble::IsEmpty () const  [inline]
```

Check if Preamble is empty.

12.241.3.6 IsValid()

```
bool gdcm::Preamble::IsValid () const  [inline], [protected]
```

12.241.3.7 operator=()

```
Preamble & gdcm::Preamble::operator= (
            Preamble const & )  [inline]
```

References Preamble(), and Create().

12.241.3.8 Print()

```
void gdcm::Preamble::Print (
            std::ostream & os) const
```

Print Preamble.

12.241.3.9 Read()

```
std::istream & gdcm::Preamble::Read (
            std::istream & is)
```

Read Preamble.

12.241.3.10 Remove()

```
void gdcm::Preamble::Remove ()
```

12.241.3.11 Valid()

```
void gdcm::Preamble::Valid ()
```

Set Preamble to the default one.

12.241.3.12 Write()

```
std::ostream const & gdcm::Preamble::Write (
            std::ostream & os) const
```

Write Preamble.

12.241.4 Friends And Related Symbol Documentation

12.241.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Preamble & _val)  [friend]
```

References Preamble().

Referenced by ~Preamble().

The documentation for this class was generated from the following file:

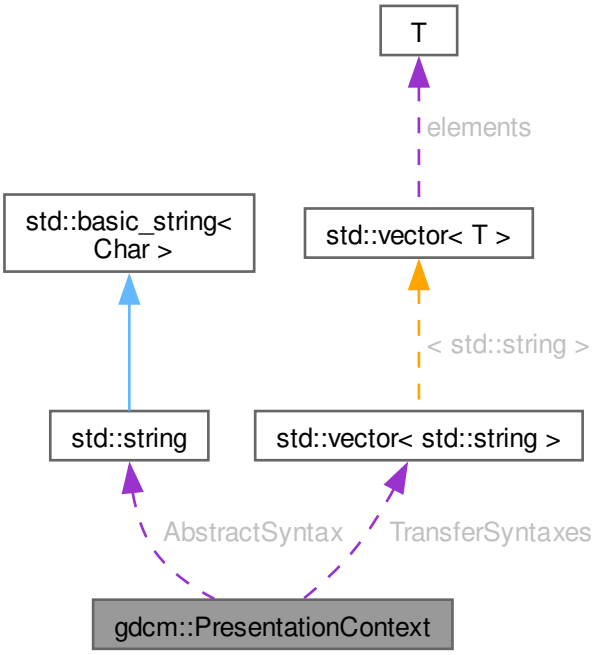
- gdcmPreamble.h

12.242 gdcm::PresentationContext Class Reference

PresentationContext.

```
#include <gdcmPresentationContext.h>
```

Collaboration diagram for gdcm::PresentationContext:



Public Types

- typedef TransferSyntaxArrayType::size_type SizeType
- typedef std::vector< std::string > TransferSyntaxArrayType

Public Member Functions

- PresentationContext ()
- PresentationContext (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
- void AddTransferSyntax (const char *tsstr)
- const char * GetAbstractSyntax () const
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- const char * GetTransferSyntax (SizeType i) const
- bool operator== (const PresentationContext &pc) const
- void Print (std::ostream &os) const
- void SetAbstractSyntax (const char *absyn)
- void SetPresentationContextID (uint8_t id)

Protected Attributes

- std::string AbstractSyntax
- uint8_t ID
- std::vector< std::string > TransferSyntaxes

12.242.1 Detailed Description

PresentationContext.

See also

PresentationContextAC PresentationContextRQ

12.242.2 Member Typedef Documentation

12.242.2.1 SizeType

```
typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType
```

12.242.2.2 TransferSyntaxArrayType

```
typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType
```

12.242.3 Constructor & Destructor Documentation

12.242.3.1 PresentationContext() [1/2]

```
gdcm::PresentationContext::PresentationContext ()
```

Referenced by operator==().

12.242.3.2 PresentationContext() [2/2]

```
gdcm::PresentationContext::PresentationContext (
            UIDs::TSName asname,
            UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
```

Initialize Presentation Context with AbstractSyntax set to asname and with a single TransferSyntax set to tsname (default to Implicit VR LittleEndian when not specified).

References gdcm::UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM.

12.242.4 Member Function Documentation

12.242.4.1 AddTransferSyntax()

```
void gdcm::PresentationContext::AddTransferSyntax (
            const char * tsstr)
```

12.242.4.2 GetAbstractSyntax()

```
const char * gdcm::PresentationContext::GetAbstractSyntax () const  [inline]
```

References AbstractSyntax.

12.242.4.3 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes () const  [inline]
```

References TransferSyntaxes.

12.242.4.4 GetPresentationContextID()

```
uint8_t gdcm::PresentationContext::GetPresentationContextID () const
```

12.242.4.5 GetTransferSyntax()

```
const char * gdcm::PresentationContext::GetTransferSyntax (
            SizeType i) const  [inline]
```

References TransferSyntaxes.

12.242.4.6 operator==()

```
bool gdcm::PresentationContext::operator== (
            const PresentationContext & pc) const  [inline]
```

References PresentationContext(), AbstractSyntax, and TransferSyntaxes.

12.242.4.7 Print()

```
void gdcm::PresentationContext::Print (
            std::ostream & os) const
```

12.242.4.8 SetAbstractSyntax()

```
void gdcm::PresentationContext::SetAbstractSyntax (
            const char * absyn)  [inline]
```

References AbstractSyntax.

12.242.4.9 SetPresentationContextID()

```
void gdcm::PresentationContext::SetPresentationContextID (
            uint8_t id)
```

12.242.5 Member Data Documentation

12.242.5.1 AbstractSyntax

```
std::string gdcm::PresentationContext::AbstractSyntax  [protected]
```

Referenced by GetAbstractSyntax(), operator==(), and SetAbstractSyntax().

12.242.5.2 ID

```
uint8_t gdcm::PresentationContext::ID  [protected]
```

12.242.5.3 TransferSyntaxes

```
std::vector<std::string> gdcm::PresentationContext::TransferSyntaxes  [protected]
```

Referenced by GetNumberOfTransferSyntaxes(), GetTransferSyntax(), and operator==().

The documentation for this class was generated from the following file:

- gdcmPresentationContext.h

12.243 gdcm::network::PresentationContextAC Class Reference

PresentationContextAC.

```
#include <gdcmPresentationContextAC.h>
```

Public Member Functions

- PresentationContextAC ()
- uint8_t GetPresentationContextID () const
- uint8_t GetReason () const
- TransferSyntaxSub const & GetTransferSyntax () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetPresentationContextID (uint8_t id)
- void SetReason (uint8_t r)
- void SetTransferSyntax (TransferSyntaxSub const &ts)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.243.1 Detailed Description

PresentationContextAC.

Table 9-18 PRESENTATION CONTEXT ITEM FIELDS

See also

PresentationContext

12.243.2 Constructor & Destructor Documentation

12.243.2.1 PresentationContextAC()

```
gdcm::network::PresentationContextAC::PresentationContextAC ()
```

12.243.3 Member Function Documentation

12.243.3.1 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID () const  [inline]
```

12.243.3.2 GetReason()

```
uint8_t gdcm::network::PresentationContextAC::GetReason () const  [inline]
```

12.243.3.3 GetTransferSyntax()

```
TransferSyntaxSub const & gdcm::network::PresentationContextAC::GetTransferSyntax () const  [inline]
```

12.243.3.4 Print()

```
void gdcm::network::PresentationContextAC::Print (
            std::ostream & os) const
```

12.243.3.5 Read()

```
std::istream & gdcm::network::PresentationContextAC::Read (
            std::istream & is)
```

12.243.3.6 SetPresentationContextID()

```
void gdcm::network::PresentationContextAC::SetPresentationContextID (
            uint8_t id)
```

12.243.3.7 SetReason()

```
void gdcm::network::PresentationContextAC::SetReason (
            uint8_t r)  [inline]
```

12.243.3.8 SetTransferSyntax()

```
void gdcm::network::PresentationContextAC::SetTransferSyntax (
            TransferSyntaxSub const & ts)
```

12.243.3.9 Size()

```
size_t gdcm::network::PresentationContextAC::Size () const
```

12.243.3.10 Write()

```
const std::ostream & gdcm::network::PresentationContextAC::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

- gdcmPresentationContextAC.h

12.244 gdcm::PresentationContextGenerator Class Reference

PresentationContextGenerator.

```
#include <gdcmPresentationContextGenerator.h>
```

Public Types

- typedef std::vector< PresentationContext > PresentationContextArrayType
- typedef PresentationContextArrayType::size_type SizeType

Public Member Functions

- PresentationContextGenerator ()
- bool AddFromFile (const File &file)
- bool GenerateFromFilenames (const Directory::FilenamesType &files)
- bool GenerateFromUID (UIDs::TSName asname)

  Generate the PresentationContext array from a UID (eg. VerificationSOPClass).
- PresentationContextArrayType const & GetPresentationContexts ()
- void SetDefaultTransferSyntax (const TransferSyntax &ts)

  Not implemented for now. GDCM internally uses Implicit Little Endian.
- void SetMergeModeToAbstractSyntax ()
- void SetMergeModeToTransferSyntax ()

Protected Member Functions

- bool AddPresentationContext (const char *absyn, const char *ts)
- const char * GetDefaultTransferSyntax () const

12.244.1 Detailed Description

PresentationContextGenerator.

This class is responsible for generating the proper PresentationContext that will be used in subsequent operation during a DICOM Query/Retrieve association. The step of the association is very sensible as special care need to be taken to explicitly define what instance are going to be send and how they are encoded.

For example a PresentationContext will express that negotiation requires that CT Image Storage are send using JPEG Lossless, while US Image Storage are sent using RLE Transfer Syntax.

Two very different API are exposed one which will always default to little endian transfer syntax see GenerateFromUID() This API is used for C-ECHO, C-FIND and C-MOVE (SCU). Another API: GenerateFromFilenames() is used for C-↩ STORE (SCU) as it will loop over all filenames argument to detect the actual encoding. and therefore find the proper encoding to be used.

Two modes are available. The default mode (SetMergeModeToAbstractSyntax) append PresentationContext (one AbstractSyntax and one TransferSyntax), as long a they are different. Eg MR Image Storage/JPEG2000 and MR Image Storage/JPEGLossless would be considered different. the other mode SetMergeModeToTransferSyntax merge any new TransferSyntax to the already existing PresentationContext in order to re-use the same AbstractSyntax.

See also

PresentationContext

Examples

CStoreQtProgress.cxx.

12.244.2 Member Typedef Documentation

12.244.2.1 PresentationContextArrayType

```
typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType
```

12.244.2.2 SizeType

```
typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType
```

12.244.3 Constructor & Destructor Documentation

12.244.3.1 PresentationContextGenerator()

```
gdcm::PresentationContextGenerator::PresentationContextGenerator ()
```

12.244.4 Member Function Documentation

12.244.4.1 AddFromFile()

```
bool gdcm::PresentationContextGenerator::AddFromFile (
            const File & file)
```

Add a single PresentationContext from a single File. Call multiple times when dealing with multiple files.

12.244.4.2 AddPresentationContext()

```
bool gdcm::PresentationContextGenerator::AddPresentationContext (
            const char * absyn,
            const char * ts)  [protected]
```

12.244.4.3 GenerateFromFilenames()

```
bool gdcm::PresentationContextGenerator::GenerateFromFilenames (
            const Directory::FilenamesType & files)
```

Generate the PresentationContext array from a File-Set. File specified needs to be valid DICOM files. Used for C-↩ STORE operations

Examples

CStoreQtProgress.cxx.

12.244.4.4 GenerateFromUID()

```
bool gdcm::PresentationContextGenerator::GenerateFromUID (
            UIDs::TSName asname)
```

Generate the PresentationContext array from a UID (eg. VerificationSOPClass).

12.244.4.5 GetDefaultTransferSyntax()

```
const char * gdcm::PresentationContextGenerator::GetDefaultTransferSyntax () const  [protected]
```

12.244.4.6 GetPresentationContexts()

```
PresentationContextArrayType const & gdcm::PresentationContextGenerator::GetPresentationContexts
() [inline]
```

Examples

CStoreQtProgress.cxx.

12.244.4.7 SetDefaultTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (
            const TransferSyntax & ts)
```

Not implemented for now. GDCM internally uses Implicit Little Endian.

12.244.4.8 SetMergeModeToAbstractSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ()
```

12.244.4.9 SetMergeModeToTransferSyntax()

```
void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ()
```

The documentation for this class was generated from the following file:

- gdcmPresentationContextGenerator.h

12.245 gdcm::network::PresentationContextRQ Class Reference

PresentationContextRQ.

```
#include <gdcmPresentationContextRQ.h>
```

Public Types

- typedef std::vector< TransferSyntaxSub >::size_type SizeType

Public Member Functions

- PresentationContextRQ ()
- PresentationContextRQ (const PresentationContext &pc)
- PresentationContextRQ (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDI
- void AddTransferSyntax (TransferSyntaxSub const &ts)
- AbstractSyntax & GetAbstractSyntax ()
- AbstractSyntax const & GetAbstractSyntax () const
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- TransferSyntaxSub & GetTransferSyntax (SizeType i)
- TransferSyntaxSub const & GetTransferSyntax (SizeType i) const
- std::vector< TransferSyntaxSub > const & GetTransferSyntaxes () const
- bool operator== (const PresentationContextRQ &pc) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetAbstractSyntax (AbstractSyntax const &absyn)
- void SetPresentationContextID (uint8_t id)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.245.1 Detailed Description

PresentationContextRQ.

Table 9-13 PRESENTATION CONTEXT ITEM FIELDS

See also

PresentationContextAC

12.245.2 Member Typedef Documentation

12.245.2.1 SizeType

```
typedef std::vector<TransferSyntaxSub>::size_type gdcm::network::PresentationContextRQ::SizeType
```

12.245.3 Constructor & Destructor Documentation

12.245.3.1 PresentationContextRQ() [1/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ ()
```

Referenced by operator==().

12.245.3.2 PresentationContextRQ() [2/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
            UIDs::TSName asname,
            UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
```

Initialize Presentation Context with AbstractSyntax set to asname and with a single TransferSyntax set to tsname (dfault to Implicit VR LittleEndian when not specified).

References gdcm::UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM.

12.245.3.3 PresentationContextRQ() [3/3]

```
gdcm::network::PresentationContextRQ::PresentationContextRQ (
            const PresentationContext & pc)
```

12.245.4 Member Function Documentation

12.245.4.1 AddTransferSyntax()

```
void gdcm::network::PresentationContextRQ::AddTransferSyntax (
            TransferSyntaxSub const & ts)
```

12.245.4.2 GetAbstractSyntax() [1/2]

```
AbstractSyntax & gdcm::network::PresentationContextRQ::GetAbstractSyntax ()  [inline]
```

12.245.4.3 GetAbstractSyntax() [2/2]

```
AbstractSyntax const & gdcm::network::PresentationContextRQ::GetAbstractSyntax () const  [inline]
```

12.245.4.4 GetNumberOfTransferSyntaxes()

```
SizeType gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes () const  [inline]
```

12.245.4.5 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID () const
```

12.245.4.6 GetTransferSyntax() [1/2]

```
TransferSyntaxSub & gdcm::network::PresentationContextRQ::GetTransferSyntax (
            SizeType i)  [inline]
```

12.245.4.7 GetTransferSyntax() [2/2]

```
TransferSyntaxSub const & gdcm::network::PresentationContextRQ::GetTransferSyntax (
            SizeType i) const  [inline]
```

12.245.4.8 GetTransferSyntaxes()

```
std::vector< TransferSyntaxSub > const & gdcm::network::PresentationContextRQ::GetTransfer↩
Syntaxes () const  [inline]
```

12.245.4.9 operator==()

```
bool gdcm::network::PresentationContextRQ::operator== (
            const PresentationContextRQ & pc) const  [inline]
```

References PresentationContextRQ().

12.245.4.10 Print()

```
void gdcm::network::PresentationContextRQ::Print (
            std::ostream & os) const
```

12.245.4.11 Read()

```
std::istream & gdcm::network::PresentationContextRQ::Read (
            std::istream & is)
```

12.245.4.12 SetAbstractSyntax()

```
void gdcm::network::PresentationContextRQ::SetAbstractSyntax (
            AbstractSyntax const & absyn)
```

12.245.4.13 SetPresentationContextID()

```
void gdcm::network::PresentationContextRQ::SetPresentationContextID (
            uint8_t id)
```

12.245.4.14 Size()

```
size_t gdcm::network::PresentationContextRQ::Size () const
```

12.245.4.15 Write()

```
const std::ostream & gdcm::network::PresentationContextRQ::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

- gdcmPresentationContextRQ.h

12.246 gdcm::network::PresentationDataValue Class Reference

PresentationDataValue.

```
#include <gdcmPresentationDataValue.h>
```

Public Member Functions

- PresentationDataValue ()
- const std::string & GetBlob () const
- bool GetIsCommand () const
- bool GetIsLastFragment () const
- uint8_t GetMessageHeader () const
- uint8_t GetPresentationContextID () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- std::istream & ReadInto (std::istream &is, std::ostream &os)
- void SetBlob (const std::string &partialblob)
- void SetCommand (bool inCommand)
- void SetDataSet (const DataSet &ds)
- void SetLastFragment (bool inLast)
- void SetMessageHeader (uint8_t messageheader)
- void SetPresentationContextID (uint8_t id)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

Static Public Member Functions

- static DataSet ConcatenatePDVBlobs (const std::vector< PresentationDataValue > &inPDVs)
- static DataSet ConcatenatePDVBlobsAsExplicit (const std::vector< PresentationDataValue > &inPDVs)

12.246.1 Detailed Description

PresentationDataValue.

Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS

12.246.2 Constructor & Destructor Documentation

12.246.2.1 PresentationDataValue()

```
gdcm::network::PresentationDataValue::PresentationDataValue ()
```

12.246.3 Member Function Documentation

12.246.3.1 ConcatenatePDVBlobs()

```
DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (
            const std::vector< PresentationDataValue > & inPDVs)  [static]
```

Warning

DataSet will be read as Implicit Little Endian TS

12.246.3.2 ConcatenatePDVBlobsAsExplicit()

```
DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (
            const std::vector< PresentationDataValue > & inPDVs)  [static]
```

12.246.3.3 GetBlob()

```
const std::string & gdcm::network::PresentationDataValue::GetBlob () const
```

12.246.3.4 GetIsCommand()

```
bool gdcm::network::PresentationDataValue::GetIsCommand () const
```

12.246.3.5 GetIsLastFragment()

```
bool gdcm::network::PresentationDataValue::GetIsLastFragment () const
```

12.246.3.6 GetMessageHeader()

```
uint8_t gdcm::network::PresentationDataValue::GetMessageHeader () const  [inline]
```

12.246.3.7 GetPresentationContextID()

```
uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID () const  [inline]
```

12.246.3.8 Print()

```
void gdcm::network::PresentationDataValue::Print (
            std::ostream & os) const
```

12.246.3.9 Read()

```
std::istream & gdcm::network::PresentationDataValue::Read (
            std::istream & is)
```

12.246.3.10 ReadInto()

```
std::istream & gdcm::network::PresentationDataValue::ReadInto (
            std::istream & is,
            std::ostream & os)
```

12.246.3.11 SetBlob()

```
void gdcm::network::PresentationDataValue::SetBlob (
            const std::string & partialblob)
```

12.246.3.12 SetCommand()

```
void gdcm::network::PresentationDataValue::SetCommand (
            bool inCommand)
```

12.246.3.13 SetDataSet()

```
void gdcm::network::PresentationDataValue::SetDataSet (
            const DataSet & ds)
```

Set DataSet. Write DataSet in implicit.

Warning

size of dataset should be below maxpdusize

12.246.3.14 SetLastFragment()

```
void gdcm::network::PresentationDataValue::SetLastFragment (
            bool inLast)
```

12.246.3.15 SetMessageHeader()

```
void gdcm::network::PresentationDataValue::SetMessageHeader (
            uint8_t messageheader)  [inline]
```

12.246.3.16 SetPresentationContextID()

```
void gdcm::network::PresentationDataValue::SetPresentationContextID (
            uint8_t id)  [inline]
```

12.246.3.17 Size()

```
size_t gdcm::network::PresentationDataValue::Size () const
```

12.246.3.18 Write()

```
const std::ostream & gdcm::network::PresentationDataValue::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

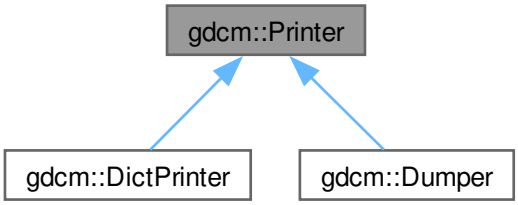
- gdcmPresentationDataValue.h

12.247 gdcm::Printer Class Reference

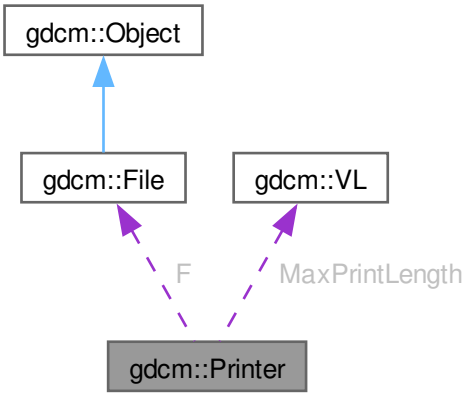
Printer class.

```
#include <gdcmPrinter.h>
```

Inheritance diagram for gdcm::Printer:



Collaboration diagram for gdcm::Printer:



Public Types

- enum PrintStyles {
  VERBOSE_STYLE = 0 ,
  CONDENSED_STYLE ,
  XML ,
  CXX }

Public Member Functions

- Printer ()
- ~Printer ()=default
- PrintStyles GetPrintStyle () const

  Get PrintStyle value.
- void Print (std::ostream &os)

  Print.
- void PrintDataSet (const DataSet &ds, std::ostream &os, const std::string &s="")

  Print an individual dataset.
- void SetColor (bool c)

  Set color mode or not.
- void SetFile (File const &f)

  Set file.
- void SetStyle (PrintStyles ps)

  Set PrintStyle value.

Protected Member Functions

- VR PrintDataElement (std::ostringstream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, std::ostream &out, std::string const &indent)
- void PrintSQ (const SequenceOfItems *sqi, std::ostream &os, std::string const &indent)

Protected Attributes

- const File * F
- VL MaxPrintLength
- PrintStyles PrintStyle

12.247.1 Detailed Description

Printer class.

Examples

DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, and DumpToshibaDTI2.cxx.

12.247.2 Member Enumeration Documentation

12.247.2.1 PrintStyles

```
enum gdcm::Printer::PrintStyles
```

Enumerator

VERBOSE_STYLE	
CONDENSED_STYLE	
XML	
CXX	

12.247.3 Constructor & Destructor Documentation

12.247.3.1 Printer()

```
gdcm::Printer::Printer ()
```

12.247.3.2 ~Printer()

```
gdcm::Printer::~Printer ()  [default]
```

12.247.4 Member Function Documentation

12.247.4.1 GetPrintStyle()

```
PrintStyles gdcm::Printer::GetPrintStyle () const  [inline]
```

Get PrintStyle value.

References PrintStyle.

12.247.4.2 Print()

```
void gdcm::Printer::Print (
            std::ostream & os)
```

Print.

Examples

DumpSiemensBase64.cxx.

12.247.4.3 PrintDataElement()

```
VR gdcm::Printer::PrintDataElement (
            std::ostringstream & os,
            const Dicts & dicts,
            const DataSet & ds,
            const DataElement & de,
            std::ostream & out,
            std::string const & indent)  [protected]
```

12.247.4.4 PrintDataSet()

```
void gdcm::Printer::PrintDataSet (
            const DataSet & ds,
            std::ostream & os,
            const std::string & s = "")
```

Print an individual dataset.

12.247.4.5 PrintSQ()

```
void gdcm::Printer::PrintSQ (
            const SequenceOfItems * sqi,
            std::ostream & os,
            std::string const & indent)  [protected]
```

12.247.4.6 SetColor()

```
void gdcm::Printer::SetColor (
            bool c)
```

Set color mode or not.

12.247.4.7 SetFile()

```
void gdcm::Printer::SetFile (
            File const & f)  [inline]
```

Set file.

Examples

DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, and DumpToshibaDTI2.cxx.

References F.

12.247.4.8 SetStyle()

```
void gdcm::Printer::SetStyle (
            PrintStyles ps) [inline]
```

Set PrintStyle value.

References PrintStyle.

12.247.5 Member Data Documentation

12.247.5.1 F

```
const File* gdcm::Printer::F [protected]
```

Referenced by SetFile().

12.247.5.2 MaxPrintLength

```
VL gdcm::Printer::MaxPrintLength [protected]
```

12.247.5.3 PrintStyle

```
PrintStyles gdcm::Printer::PrintStyle [protected]
```

Referenced by gdcm::Dumper::Dumper(), GetPrintStyle(), and SetStyle().

The documentation for this class was generated from the following file:

- gdcmPrinter.h

12.248 gdcm::PrivateDict Class Reference

Private Dict.

```
#include <gdcmDict.h>
```

Public Member Functions

- PrivateDict ()=default
- ~PrivateDict ()=default
- void AddDictEntry (const PrivateTag &tag, const DictEntry &de)
- bool FindDictEntry (const PrivateTag &tag) const
- const DictEntry & GetDictEntry (const PrivateTag &tag) const
- bool IsEmpty () const
- void PrintXML () const
- bool RemoveDictEntry (const PrivateTag &tag)

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &os, const PrivateDict &val)

12.248.1 Detailed Description

Private Dict.

12.248.2 Constructor & Destructor Documentation

12.248.2.1 PrivateDict()

```
gdcm::PrivateDict::PrivateDict () [default]
```

Referenced by LoadDefault(), and operator<<.

12.248.2.2 ~PrivateDict()

```
gdcm::PrivateDict::~PrivateDict () [default]
```

12.248.3 Member Function Documentation

12.248.3.1 AddDictEntry()

```
void gdcm::PrivateDict::AddDictEntry (
            const PrivateTag & tag,
            const DictEntry & de) [inline]
```

References GetDictEntry(), gdcm::DictEntry::GetVM(), gdcm::DictEntry::GetVR(), gdcm::DictEntry::SetVM(), gdcm::DictEntry::SetVR(), and gdcm::VR::UN.

12.248.3.2 FindDictEntry()

```
bool gdcm::PrivateDict::FindDictEntry (
            const PrivateTag & tag) const [inline]
```

12.248.3.3 GetDictEntry()

```
const DictEntry & gdcm::PrivateDict::GetDictEntry (
            const PrivateTag & tag) const  [inline]
```

Referenced by AddDictEntry().

12.248.3.4 IsEmpty()

```
bool gdcm::PrivateDict::IsEmpty () const  [inline]
```

12.248.3.5 LoadDefault()

```
void gdcm::PrivateDict::LoadDefault ()  [protected]
```

References PrivateDict().

12.248.3.6 PrintXML()

```
void gdcm::PrivateDict::PrintXML () const  [inline]
```

References gdcm::Tag::GetElement(), gdcm::Tag::GetGroup(), gdcm::DictEntry::GetName(), gdcm::PrivateTag::GetOwner(), gdcm::DictEntry::GetVM(), and gdcm::DictEntry::GetVR().

12.248.3.7 RemoveDictEntry()

```
bool gdcm::PrivateDict::RemoveDictEntry (
            const PrivateTag & tag)  [inline]
```

Remove entry 'tag'. Return true on success (element was found and remove). return false if element was not found.

12.248.4 Friends And Related Symbol Documentation

12.248.4.1 Dicts

```
friend class Dicts  [friend]
```

References Dicts.

Referenced by Dicts.

12.248.4.2 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const PrivateDict & val)  [friend]
```

References PrivateDict().

The documentation for this class was generated from the following file:

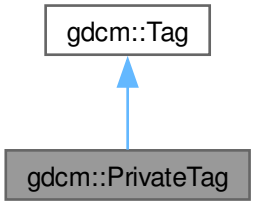
- gdcmDict.h

12.249 gdcm::PrivateTag Class Reference

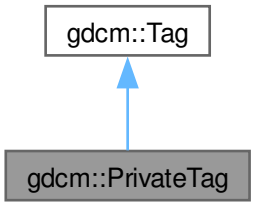
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner).

```
#include <gdcmPrivateTag.h>
```

Inheritance diagram for gdcm::PrivateTag:



Collaboration diagram for gdcm::PrivateTag:



Public Member Functions

- PrivateTag (Tag const &t, const char ∗owner="")
- PrivateTag (uint16_t group=0, uint16_t element=0, const char ∗owner="")
- DataElement GetAsDataElement () const
- const char ∗ GetOwner () const
- bool operator!= (const PrivateTag &_val) const
- bool operator!= (const Tag &_val) const
- bool operator< (const PrivateTag &_val) const
- PrivateTag & operator= (const PrivateTag &_val)
- bool operator== (const PrivateTag &_val) const
- bool operator== (const Tag &_val) const
- bool ReadFromCommaSeparatedString (const char ∗str)
- void SetOwner (const char ∗owner)

Public Member Functions inherited from gdcm::Tag

- Tag (const Tag &_val)
- Tag (uint16_t group, uint16_t element)

  Constructor with 2∗uint16_t.
- Tag (uint32_t tag=0)

  Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.
- uint16_t GetElement () const

  Returns the 'Element number' of the given Tag.
- uint32_t GetElementTag () const

  Returns the full tag value of the given Tag.
- uint16_t GetGroup () const

  Returns the 'Group number' of the given Tag.
- uint32_t GetLength () const

  return the length of tag (read: size on disk)
- Tag GetPrivateCreator () const

  Return the Private Creator Data Element tag of a private data element.
- bool IsGroupLength () const

  return whether the tag correspond to a group length tag:
- bool IsGroupXX (const Tag &t) const

  e.g 6002,3000 belong to groupXX: 6000,3000
- bool IsIllegal () const

  return if the tag is considered to be an illegal tag
- bool IsPrivate () const
- bool IsPrivateCreator () const
- bool IsPublic () const
- bool operator!= (const Tag &_val) const
- bool operator< (const Tag &_val) const
- bool operator<= (const Tag &t2) const
- Tag & operator= (const Tag &_val)
- bool operator== (const Tag &_val) const
- uint16_t & operator[] (const unsigned int &_id)

  Returns the Group or Element of the given Tag, depending on id (0/1).

- const uint16_t & operator[ ] (const unsigned int &_id) const

  Returns the Group or Element of the given Tag, depending on id (0/1).
- std::string PrintAsContinuousString () const
- std::string PrintAsContinuousUpperCaseString () const

  Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.
- std::string PrintAsPipeSeparatedString () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)

  Read a tag from binary representation.
- bool ReadFromCommaSeparatedString (const char *str)
- bool ReadFromContinuousString (const char *str)
- bool ReadFromPipeSeparatedString (const char *str)
- void SetElement (uint16_t element)

  Sets the 'Element number' of the given Tag.
- void SetElementTag (uint16_t group, uint16_t element)

  Sets the 'Group number' & 'Element number' of the given Tag.
- void SetElementTag (uint32_t tag)

  Sets the full tag value of the given Tag.
- void SetGroup (uint16_t group)

  Sets the 'Group number' of the given Tag.
- void SetPrivateCreator (Tag const &t)

  Set private creator:
- template<typename TSwap>
  const std::ostream & Write (std::ostream &os) const

  Write a tag in binary rep.

Friends

- std::ostream & operator<< (std::ostream &_os, const PrivateTag &_val)

12.249.1 Detailed Description

Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner).

Note

private tag have element value in: [0x10,0xff], for instance 0x0009,0x0000 is NOT a private tag

Examples

ChangePrivateTags.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, ELSCINT1WaveToText.cxx, FileStreaming.cs, GetSubSequenceData.cxx, MrProtocol.cxx, PublicDict.cxx, ReadGEMSSDO.cxx, csa2img.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.249.2 Constructor & Destructor Documentation

12.249.2.1 PrivateTag() [1/2]

```
gdcm::PrivateTag::PrivateTag (
            uint16_t group = 0,
            uint16_t element = 0,
            const char * owner = "")  [inline]
```

References gdcm::Tag::Tag(), and gdcm::Tag::SetElement().

Referenced by operator!=(), operator<(), operator<<, operator=(), and operator==().

12.249.2.2 PrivateTag() [2/2]

```
gdcm::PrivateTag::PrivateTag (
            Tag const & t,
            const char * owner = "")  [inline]
```

References gdcm::Tag::Tag(), gdcm::Tag::GetElement(), and gdcm::Tag::SetElement().

12.249.3 Member Function Documentation

12.249.3.1 GetAsDataElement()

```
DataElement gdcm::PrivateTag::GetAsDataElement () const
```

12.249.3.2 GetOwner()

```
const char * gdcm::PrivateTag::GetOwner () const  [inline]
```

Examples

PublicDict.cxx.

Referenced by gdcm::PrivateDict::PrintXML().

12.249.3.3 operator"!=() [1/2]

```
bool gdcm::PrivateTag::operator!= (
            const PrivateTag & _val) const  [inline]
```

References PrivateTag(), and gdcm::Tag::GetElementTag().

12.249.3.4 operator"!=() [2/2]

```
bool gdcm::PrivateTag::operator!= (
            const Tag & _val) const  [inline]
```

References gdcm::Tag::Tag(), and gdcm::Tag::GetElementTag().

12.249.3.5 operator<()

```
bool gdcm::PrivateTag::operator< (
            const PrivateTag & _val) const
```

References PrivateTag().

12.249.3.6 operator=()

```
PrivateTag & gdcm::PrivateTag::operator= (
            const PrivateTag & _val)  [inline]
```

References PrivateTag(), gdcm::Tag::GetElementTag(), and gdcm::Tag::SetElementTag().

12.249.3.7 operator==() [1/2]

```
bool gdcm::PrivateTag::operator== (
            const PrivateTag & _val) const  [inline]
```

References PrivateTag(), and gdcm::Tag::GetElementTag().

12.249.3.8 operator==() [2/2]

```
bool gdcm::PrivateTag::operator== (
            const Tag & _val) const  [inline]
```

References gdcm::Tag::Tag(), and gdcm::Tag::GetElementTag().

12.249.3.9 ReadFromCommaSeparatedString()

```
bool gdcm::PrivateTag::ReadFromCommaSeparatedString (
            const char * str)
```

Read PrivateTag from a string. Element number will be truncated to 8bits. Eg: "1234,5678,GDCM" is private tag: (1234,78,"GDCM")

12.249.3.10 SetOwner()

```
void gdcm::PrivateTag::SetOwner (
            const char * owner)  [inline]
```

References gdcm::String<'\\', 64 >::Trim().

12.249.4 Friends And Related Symbol Documentation

12.249.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const PrivateTag & _val)  [friend]
```

References PrivateTag(), and operator<<.

Referenced by operator<<.

The documentation for this class was generated from the following file:

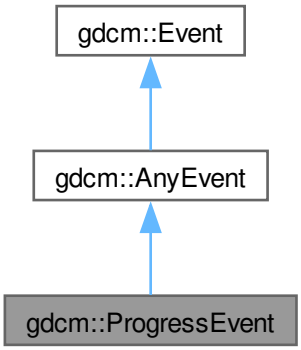
- gdcmPrivateTag.h

12.250 gdcm::ProgressEvent Class Reference

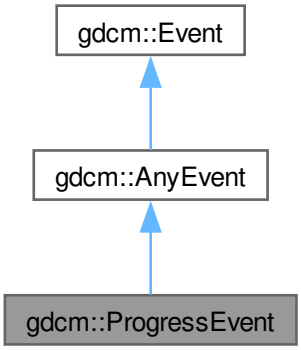
ProgressEvent.

```
#include <gdcmProgressEvent.h>
```

Inheritance diagram for gdcm::ProgressEvent:



Collaboration diagram for gdcm::ProgressEvent:



Public Types

- typedef ProgressEvent Self
- typedef AnyEvent Superclass

Public Member Functions

- ProgressEvent (const Self &s)
- ProgressEvent (double p=0)
- ~ProgressEvent () override=default
- bool CheckEvent (const ::gdcm::Event *e) const override
- const char * GetEventName () const override
- double GetProgress () const
- ::gdcm::Event * MakeObject () const override
- void operator= (const Self &)=delete
- void SetProgress (double p)

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

12.250.1 Detailed Description

ProgressEvent.

Special type of event triggered during

See also

AnyEvent

Examples

BasicAnonymizer.cs, Cleaner.cs, and ClinicalTrialIdentificationWorkflow.cs.

12.250.2 Member Typedef Documentation

12.250.2.1 Self

```
typedef ProgressEvent gdcm::ProgressEvent::Self
```

12.250.2.2 Superclass

```
typedef AnyEvent gdcm::ProgressEvent::Superclass
```

12.250.3 Constructor & Destructor Documentation

12.250.3.1 ProgressEvent() [1/2]

```
gdcm::ProgressEvent::ProgressEvent (
            double p = 0)  [inline]
```

12.250.3.2 ~ProgressEvent()

```
gdcm::ProgressEvent::~ProgressEvent ()  [override], [default]
```

12.250.3.3 ProgressEvent() [2/2]

```
gdcm::ProgressEvent::ProgressEvent (
            const Self & s)  [inline]
```

12.250.4 Member Function Documentation

12.250.4.1 CheckEvent()

```
bool gdcm::ProgressEvent::CheckEvent (
            const ::gdcm::Event * e) const  [inline], [override]
```

12.250.4.2 GetEventName()

```
const char * gdcm::ProgressEvent::GetEventName () const  [inline], [override], [virtual]
```

Return the StringName associated with the event.

Implements gdcm::Event.

12.250.4.3 GetProgress()

```
double gdcm::ProgressEvent::GetProgress () const  [inline]
```

Examples

BasicAnonymizer.cs, Cleaner.cs, and ClinicalTrialIdentificationWorkflow.cs.

12.250.4.4 MakeObject()

```
::gdcm::Event * gdcm::ProgressEvent::MakeObject () const  [inline], [override], [virtual]
```

Create an Event of this type This method work as a Factory for creating events of each particular type.

Implements gdcm::Event.

12.250.4.5 operator=()

```
void gdcm::ProgressEvent::operator= (
            const Self & )  [delete]
```

12.250.4.6 SetProgress()

```
void gdcm::ProgressEvent::SetProgress (
            double p)  [inline]
```

The documentation for this class was generated from the following file:

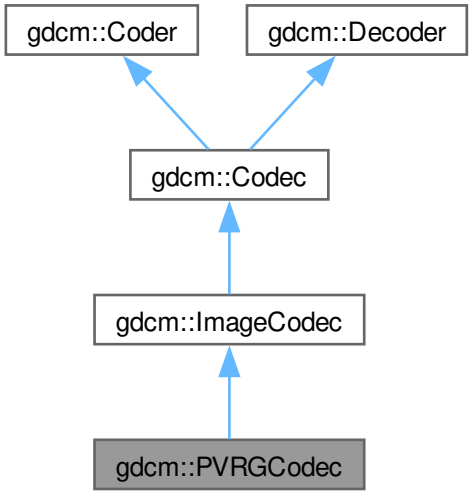
- gdcmProgressEvent.h

12.251 gdcm::PVRGCodec Class Reference

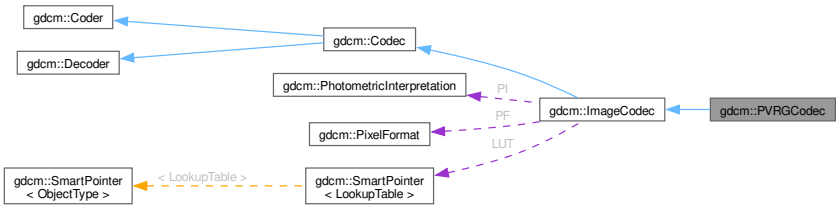
PVRGCodec.

```
#include <gdcmPVRGCodec.h>
```

Inheritance diagram for gdcm::PVRGCodec:



Collaboration diagram for gdcm::PVRGCodec:



Public Member Functions

- PVRGCodec ()
- ~PVRGCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).

- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- void SetLossyFlag (bool l)

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- virtual bool GetHeaderInfo (std::istream &is_, TransferSyntax &ts)
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Member Functions inherited from gdcm::ImageCodec

- virtual bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- virtual bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is_, std::ostream &os) override
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsFrameEncoder ()
- virtual bool IsRowEncoder ()
- virtual bool IsValid (PhotometricInterpretation const &pi)
- virtual bool StartEncode (std::ostream &os)
- virtual bool StopEncode (std::ostream &os)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.251.1 Detailed Description

PVRGCodec.

Note

pvrg is a broken implementation of the JPEG standard. It is known to have a bug in the 16bits lossless implementation of the standard.

In an ideal world, you should not need this codec at all. But to support some broken file such as:

PHILIPS_Gyroscan-12-Jpeg_Extended_Process_2_4.dcm

we have to...

12.251.2 Constructor & Destructor Documentation

12.251.2.1 PVRGCodec()

```
gdcm::PVRGCodec::PVRGCodec ()
```

12.251.2.2 ~PVRGCodec()

```
gdcm::PVRGCodec::~PVRGCodec ()  [override]
```

12.251.3 Member Function Documentation

12.251.3.1 CanCode()

```
bool gdcm::PVRGCodec::CanCode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

12.251.3.2 CanDecode()

```
bool gdcm::PVRGCodec::CanDecode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.251.3.3 Clone()

```
ImageCodec * gdcm::PVRGCodec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.251.3.4 Code()

```
bool gdcm::PVRGCodec::Code (
            DataElement const & in_,
            DataElement & out_)  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

12.251.3.5 Decode()

```
bool gdcm::PVRGCodec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

12.251.3.6 SetLossyFlag()

```
void gdcm::PVRGCodec::SetLossyFlag (
            bool l)
```

The documentation for this class was generated from the following file:

- gdcmPVRGCodec.h

12.252 gdcm::PythonFilter Class Reference

PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

```
#include <gdcmPythonFilter.h>
```

Public Member Functions

- PythonFilter ()
- ~PythonFilter ()
- File & GetFile ()
- const File & GetFile () const
- void SetDicts (const Dicts &dicts)
- void SetFile (const File &f)
- PyObject * ToPyObject (const Tag &t) const
- void UseDictAlways (bool)

12.252.1 Detailed Description

PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

12.252.2 Constructor & Destructor Documentation

12.252.2.1 PythonFilter()

```
gdcm::PythonFilter::PythonFilter ()
```

12.252.2.2 ~PythonFilter()

```
gdcm::PythonFilter::~PythonFilter ()
```

12.252.3 Member Function Documentation

12.252.3.1 GetFile() [1/2]

```
File & gdcm::PythonFilter::GetFile ()
```

12.252.3.2 GetFile() [2/2]

```
const File & gdcm::PythonFilter::GetFile () const
```

12.252.3.3 SetDicts()

```
void gdcm::PythonFilter::SetDicts (
            const Dicts & dicts)
```

12.252.3.4 SetFile()

```
void gdcm::PythonFilter::SetFile (
            const File & f)
```

12.252.3.5 ToPyObject()

```
PyObject * gdcm::PythonFilter::ToPyObject (
            const Tag & t) const
```

12.252.3.6 UseDictAlways()

```
void gdcm::PythonFilter::UseDictAlways (
            bool )  [inline]
```

The documentation for this class was generated from the following file:

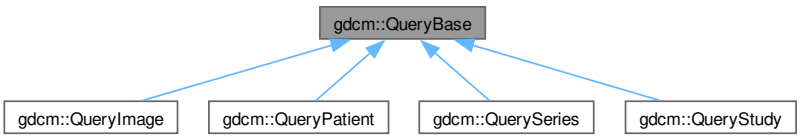
- gdcmPythonFilter.h

12.253 gdcm::QueryBase Class Reference

QueryBase.

```
#include <gdcmQueryBase.h>
```

Inheritance diagram for gdcm::QueryBase:



Public Member Functions

- virtual ~QueryBase ()=default
- std::vector< Tag > GetAllRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetAllTags (const ERootType &inRootType) const
- virtual std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const =0

  Return all Unique Key for a particular Query Root type (from the same level and above).
- virtual const char * GetName () const =0
- virtual std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const =0
- virtual DataElement GetQueryLevel () const =0
- virtual std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const =0
- virtual std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const =0

12.253.1 Detailed Description

QueryBase.

contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE

There are four levels of C-FIND and C-MOVE query:

- Patient
- Study
- Series
- Image

Each one has its own required and optional tags. This class provides an interface for getting those tags. This is an interface class.

See 3.4 C 6.1 and 3.4 C 6.2 for the patient and study root query types. These sections define the tags allowed by a particular query. The caller must pass in which root type they want, patient or study. A third root type, Modality Worklist Query, isn't yet supported.

This class (or rather it's derived classes) will be held in the RootQuery types. These query types actually make the dataset, and will use this dataset to list the required, unique, and optional tags for each type of query. This design is somewhat overly complicated, but is kept so that if we ever wanted to try to guess the query type from the given tags, we could do so.

12.253.2 Constructor & Destructor Documentation

12.253.2.1 ~QueryBase()

```
virtual gdcm::QueryBase::~QueryBase () [virtual], [default]
```

12.253.3 Member Function Documentation

12.253.3.1 GetAllRequiredTags()

```
std::vector< Tag > gdcm::QueryBase::GetAllRequiredTags (
            const ERootType & inRootType) const
```

In order to validate a query dataset we need to check that there exists at least one required (or unique) key

12.253.3.2 GetAllTags()

```
std::vector< Tag > gdcm::QueryBase::GetAllTags (
            const ERootType & inRootType) const
```

In order to validate a query dataset, just check for the presence of a tag, not it's requirement level in the spec

12.253.3.3 GetHierachicalSearchTags()

```
virtual std::vector< Tag > gdcm::QueryBase::GetHierachicalSearchTags (
            const ERootType & inRootType) const  [pure virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

12.253.3.4 GetName()

```
virtual const char * gdcm::QueryBase::GetName () const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

12.253.3.5 GetOptionalTags()

```
virtual std::vector< Tag > gdcm::QueryBase::GetOptionalTags (
            const ERootType & inRootType) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

12.253.3.6 GetQueryLevel()

```
virtual DataElement gdcm::QueryBase::GetQueryLevel () const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

12.253.3.7 GetRequiredTags()

```
virtual std::vector< Tag > gdcm::QueryBase::GetRequiredTags (
            const ERootType & inRootType) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

12.253.3.8 GetUniqueTags()

```
virtual std::vector< Tag > gdcm::QueryBase::GetUniqueTags (
            const ERootType & inRootType) const  [pure virtual]
```

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

The documentation for this class was generated from the following file:

- gdcmQueryBase.h

12.254 gdcm::QueryFactory Class Reference

QueryFactory.h.

```
#include <gdcmQueryFactory.h>
```

Static Public Member Functions

- static ECharSet GetCharacterFromCurrentLocale ()
- static void ListCharSets (std::ostream &os)

  List all possible CharSet.
- static DataElement ProduceCharacterSetDataElement (const std::vector< ECharSet > &inCharSetType)
- static BaseQuery * ProduceQuery (const std::string &sopInstanceUID, ENQueryType inQueryType)
- static BaseRootQuery * ProduceQuery (ERootType inRootType, EQueryType inQueryType, EQueryLevel in↩ QueryLevel)

12.254.1 Detailed Description

QueryFactory.h.

contains: a class to produce a query based off of user-entered information

Essentially, this class is used to construct a query based off of user input (typically from the command line; if in code directly, the query itself could just be instantiated)

In theory, could also be used as the interface to validate incoming datasets as belonging to a particular query style

12.254.2 Member Function Documentation

12.254.2.1 GetCharacterFromCurrentLocale()

```
ECharSet gdcm::QueryFactory::GetCharacterFromCurrentLocale ()  [static]
```

This function will return the corresponding ECharSet associated with the current locale of the running system (based on the value of locale()).

12.254.2.2 ListCharSets()

```
void gdcm::QueryFactory::ListCharSets (
            std::ostream & os)  [static]
```

List all possible CharSet.

12.254.2.3 ProduceCharacterSetDataElement()

```
DataElement gdcm::QueryFactory::ProduceCharacterSetDataElement (
            const std::vector< ECharSet > & inCharSetType) [static]
```

This function will produce the appropriate dataelement given a list of charsets. The first charset will be used directly, while the second and subsequent will be prepended with "ISO2022 ". Redundant character sets are not permitted, so if they are encountered, they will just be skipped. if UTF8 or GB18030 is used, no subsequent character sets will be used if the vector passed in is empty, then the dataelement that's passed out will be empty and Latin1 is the presumed encoding

12.254.2.4 ProduceQuery() [1/2]

```
BaseQuery * gdcm::QueryFactory::ProduceQuery (
            const std::string & sopInstanceUID,
            ENQueryType inQueryType) [static]
```

12.254.2.5 ProduceQuery() [2/2]

```
BaseRootQuery * gdcm::QueryFactory::ProduceQuery (
            ERootType inRootType,
            EQueryType inQueryType,
            EQueryLevel inQueryLevel) [static]
```

this function will produce a query (basically, a wrapper to a dataset that can validate whether or not the query is a valid cfind/cmove query) and the level of the query (patient, study, series, image). If the user provides an invalid instantiation (ie, study root type, query level of patient), then the result is NULL.

The documentation for this class was generated from the following file:

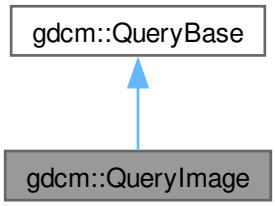
- gdcmQueryFactory.h

12.255 gdcm::QueryImage Class Reference

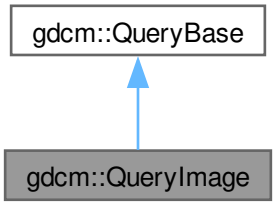
QueryImage.

```
#include <gdcmQueryImage.h>
```

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for gdcm::QueryImage:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const override
  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const override
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const override
- DataElement GetQueryLevel () const override
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const override
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const override

Public Member Functions inherited from gdcm::QueryBase

- virtual ~QueryBase ()=default
- std::vector< Tag > GetAllRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetAllTags (const ERootType &inRootType) const

12.255.1 Detailed Description

QueryImage.

contains: class to construct an image-based query for C-FIND and C-MOVE

12.255.2 Member Function Documentation

12.255.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryImage::GetHierachicalSearchTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

12.255.2.2 GetName()

```
const char * gdcm::QueryImage::GetName () const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.255.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryImage::GetOptionalTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.255.2.4 GetQueryLevel()

```
DataElement gdcm::QueryImage::GetQueryLevel () const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.255.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryImage::GetRequiredTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.255.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryImage::GetUniqueTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

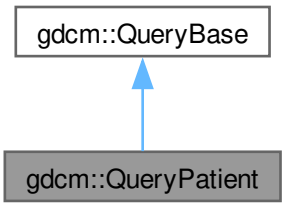
- gdcmQueryImage.h

12.256 gdcm::QueryPatient Class Reference

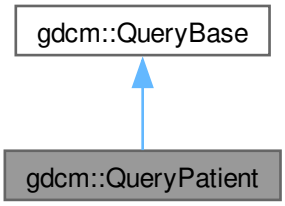
QueryPatient.

```
#include <gdcmQueryPatient.h>
```

Inheritance diagram for gdcm::QueryPatient:



Collaboration diagram for gdcm::QueryPatient:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const override

  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const override
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const override
- DataElement GetQueryLevel () const override
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const override
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const override

Public Member Functions inherited from gdcm::QueryBase

- virtual ~QueryBase ()=default
- std::vector< Tag > GetAllRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetAllTags (const ERootType &inRootType) const

12.256.1 Detailed Description

QueryPatient.

contains: class to construct a patient-based query for c-find and c-move

12.256.2 Member Function Documentation

12.256.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryPatient::GetHierachicalSearchTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

12.256.2.2 GetName()

```
const char * gdcm::QueryPatient::GetName () const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.256.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryPatient::GetOptionalTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.256.2.4 GetQueryLevel()

```
DataElement gdcm::QueryPatient::GetQueryLevel () const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.256.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryPatient::GetRequiredTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.256.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryPatient::GetUniqueTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

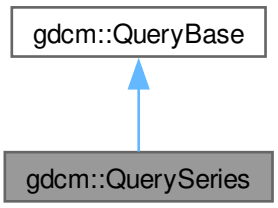
- gdcmQueryPatient.h

12.257 gdcm::QuerySeries Class Reference

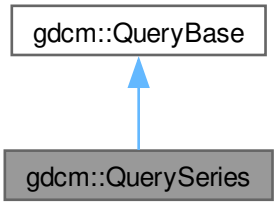
QuerySeries.

```
#include <gdcmQuerySeries.h>
```

Inheritance diagram for gdcm::QuerySeries:



Collaboration diagram for gdcm::QuerySeries:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const override
  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const override
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const override
- DataElement GetQueryLevel () const override
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const override
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const override

Public Member Functions inherited from gdcm::QueryBase

- virtual ~QueryBase ()=default
- std::vector< Tag > GetAllRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetAllTags (const ERootType &inRootType) const

12.257.1 Detailed Description

QuerySeries.

contains: class to construct a series-based query for c-find and c-move

12.257.2 Member Function Documentation

12.257.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QuerySeries::GetHierachicalSearchTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

12.257.2.2 GetName()

```
const char * gdcm::QuerySeries::GetName () const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.257.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QuerySeries::GetOptionalTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.257.2.4 GetQueryLevel()

```
DataElement gdcm::QuerySeries::GetQueryLevel () const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.257.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QuerySeries::GetRequiredTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.257.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QuerySeries::GetUniqueTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

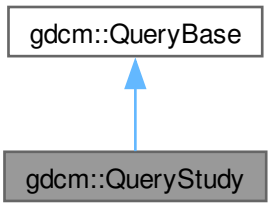
- gdcmQuerySeries.h

12.258 gdcm::QueryStudy Class Reference

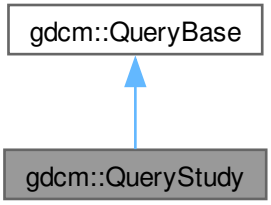
QueryStudy.h.

```
#include <gdcmQueryStudy.h>
```

Inheritance diagram for gdcm::QueryStudy:



Collaboration diagram for gdcm::QueryStudy:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const override

  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const override
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const override
- DataElement GetQueryLevel () const override
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const override
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const override

Public Member Functions inherited from gdcm::QueryBase

- virtual ~QueryBase ()=default
- std::vector< Tag > GetAllRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetAllTags (const ERootType &inRootType) const

12.258.1 Detailed Description

QueryStudy.h.

contains: class to construct a study-based query for C-FIND and C-MOVE

12.258.2 Member Function Documentation

12.258.2.1 GetHierachicalSearchTags()

```
std::vector< Tag > gdcm::QueryStudy::GetHierachicalSearchTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

12.258.2.2 GetName()

```
const char * gdcm::QueryStudy::GetName () const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.258.2.3 GetOptionalTags()

```
std::vector< Tag > gdcm::QueryStudy::GetOptionalTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.258.2.4 GetQueryLevel()

```
DataElement gdcm::QueryStudy::GetQueryLevel () const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.258.2.5 GetRequiredTags()

```
std::vector< Tag > gdcm::QueryStudy::GetRequiredTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

12.258.2.6 GetUniqueTags()

```
std::vector< Tag > gdcm::QueryStudy::GetUniqueTags (
            const ERootType & inRootType) const  [override], [virtual]
```

Implements gdcm::QueryBase.

The documentation for this class was generated from the following file:

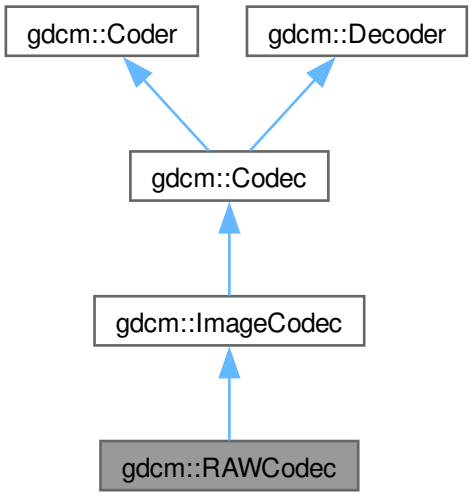
- gdcmQueryStudy.h

12.259 gdcm::RAWCodec Class Reference

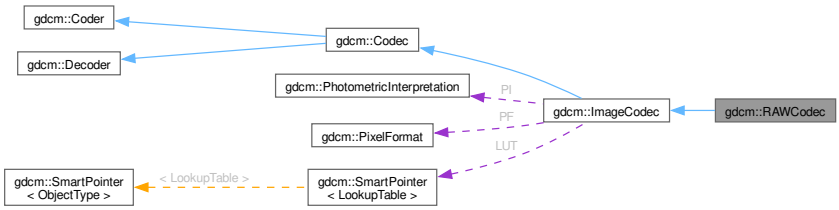
RAWCodec class.

```
#include <gdcmRAWCodec.h>
```

Inheritance diagram for gdcm::RAWCodec:



Collaboration diagram for gdcm::RAWCodec:



Public Member Functions

- RAWCodec ()
- ~RAWCodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- bool DecodeBytes (const char *inBytes, size_t inBufferLength, char *outBytes, size_t inOutBufferLength)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])
- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)

- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool DecodeByStreams (std::istream &is, std::ostream &os) override

Protected Member Functions inherited from gdcm::ImageCodec

- virtual bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- virtual bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsFrameEncoder ()
- virtual bool IsRowEncoder ()
- virtual bool IsValid (PhotometricInterpretation const &pi)
- virtual bool StartEncode (std::ostream &os)
- virtual bool StopEncode (std::ostream &os)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.259.1 Detailed Description

RAWCodec class.

12.259.2 Constructor & Destructor Documentation

12.259.2.1 RAWCodec()

```
gdcm::RAWCodec::RAWCodec ()
```

12.259.2.2 ~RAWCodec()

```
gdcm::RAWCodec::~RAWCodec ()  [override]
```

12.259.3 Member Function Documentation

12.259.3.1 CanCode()

```
bool gdcm::RAWCodec::CanCode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

12.259.3.2 CanDecode()

```
bool gdcm::RAWCodec::CanDecode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.259.3.3 Clone()

```
ImageCodec * gdcm::RAWCodec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.259.3.4 Code()

```
bool gdcm::RAWCodec::Code (
            DataElement const & in_,
            DataElement & out_)  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

12.259.3.5 Decode()

```
bool gdcm::RAWCodec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

12.259.3.6 DecodeByStreams()

```
bool gdcm::RAWCodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.259.3.7 DecodeBytes()

```
bool gdcm::RAWCodec::DecodeBytes (
            const char * inBytes,
            size_t inBufferLength,
            char * outBytes,
            size_t inOutBufferLength)
```

Used by the ImageStreamReader– converts a read in buffer into one with the proper encodings.

12.259.3.8 GetHeaderInfo()

```
bool gdcm::RAWCodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

The documentation for this class was generated from the following file:

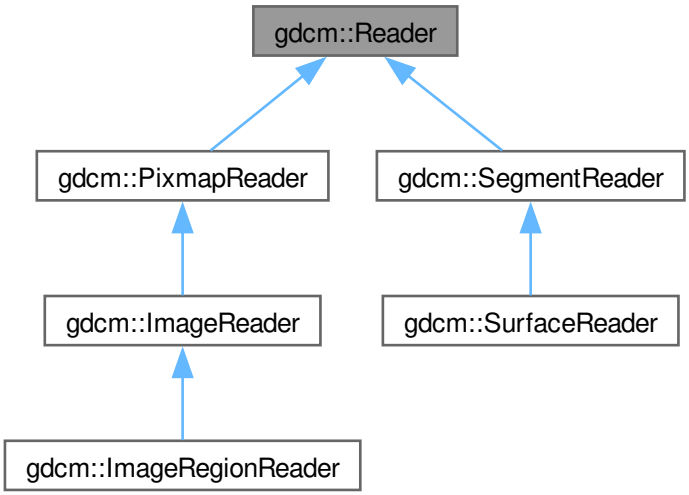
- gdcmRAWCodec.h

12.260 gdcm::Reader Class Reference

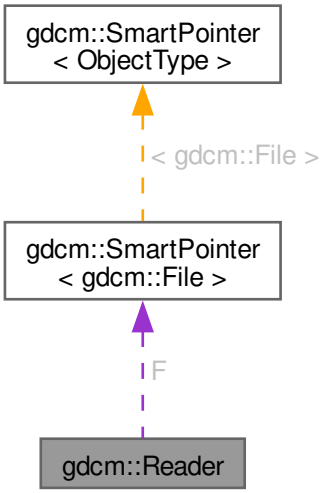
Reader ala DOM (Document Object Model).

```
#include <gdcmReader.h>
```

Inheritance diagram for gdcm::Reader:



Collaboration diagram for gdcm::Reader:



Public Member Functions

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- File & GetFile ()

  Set/Get File.
- const File & GetFile () const

  Set/Get File.
- size_t GetStreamCurrentPosition () const
- virtual bool Read ()

  Main function to read a file.
- bool ReadSelectedPrivateTags (std::set< PrivateTag > const &ptags, bool readvalues=true)

  Will only read the specified selected private tags.
- bool ReadSelectedTags (std::set< Tag > const &tags, bool readvalues=true)

  Will only read the specified selected tags.
- bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())
- void SetFile (File &file)

  Set/Get File.
- void SetFileName (const char *filename_native)
- void SetStream (std::istream &input_stream)

  Set the open-ed stream directly.

Protected Member Functions

- std::istream * GetStreamPtr () const
- bool ReadDataSet ()
- bool ReadMetaInformation ()
- bool ReadPreamble ()

Protected Attributes

- SmartPointer< File > F

Friends

- class StreamImageReader

12.260.1 Detailed Description

Reader ala DOM (Document Object Model).

This class is a non-validating reader, it will only performs well- formedness check only, and to some extent catch known error (non well-formed document).

Detailed description here

A DataSet DOES NOT contains group 0x0002 (see FileMetaInformation)

This is really a DataSet reader. This will not make sure the dataset conform to any IOD at all. This is a completely different step. The reasoning was that user could control the IOD there lib would handle and thus we would not be able to read a DataSet if the IOD was not found Instead we separate the reading from the validation.

Note

From GDCM1.x. Users will realize that one feature is missing from this DOM implementation. In GDCM 1.x user used to be able to control the size of the Value to be read. By default it was 0xfff. The main author of GDCM2 thought this was too dangerous and harmful and therefore this feature did not make it into GDCM2

Warning

GDCM will not produce warning for unorder (non-alphabetical order).

See also

Writer FileMetaInformation DataSet File

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateFakeRTDOSE.cxx, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixOrientation.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadUTF8QtDir.cxx, ReformatFile.cs, SimplePrint.cs, SimplePrintPatientName.cs, TestReader.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.260.2 Constructor & Destructor Documentation

12.260.2.1 Reader()

```
gdcm::Reader::Reader ()
```

12.260.2.2 ~Reader()

```
virtual gdcm::Reader::~Reader ()  [virtual]
```

12.260.3 Member Function Documentation

12.260.3.1 CanRead()

```
bool gdcm::Reader::CanRead () const
```

Test whether this is a DICOM file

Warning

need to call either SetFileName or SetStream first

Examples

ReadUTF8QtDir.cxx.

12.260.3.2 GetFile() [1/2]

```
File & gdcm::Reader::GetFile ()  [inline]
```

Set/Get File.

References F.

12.260.3.3 GetFile() [2/2]

```
const File & gdcm::Reader::GetFile () const  [inline]
```

Set/Get File.

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, Cleaner.cs, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, DecompressImage.cs, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReformatFile.cs, SimplePrint.cs, SimplePrintPatientName.cs, StandardizeFiles.cs, TestReader.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

References F.

12.260.3.4 GetStreamCurrentPosition()

```
size_t gdcm::Reader::GetStreamCurrentPosition () const
```

For wrapped language. return type is compatible with System::FileSize return type Use native std::streampos / std↩ ::streamoff directly from the stream from C++

Examples

ExtractImageRegion.cs.

12.260.3.5 GetStreamPtr()

```
std::istream * gdcm::Reader::GetStreamPtr () const  [inline], [protected]
```

12.260.3.6 Read()

```
virtual bool gdcm::Reader::Read ()  [virtual]
```

Main function to read a file.

Reimplemented in gdcm::ImageReader, gdcm::ImageRegionReader, gdcm::PixmapReader, gdcm::SegmentReader, and gdcm::SurfaceReader.

Examples

BasicAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, Cleaner.cs, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateFakeRTDOSE.cxx, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixOrientation.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReformatFile.cs, SimplePrint.cs, SimplePrintPatientName.cs, TestReader.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.260.3.7 ReadDataSet()

```
bool gdcm::Reader::ReadDataSet ()  [protected]
```

12.260.3.8 ReadMetaInformation()

```
bool gdcm::Reader::ReadMetaInformation ()  [protected]
```

12.260.3.9 ReadPreamble()

```
bool gdcm::Reader::ReadPreamble () [protected]
```

12.260.3.10 ReadSelectedPrivateTags()

```
bool gdcm::Reader::ReadSelectedPrivateTags (
            std::set< PrivateTag > const & ptags,
            bool readvalues = true)
```

Will only read the specified selected private tags.

12.260.3.11 ReadSelectedTags()

```
bool gdcm::Reader::ReadSelectedTags (
            std::set< Tag > const & tags,
            bool readvalues = true)
```

Will only read the specified selected tags.

12.260.3.12 ReadUpToTag()

```
bool gdcm::Reader::ReadUpToTag (
            const Tag & tag,
            std::set< Tag > const & skiptags = std::set< Tag >())
```

Will read only up to Tag

Parameters

tag	and skipping any tag specified in
skiptags	

Examples

DumpVisusChange.cxx.

12.260.3.13 SetFile()

```
void gdcm::Reader::SetFile (
            File & file) [inline]
```

Set/Get File.

References F.

12.260.3.14 SetFileName()

```
void gdcm::Reader::SetFileName (
            const char * filename_native)
```

Set the filename to open. This will create a std::ifstream internally See SetStream if you are dealing with different std::istream object

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CheckBigEndianBug.cxx, Cleaner.cs, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CompressImage.cxx, CompressLossyJPEG.cs, ConvertToQImage.cxx, CreateFakeRTDOSE.cxx, DecompressImage.cs, DeriveSeries.cxx, DiffFile.cxx, DumpADAC.cxx, DumpCSA.cs, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetArray.cs, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, PrintLUT.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadMultiTimesException.cxx, ReadUTF8QtDir.cxx, ReformatFile.cs, RescaleImage.cs, SimplePrint.cs, SimplePrintPatientName.cs, StandardizeFiles.cs, TemplateEmptyImage.cxx, TestReader.cxx, csa2img.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, rle2img.cxx, and threadgdcm.cxx.

12.260.3.15 SetStream()

```
void gdcm::Reader::SetStream (
            std::istream & input_stream)  [inline]
```

Set the open-ed stream directly.

Examples

ReadUTF8QtDir.cxx.

12.260.4 Friends And Related Symbol Documentation

12.260.4.1 StreamImageReader

```
friend class StreamImageReader  [friend]
```

References StreamImageReader.

Referenced by StreamImageReader.

12.260.5 Member Data Documentation

12.260.5.1 F

```
SmartPointer<File> gdcm::Reader::F  [protected]
```

Referenced by GetFile(), GetFile(), and SetFile().

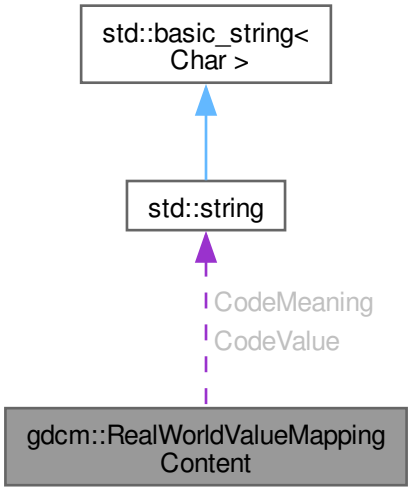
The documentation for this class was generated from the following file:

- gdcmReader.h

12.261 gdcm::RealWorldValueMappingContent Struct Reference

```
#include <gdcmImageHelper.h>
```

Collaboration diagram for gdcm::RealWorldValueMappingContent:



Public Attributes

- std::string CodeMeaning
- std::string CodeValue
- double RealWorldValueIntercept
- double RealWorldValueSlope

12.261.1 Member Data Documentation

12.261.1.1 CodeMeaning

```
std::string gdcm::RealWorldValueMappingContent::CodeMeaning
```

12.261.1.2 CodeValue

```
std::string gdcm::RealWorldValueMappingContent::CodeValue
```

12.261.1.3 RealWorldValueIntercept

```
double gdcm::RealWorldValueMappingContent::RealWorldValueIntercept
```

12.261.1.4 RealWorldValueSlope

```
double gdcm::RealWorldValueMappingContent::RealWorldValueSlope
```

The documentation for this struct was generated from the following file:

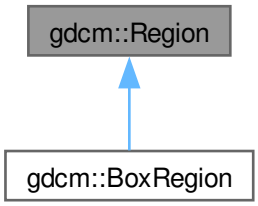
- gdcmImageHelper.h

12.262 gdcm::Region Class Reference

Class for manipulation region.

```
#include <gdcmRegion.h>
```

Inheritance diagram for gdcm::Region:



Public Member Functions

- Region ()
- virtual ∼Region ()
- virtual size_t Area () const =0

  compute the area
- virtual Region ∗ Clone () const =0
- virtual BoxRegion ComputeBoundingBox ()=0

  Return the Axis-Aligned minimum bounding box for all regions.
- virtual bool Empty () const =0

  return whether this domain is empty:
- virtual bool IsValid () const =0

  return whether this is valid domain
- virtual void Print (std::ostream &os=std::cout) const

  Print.

12.262.1 Detailed Description

Class for manipulation region.

12.262.2 Constructor & Destructor Documentation

12.262.2.1 Region()

```
gdcm::Region::Region ()
```

Referenced by gdcm::BoxRegion::Clone(), and Clone().

12.262.2.2 ∼Region()

```
virtual gdcm::Region::~Region ()  [virtual]
```

12.262.3 Member Function Documentation

12.262.3.1 Area()

```
virtual size_t gdcm::Region::Area () const  [pure virtual]
```

compute the area

Implemented in gdcm::BoxRegion.

12.262.3.2 Clone()

```
virtual Region * gdcm::Region::Clone () const  [pure virtual]
```

Implemented in gdcm::BoxRegion.

References Region().

12.262.3.3 ComputeBoundingBox()

```
virtual BoxRegion gdcm::Region::ComputeBoundingBox ()  [pure virtual]
```

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in gdcm::BoxRegion.

12.262.3.4 Empty()

```
virtual bool gdcm::Region::Empty () const  [pure virtual]
```

return whether this domain is empty:

Implemented in gdcm::BoxRegion.

12.262.3.5 IsValid()

```
virtual bool gdcm::Region::IsValid () const  [pure virtual]
```

return whether this is valid domain

Implemented in gdcm::BoxRegion.

12.262.3.6 Print()

```
virtual void gdcm::Region::Print (
            std::ostream & os = std::cout) const  [virtual]
```

Print.

Reimplemented in gdcm::BoxRegion.

Referenced by gdcm::operator<<().

The documentation for this class was generated from the following file:

- gdcmRegion.h

12.263 gdcm::Rescaler Class Reference

Rescale class.

```
#include <gdcmRescaler.h>
```

Public Member Functions

- Rescaler ()
- ~Rescaler ()=default
- PixelFormat::ScalarType ComputeInterceptSlopePixelType ()
- PixelFormat ComputePixelTypeFromMinMax ()
- double GetIntercept () const
- double GetSlope () const
- bool InverseRescale (char *out, const char *in, size_t n)

  Inverse transform.
- bool Rescale (char *out, const char *in, size_t n)

  Direct transform.
- void SetIntercept (double i)

  Set Intercept: used for both direct&inverse transformation.
- void SetMinMaxForPixelType (double min, double max)
- void SetPixelFormat (PixelFormat const &pf)

  Set Pixel Format of input data.
- void SetSlope (double s)

  Set Slope: user for both direct&inverse transformation.
- void SetTargetPixelType (PixelFormat const &targetst)
- void SetUseTargetPixelType (bool b)

  Override default behavior of Rescale.

Protected Member Functions

- template<typename TIn>
  void InverseRescaleFunctionIntoBestFit (char *out, const TIn *in, size_t n)
- template<typename TIn>
  void RescaleFunctionIntoBestFit (char *out, const TIn *in, size_t n)

12.263.1 Detailed Description

Rescale class.

This class is meant to apply the linear transform of Stored Pixel Value to Real World Value. This is mostly found in CT or PET dataset, where the value are stored using one type, but need to be converted to another scale using a linear transform. There are basically two cases: In CT: the linear transform is generally integer based. E.g. the Stored Pixel Type is unsigned short 12bits, but to get Hounsfield unit, one need to apply the linear transform:

$$RWV = 1. * SV - 1024$$

So the best scalar to store the Real World Value will be 16 bits signed type.

In PET: the linear transform is generally floating point based. Since the dynamic range can be quite high, the Rescale Slope / Rescale Intercept can be changing throughout the Series. So it is important to read all linear transform and deduce the best Pixel Type only at the end (when all the images to be read have been parsed).

Warning

Internally any time a floating point value is found either in the Rescale Slope or the Rescale Intercept it is assumed that the best matching output pixel type is FLOAT64 (in previous implementation it was FLOAT32). Because VR:DS is closer to a 64bits floating point type FLOAT64 is thus a best matching pixel type for the floating point transformation.

Example: Let say input is FLOAT64, and we want UINT16 as output, we would do:

```
Rescaler ir;
ir.SetIntercept( 0 );
ir.SetSlope( 5.6789 );
ir.SetPixelFormat( FLOAT64 );
ir.SetMinMaxForPixelType( ((PixelFormat)UINT16).GetMin(), ((PixelFormat)UINT16).GetMax() );
ir.InverseRescale(output,input,numberofbytes );
```

Note

handle floating point transformation back and forth to integer properly (no loss)

See also

Unpacker12Bits

Examples

RescaleImage.cs.

12.263.2 Constructor & Destructor Documentation

12.263.2.1 Rescaler()

```
gdcm::Rescaler::Rescaler () [inline]
```

12.263.2.2 ~Rescaler()

```
gdcm::Rescaler::~Rescaler () [default]
```

12.263.3 Member Function Documentation

12.263.3.1 ComputeInterceptSlopePixelType()

```
PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ()
```

Compute the Pixel Format of the output data Used for direct transformation

Examples

RescaleImage.cs.

12.263.3.2 ComputePixelTypeFromMinMax()

```
PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ()
```

Compute the Pixel Format of the output data Used for inverse transformation

12.263.3.3 GetIntercept()

```
double gdcm::Rescaler::GetIntercept () const  [inline]
```

12.263.3.4 GetSlope()

```
double gdcm::Rescaler::GetSlope () const  [inline]
```

12.263.3.5 InverseRescale()

```
bool gdcm::Rescaler::InverseRescale (
            char * out,
            const char * in,
            size_t n)
```

Inverse transform.

12.263.3.6 InverseRescaleFunctionIntoBestFit()

```
template<typename TIn>
void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (
            char * out,
            const TIn * in,
            size_t n)  [protected]
```

12.263.3.7 Rescale()

```
bool gdcm::Rescaler::Rescale (
            char * out,
            const char * in,
            size_t n)
```

Direct transform.

Examples

RescaleImage.cs.

12.263.3.8 RescaleFunctionIntoBestFit()

```
template<typename TIn>
void gdcm::Rescaler::RescaleFunctionIntoBestFit (
            char * out,
            const TIn * in,
            size_t n)  [protected]
```

12.263.3.9 SetIntercept()

```
void gdcm::Rescaler::SetIntercept (
            double i)  [inline]
```

Set Intercept: used for both direct&inverse transformation.

Examples

RescaleImage.cs.

12.263.3.10 SetMinMaxForPixelType()

```
void gdcm::Rescaler::SetMinMaxForPixelType (
            double min,
            double max)
```

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

12.263.3.11 SetPixelFormat()

```
void gdcm::Rescaler::SetPixelFormat (
            PixelFormat const & pf)  [inline]
```

Set Pixel Format of input data.

Examples

RescaleImage.cs.

12.263.3.12 SetSlope()

```
void gdcm::Rescaler::SetSlope (
            double s)  [inline]
```

Set Slope: user for both direct&inverse transformation.

Examples

RescaleImage.cs.

12.263.3.13 SetTargetPixelType()

```
void gdcm::Rescaler::SetTargetPixelType (
            PixelFormat const & targetst)
```

By default (when UseTargetPixelType is false), a best matching Target Pixel Type is computed. However user can override this auto selection by switching UseTargetPixelType:true and also specifying the specific Target Pixel Type

12.263.3.14 SetUseTargetPixelType()

```
void gdcm::Rescaler::SetUseTargetPixelType (
            bool b)
```

Override default behavior of Rescale.

The documentation for this class was generated from the following file:

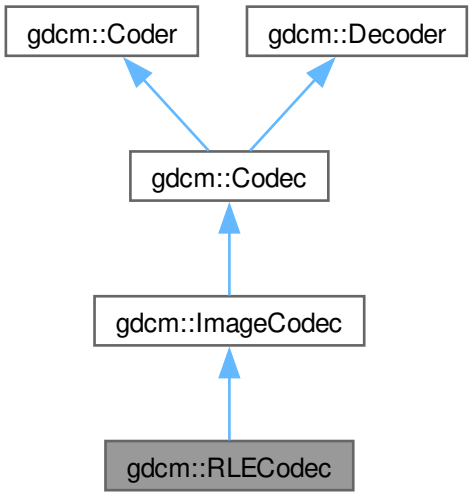
- gdcmRescaler.h

12.264 gdcm::RLECodec Class Reference

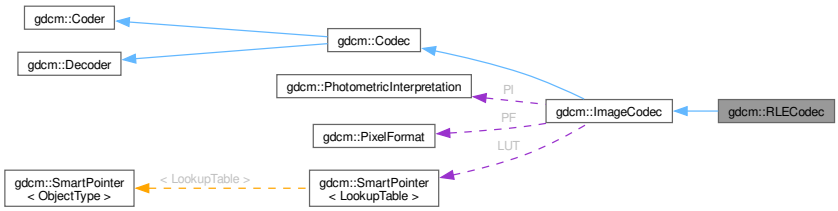
Class to do RLE.

```
#include <gdcmRLECodec.h>
```

Inheritance diagram for gdcm::RLECodec:



Collaboration diagram for gdcm::RLECodec:



Public Member Functions

- RLECodec ()
- ~RLECodec () override
- bool CanCode (TransferSyntax const &ts) const override

  Return whether this coder support this transfer syntax (can code it).
- bool CanDecode (TransferSyntax const &ts) const override

  Return whether this decoder support this transfer syntax (can decode it).
- ImageCodec * Clone () const override
- bool Code (DataElement const &in, DataElement &out) override

  Code.
- bool Decode (DataElement const &is, DataElement &os) override

  Decode.
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts) override
- void SetBufferLength (unsigned long l)
- void SetLength (unsigned long l)

Public Member Functions inherited from gdcm::ImageCodec

- ImageCodec ()
- ~ImageCodec () override
- bool CleanupUnusedBits (char *data, size_t datalen)
- const unsigned int * GetDimensions () const
- bool GetLossyFlag () const
- const LookupTable & GetLUT () const
- bool GetNeedByteSwap () const
- unsigned int GetNumberOfDimensions () const
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- PixelFormat & GetPixelFormat ()
- const PixelFormat & GetPixelFormat () const
- unsigned int GetPlanarConfiguration () const
- bool IsLossy () const
- void SetDimensions (const std::vector< unsigned int > &d)
- void SetDimensions (const unsigned int d[3])

- void SetLossyFlag (bool l)
- void SetLUT (LookupTable const &lut)
- void SetNeedByteSwap (bool b)
- void SetNeedOverlayCleanup (bool b)
- void SetNumberOfDimensions (unsigned int dim)
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)
- virtual void SetPixelFormat (PixelFormat const &pf)
- void SetPlanarConfiguration (unsigned int pc)

Public Member Functions inherited from gdcm::Coder

- virtual ~Coder ()=default

Public Member Functions inherited from gdcm::Decoder

- virtual ~Decoder ()=default

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen) override
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen) override
- bool DecodeByStreams (std::istream &is, std::ostream &os) override
- bool DecodeExtent (char *buffer, unsigned int XMin, unsigned int XMax, unsigned int YMin, unsigned int YMax, unsigned int ZMin, unsigned int ZMax, std::istream &is)
- bool IsFrameEncoder () override
- bool IsRowEncoder () override
- bool StartEncode (std::ostream &) override
- bool StopEncode (std::ostream &) override

Protected Member Functions inherited from gdcm::ImageCodec

- bool DoByteSwap (std::istream &is_, std::ostream &os)
- bool DoInvertMonochrome (std::istream &is_, std::ostream &os)
- bool DoOverlayCleanup (std::istream &is_, std::ostream &os)
- bool DoPaddedCompositePixelCode (std::istream &is_, std::ostream &os)
- bool DoPlanarConfiguration (std::istream &is_, std::ostream &os)
- bool DoSimpleCopy (std::istream &is_, std::ostream &os)
- bool DoYBR (std::istream &is_, std::ostream &os)
- bool DoYBRFull422 (std::istream &is_, std::ostream &os)
- virtual bool IsValid (PhotometricInterpretation const &pi)

Protected Member Functions inherited from gdcm::Coder

- virtual bool InternalCode (const char *bv, unsigned long len, std::ostream &os)

Friends

- class ImageRegionReader

Additional Inherited Members

Protected Types inherited from gdcm::ImageCodec

- typedef SmartPointer< LookupTable > LUTPtr

Protected Attributes inherited from gdcm::ImageCodec

- unsigned int Dimensions [3]
- bool LossyFlag
- LUTPtr LUT
- bool NeedByteSwap
- bool NeedOverlayCleanup
- unsigned int NumberOfDimensions
- PixelFormat PF
- PhotometricInterpretation PI
- unsigned int PlanarConfiguration
- bool RequestPaddedCompositePixelCode
- bool RequestPlanarConfiguration

12.264.1 Detailed Description

Class to do RLE.

Note

ANSI X3.9 A.4.2 RLE Compression Annex G defines a RLE Compression Transfer Syntax. This transfer Syntax is identified by the UID value "1.2.840.10008.1.2.5". If the object allows multi-frame images in the pixel data field, then each frame shall be encoded separately. Each frame shall be encoded in one and only one Fragment (see PS 3.5.8.2).

12.264.2 Constructor & Destructor Documentation

12.264.2.1 RLECodec()

```
gdcm::RLECodec::RLECodec ()
```

12.264.2.2 ~RLECodec()

```
gdcm::RLECodec::~RLECodec () [override]
```

12.264.3 Member Function Documentation

12.264.3.1 AppendFrameEncode()

```
bool gdcm::RLECodec::AppendFrameEncode (
            std::ostream & out,
            const char * data,
            size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.264.3.2 AppendRowEncode()

```
bool gdcm::RLECodec::AppendRowEncode (
            std::ostream & out,
            const char * data,
            size_t datalen)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.264.3.3 CanCode()

```
bool gdcm::RLECodec::CanCode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this coder support this transfer syntax (can code it).

Reimplemented from gdcm::ImageCodec.

12.264.3.4 CanDecode()

```
bool gdcm::RLECodec::CanDecode (
            TransferSyntax const & ) const  [override], [virtual]
```

Return whether this decoder support this transfer syntax (can decode it).

Reimplemented from gdcm::ImageCodec.

12.264.3.5 Clone()

```
ImageCodec * gdcm::RLECodec::Clone () const  [override], [virtual]
```

Implements gdcm::ImageCodec.

References gdcm::ImageCodec::ImageCodec().

12.264.3.6 Code()

```
bool gdcm::RLECodec::Code (
            DataElement const & in_,
            DataElement & out_)  [override], [virtual]
```

Code.

Reimplemented from gdcm::Coder.

12.264.3.7 Decode()

```
bool gdcm::RLECodec::Decode (
            DataElement const & ,
            DataElement & )  [override], [virtual]
```

Decode.

Reimplemented from gdcm::ImageCodec.

12.264.3.8 DecodeByStreams()

```
bool gdcm::RLECodec::DecodeByStreams (
            std::istream & is,
            std::ostream & os)  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.264.3.9 DecodeExtent()

```
bool gdcm::RLECodec::DecodeExtent (
            char * buffer,
            unsigned int XMin,
            unsigned int XMax,
            unsigned int YMin,
            unsigned int YMax,
            unsigned int ZMin,
            unsigned int ZMax,
            std::istream & is)  [protected]
```

12.264.3.10 GetBufferLength()

```
unsigned long gdcm::RLECodec::GetBufferLength () const  [inline]
```

12.264.3.11 GetHeaderInfo()

```
bool gdcm::RLECodec::GetHeaderInfo (
            std::istream & is,
            TransferSyntax & ts)  [override], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.264.3.12 IsFrameEncoder()

```
bool gdcm::RLECodec::IsFrameEncoder ()  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.264.3.13 IsRowEncoder()

```
bool gdcm::RLECodec::IsRowEncoder ()  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.264.3.14 SetBufferLength()

```
void gdcm::RLECodec::SetBufferLength (
            unsigned long l)  [inline]
```

12.264.3.15 SetLength()

```
void gdcm::RLECodec::SetLength (
            unsigned long l)  [inline]
```

12.264.3.16 StartEncode()

```
bool gdcm::RLECodec::StartEncode (
            std::ostream & )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.264.3.17 StopEncode()

```
bool gdcm::RLECodec::StopEncode (
            std::ostream & )  [override], [protected], [virtual]
```

Reimplemented from gdcm::ImageCodec.

12.264.4 Friends And Related Symbol Documentation

12.264.4.1 ImageRegionReader

```
friend class ImageRegionReader  [friend]
```

References ImageRegionReader.

Referenced by ImageRegionReader.

The documentation for this class was generated from the following file:

- gdcmRLECodec.h

12.265 gdcm::network::RoleSelectionSub Class Reference

RoleSelectionSub.

```
#include <gdcmRoleSelectionSub.h>
```

Public Member Functions

- RoleSelectionSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetTuple (const char *uid, uint8_t scurole, uint8_t scprole)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.265.1 Detailed Description

RoleSelectionSub.

PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

12.265.2 Constructor & Destructor Documentation

12.265.2.1 RoleSelectionSub()

```
gdcm::network::RoleSelectionSub::RoleSelectionSub ()
```

12.265.3 Member Function Documentation

12.265.3.1 Print()

```
void gdcm::network::RoleSelectionSub::Print (
            std::ostream & os) const
```

12.265.3.2 Read()

```
std::istream & gdcm::network::RoleSelectionSub::Read (
            std::istream & is)
```

12.265.3.3 SetTuple()

```
void gdcm::network::RoleSelectionSub::SetTuple (
            const char * uid,
            uint8_t scurole,
            uint8_t scprole)
```

12.265.3.4 Size()

```
size_t gdcm::network::RoleSelectionSub::Size () const
```

12.265.3.5 Write()

```
const std::ostream & gdcm::network::RoleSelectionSub::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

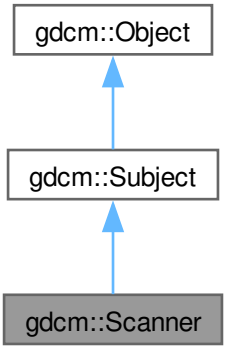
- gdcmRoleSelectionSub.h

12.266 gdcm::Scanner Class Reference

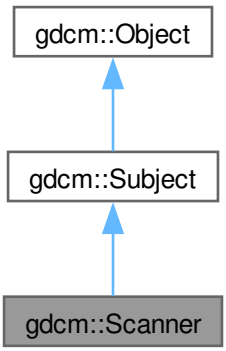
Scanner.

```
#include <gdcmScanner.h>
```

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



Classes

- struct ltstr

Public Types

- typedef MappingType::const_iterator ConstIterator
- typedef std::map< const char *, TagToValue, ltstr > MappingType
- typedef std::map< Tag, const char * > TagToValue
- typedef TagToValue::value_type TagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- Scanner ()
- ~Scanner () override
- void AddPrivateTag (PrivateTag const &t)
- void AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- void AddTag (Tag const &t)

  Add a tag that will need to be read. Those are root level tags.
- ConstIterator Begin () const
- void ClearSkipTags ()
- void ClearTags ()
- ConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const
- Directory::FilenamesType GetKeys () const
- TagToValue const & GetMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- TagToValue const & GetMappingFromTagToValue (Tag const &t, const char *value) const

  See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.
- MappingType const & GetMappings () const

  Mappings are the mapping from a particular tag to the map, mapping filename to value:
- Directory::FilenamesType GetOrderedValues (Tag const &t) const
- const char * GetValue (const char *filename, Tag const &t) const
- ValuesType const & GetValues () const

  Get all the values found (in lexicographic order).
- ValuesType GetValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const override

  Print result.
- void PrintTable (std::ostream &os) const
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Static Public Member Functions

- static SmartPointer< Scanner > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- void ProcessPublicTag (StringFilter &sf, const char *filename)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Friends

- std::ostream & operator<< (std::ostream &_os, const Scanner &s)

12.266.1 Detailed Description

Scanner.

This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

Examples

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SortImage.cxx, and VolumeSorter.cxx.

12.266.2 Member Typedef Documentation

12.266.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::Scanner::ConstIterator
```

12.266.2.2 MappingType

```
typedef std::map<const char *,TagToValue, ltstr> gdcm::Scanner::MappingType
```

12.266.2.3 TagToValue

```
typedef std::map<Tag, const char*> gdcm::Scanner::TagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char $*$ and not std::string since we are pointing to existing std::string (hold in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) $<=$ sizeof(pointer)

12.266.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType
```

12.266.2.5 ValuesType

```
typedef std::set< std::string > gdcm::Scanner::ValuesType
```

Examples

DiscriminateVolume.cxx, SortImage.cxx, and VolumeSorter.cxx.

12.266.3 Constructor & Destructor Documentation

12.266.3.1 Scanner()

```
gdcm::Scanner::Scanner () [inline]
```

Referenced by New(), and operator<<.

12.266.3.2 ~Scanner()

```
gdcm::Scanner::~Scanner () [override]
```

12.266.4 Member Function Documentation

12.266.4.1 AddPrivateTag()

```
void gdcm::Scanner::AddPrivateTag (
            PrivateTag const & t)
```

12.266.4.2 AddSkipTag()

```
void gdcm::Scanner::AddSkipTag (
            Tag const & t)
```

Add a tag that will need to be skipped. Those are root level skip tags.

12.266.4.3 AddTag()

```
void gdcm::Scanner::AddTag (
            Tag const & t)
```

Add a tag that will need to be read. Those are root level tags.

Examples

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SortImage.cxx, and VolumeSorter.cxx.

12.266.4.4 Begin()

```
ConstIterator gdcm::Scanner::Begin () const  [inline]
```

12.266.4.5 ClearSkipTags()

```
void gdcm::Scanner::ClearSkipTags ()
```

12.266.4.6 ClearTags()

```
void gdcm::Scanner::ClearTags ()
```

12.266.4.7 End()

```
ConstIterator gdcm::Scanner::End () const  [inline]
```

12.266.4.8 GetAllFilenamesFromTagToValue()

```
Directory::FilenamesType gdcm::Scanner::GetAllFilenamesFromTagToValue (
            Tag const & t,
            const char * valueref) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

12.266.4.9 GetFilenameFromTagToValue()

```
const char * gdcm::Scanner::GetFilenameFromTagToValue (
            Tag const & t,
            const char * valueref) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

12.266.4.10 GetFilenames()

```
Directory::FilenamesType const & gdcm::Scanner::GetFilenames () const  [inline]
```

12.266.4.11 GetKeys()

```
Directory::FilenamesType gdcm::Scanner::GetKeys () const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

Examples

VolumeSorter.cxx.

12.266.4.12 GetMapping()

```
TagToValue const & gdcm::Scanner::GetMapping (
            const char * filename) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples

DumpToSQLITE3.cxx.

12.266.4.13 GetMappingFromTagToValue()

```
TagToValue const & gdcm::Scanner::GetMappingFromTagToValue (
            Tag const & t,
            const char * value) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

12.266.4.14 GetMappings()

```
MappingType const & gdcm::Scanner::GetMappings () const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

12.266.4.15 GetOrderedValues()

```
Directory::FilenamesType gdcm::Scanner::GetOrderedValues (
            Tag const & t) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

12.266.4.16 GetValue()

```
const char * gdcm::Scanner::GetValue (
            const char * filename,
            Tag const & t) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

12.266.4.17 GetValues() [1/2]

```
ValuesType const & gdcm::Scanner::GetValues () const  [inline]
```

Get all the values found (in lexicographic order).

Examples

SortImage.cxx, and VolumeSorter.cxx.

12.266.4.18 GetValues() [2/2]

```
ValuesType gdcm::Scanner::GetValues (
            Tag const & t) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

12.266.4.19 IsKey()

```
bool gdcm::Scanner::IsKey (
            const char * filename) const
```

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

Examples

DumpToSQLITE3.cxx.

12.266.4.20 New()

```
SmartPointer< Scanner > gdcm::Scanner::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

References Scanner().

12.266.4.21 Print()

```
void gdcm::Scanner::Print (
            std::ostream & os) const [override], [virtual]
```

Print result.

Reimplemented from gdcm::Object.

Referenced by operator<<.

12.266.4.22 PrintTable()

```
void gdcm::Scanner::PrintTable (
            std::ostream & os) const
```

12.266.4.23 ProcessPublicTag()

```
void gdcm::Scanner::ProcessPublicTag (
            StringFilter & sf,
            const char * filename) [protected]
```

12.266.4.24 Scan()

```
bool gdcm::Scanner::Scan (
            Directory::FilenamesType const & filenames)
```

Start the scan !

Examples

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SortImage.cxx, and VolumeSorter.cxx.

12.266.5 Friends And Related Symbol Documentation

12.266.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Scanner & s)  [friend]
```

References Scanner(), and Print().

The documentation for this class was generated from the following file:

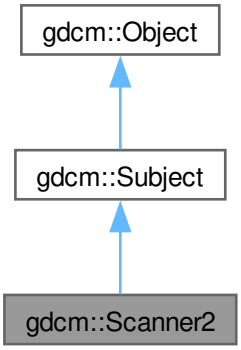
- gdcmScanner.h

12.267 gdcm::Scanner2 Class Reference

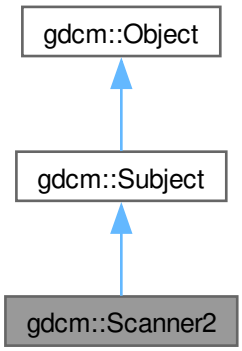
Scanner2.

```
#include <gdcmScanner2.h>
```

Inheritance diagram for gdcm::Scanner2:



Collaboration diagram for gdcm::Scanner2:



Classes

- struct ltstr

Public Types

- typedef PrivateMappingType::const_iterator PrivateConstIterator
- typedef std::map< const char *, PrivateTagToValue, ltstr > PrivateMappingType
- typedef std::map< PrivateTag, const char * > PrivateTagToValue
- typedef PrivateTagToValue::value_type PrivateTagToValueValueType
- typedef PublicMappingType::const_iterator PublicConstIterator
- typedef std::map< const char *, PublicTagToValue, ltstr > PublicMappingType
- typedef std::map< Tag, const char * > PublicTagToValue
- typedef PublicTagToValue::value_type PublicTagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- Scanner2 ()
- ~Scanner2 () override
- bool AddPrivateTag (PrivateTag const &pt)
- bool AddPublicTag (Tag const &t)

  Add a tag that will need to be read. Those are root level tags.
- bool AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- PublicConstIterator Begin () const
- void ClearPrivateTags ()
- void ClearPublicTags ()

- void ClearSkipTags ()
- PublicConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue (PrivateTag const &pt, const char *valueref) const
- Directory::FilenamesType GetAllFilenamesFromPublicTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromPrivateTagToValue (PrivateTag const &pt, const char *valueref) const
- const char * GetFilenameFromPublicTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const

  Return the list of filenames.
- Directory::FilenamesType GetKeys () const
- PrivateTagToValue const & GetMappingFromPrivateTagToValue (PrivateTag const &pt, const char *value) const
- PublicTagToValue const & GetMappingFromPublicTagToValue (Tag const &t, const char *value) const

  See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.
- PrivateTagToValue const & GetPrivateMapping (const char *filename) const
- PrivateMappingType const & GetPrivateMappings () const
- Directory::FilenamesType GetPrivateOrderedValues (PrivateTag const &pt) const
- const char * GetPrivateValue (const char *filename, PrivateTag const &t) const
- ValuesType GetPrivateValues (PrivateTag const &pt) const

  Get all the values found (in lexicographic order) associated with PrivateTag 'pt'.
- PublicTagToValue const & GetPublicMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- PublicMappingType const & GetPublicMappings () const

  Mappings are the mapping from a particular tag to the map, mapping filename to value:
- Directory::FilenamesType GetPublicOrderedValues (Tag const &t) const
- const char * GetPublicValue (const char *filename, Tag const &t) const
- ValuesType GetPublicValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- ValuesType const & GetValues () const

  Get all the values found (in lexicographic order).
- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const override

  Print result.
- void PrintTable (std::ostream &os, bool header=false) const

  Print result as CSV table.
- PrivateConstIterator PrivateBegin () const
- PrivateConstIterator PrivateEnd () const
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Static Public Member Functions

- static SmartPointer< Scanner2 > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- void ProcessPrivateTag (StringFilter &sf, const char *filename)
- void ProcessPublicTag (StringFilter &sf, const char *filename)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Friends

- std::ostream & operator<< (std::ostream &_os, const Scanner2 &s)

12.267.1 Detailed Description

Scanner2.

This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

12.267.2 Member Typedef Documentation

12.267.2.1 PrivateConstIterator

```
typedef PrivateMappingType::const_iterator gdcm::Scanner2::PrivateConstIterator
```

12.267.2.2 PrivateMappingType

```
typedef std::map<const char *,PrivateTagToValue, ltstr> gdcm::Scanner2::PrivateMappingType
```

12.267.2.3 PrivateTagToValue

```
typedef std::map<PrivateTag, const char*> gdcm::Scanner2::PrivateTagToValue
```

12.267.2.4 PrivateTagToValueValueType

```
typedef PrivateTagToValue::value_type gdcm::Scanner2::PrivateTagToValueValueType
```

12.267.2.5 PublicConstIterator

```
typedef PublicMappingType::const_iterator gdcm::Scanner2::PublicConstIterator
```

12.267.2.6 PublicMappingType

```
typedef std::map<const char *,PublicTagToValue, ltstr> gdcm::Scanner2::PublicMappingType
```

12.267.2.7 PublicTagToValue

```
typedef std::map<Tag, const char*> gdcm::Scanner2::PublicTagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char $*$ and not std::string since we are pointing to existing std::string (held in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

12.267.2.8 PublicTagToValueValueType

```
typedef PublicTagToValue::value_type gdcm::Scanner2::PublicTagToValueValueType
```

12.267.2.9 ValuesType

```
typedef std::set< std::string > gdcm::Scanner2::ValuesType
```

12.267.3 Constructor & Destructor Documentation

12.267.3.1 Scanner2()

```
gdcm::Scanner2::Scanner2 () [inline]
```

Referenced by New(), and operator<<.

12.267.3.2 ~Scanner2()

```
gdcm::Scanner2::~Scanner2 () [override]
```

12.267.4 Member Function Documentation

12.267.4.1 AddPrivateTag()

```
bool gdcm::Scanner2::AddPrivateTag (
            PrivateTag const & pt)
```

12.267.4.2 AddPublicTag()

```
bool gdcm::Scanner2::AddPublicTag (
            Tag const & t)
```

Add a tag that will need to be read. Those are root level tags.

12.267.4.3 AddSkipTag()

```
bool gdcm::Scanner2::AddSkipTag (
            Tag const & t)
```

Add a tag that will need to be skipped. Those are root level skip tags.

12.267.4.4 Begin()

```
PublicConstIterator gdcm::Scanner2::Begin () const [inline]
```

12.267.4.5 ClearPrivateTags()

```
void gdcm::Scanner2::ClearPrivateTags ()
```

12.267.4.6 ClearPublicTags()

```
void gdcm::Scanner2::ClearPublicTags ()
```

12.267.4.7 ClearSkipTags()

```
void gdcm::Scanner2::ClearSkipTags ()
```

12.267.4.8 End()

```
PublicConstIterator gdcm::Scanner2::End () const  [inline]
```

12.267.4.9 GetAllFilenamesFromPrivateTagToValue()

```
Directory::FilenamesType gdcm::Scanner2::GetAllFilenamesFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * valueref) const
```

12.267.4.10 GetAllFilenamesFromPublicTagToValue()

```
Directory::FilenamesType gdcm::Scanner2::GetAllFilenamesFromPublicTagToValue (
            Tag const & t,
            const char * valueref) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

12.267.4.11 GetFilenameFromPrivateTagToValue()

```
const char * gdcm::Scanner2::GetFilenameFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * valueref) const
```

12.267.4.12 GetFilenameFromPublicTagToValue()

```
const char * gdcm::Scanner2::GetFilenameFromPublicTagToValue (
            Tag const & t,
            const char * valueref) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

12.267.4.13 GetFilenames()

```
Directory::FilenamesType const & gdcm::Scanner2::GetFilenames () const  [inline]
```

Return the list of filenames.

12.267.4.14 GetKeys()

```
Directory::FilenamesType gdcm::Scanner2::GetKeys () const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

12.267.4.15 GetMappingFromPrivateTagToValue()

```
PrivateTagToValue const & gdcm::Scanner2::GetMappingFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * value) const
```

12.267.4.16 GetMappingFromPublicTagToValue()

```
PublicTagToValue const & gdcm::Scanner2::GetMappingFromPublicTagToValue (
            Tag const & t,
            const char * value) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

12.267.4.17 GetPrivateMapping()

```
PrivateTagToValue const & gdcm::Scanner2::GetPrivateMapping (
            const char * filename) const
```

12.267.4.18 GetPrivateMappings()

```
PrivateMappingType const & gdcm::Scanner2::GetPrivateMappings () const  [inline]
```

12.267.4.19 GetPrivateOrderedValues()

```
Directory::FilenamesType gdcm::Scanner2::GetPrivateOrderedValues (
            PrivateTag const & pt) const
```

12.267.4.20 GetPrivateValue()

```
const char * gdcm::Scanner2::GetPrivateValue (
            const char * filename,
            PrivateTag const & t) const
```

12.267.4.21 GetPrivateValues()

```
ValuesType gdcm::Scanner2::GetPrivateValues (
            PrivateTag const & pt) const
```

Get all the values found (in lexicographic order) associated with PrivateTag 'pt'.

12.267.4.22 GetPublicMapping()

```
PublicTagToValue const & gdcm::Scanner2::GetPublicMapping (
            const char * filename) const
```

Get the std::map mapping filenames to value for file 'filename'.

12.267.4.23 GetPublicMappings()

```
PublicMappingType const & gdcm::Scanner2::GetPublicMappings () const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

12.267.4.24 GetPublicOrderedValues()

```
Directory::FilenamesType gdcm::Scanner2::GetPublicOrderedValues (
            Tag const & t) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

12.267.4.25 GetPublicValue()

```
const char * gdcm::Scanner2::GetPublicValue (
            const char * filename,
            Tag const & t) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

12.267.4.26 GetPublicValues()

```
ValuesType gdcm::Scanner2::GetPublicValues (
            Tag const & t) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

12.267.4.27 GetValues()

```
ValuesType const & gdcm::Scanner2::GetValues () const  [inline]
```

Get all the values found (in lexicographic order).

12.267.4.28 IsKey()

```
bool gdcm::Scanner2::IsKey (
            const char * filename) const
```

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

12.267.4.29 New()

```
SmartPointer< Scanner2 > gdcm::Scanner2::New ()  [inline], [static]
```

for wrapped language: instantiate a reference counted object

References Scanner2().

12.267.4.30 Print()

```
void gdcm::Scanner2::Print (
            std::ostream & os) const  [override], [virtual]
```

Print result.

Reimplemented from gdcm::Object.

Referenced by operator<<.

12.267.4.31 PrintTable()

```
void gdcm::Scanner2::PrintTable (
            std::ostream & os,
            bool header = false) const
```

Print result as CSV table.

12.267.4.32 PrivateBegin()

```
PrivateConstIterator gdcm::Scanner2::PrivateBegin () const  [inline]
```

12.267.4.33 PrivateEnd()

```
PrivateConstIterator gdcm::Scanner2::PrivateEnd () const  [inline]
```

12.267.4.34 ProcessPrivateTag()

```
void gdcm::Scanner2::ProcessPrivateTag (
            StringFilter & sf,
            const char * filename)  [protected]
```

12.267.4.35 ProcessPublicTag()

```
void gdcm::Scanner2::ProcessPublicTag (
            StringFilter & sf,
            const char * filename)  [protected]
```

12.267.4.36 Scan()

```
bool gdcm::Scanner2::Scan (
            Directory::FilenamesType const & filenames)
```

Start the scan !

12.267.5 Friends And Related Symbol Documentation

12.267.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Scanner2 & s)  [friend]
```

References Scanner2(), and Print().

The documentation for this class was generated from the following file:

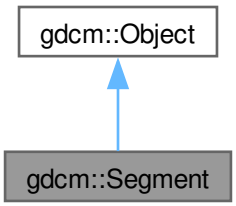
- gdcmScanner2.h

12.268 gdcm::Segment Class Reference

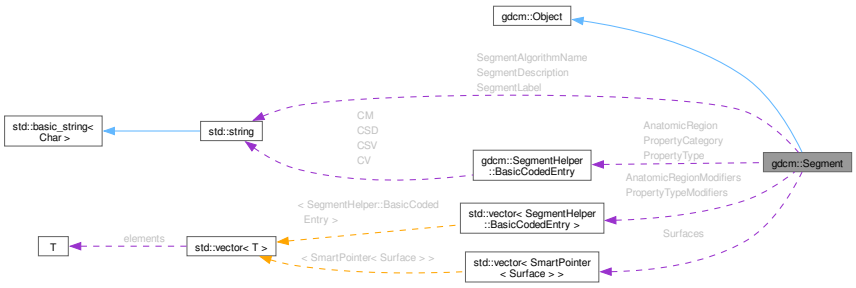
This class defines a segment.

```
#include <gdcmSegment.h>
```

Inheritance diagram for gdcm::Segment:



Collaboration diagram for gdcm::Segment:



Public Types

- enum ALGOType {
  AUTOMATIC = 0 ,
  SEMIAUTOMATIC ,
  MANUAL ,
  ALGOType_END }
- typedef std::vector< SegmentHelper::BasicCodedEntry > BasicCodedEntryVector
- typedef std::vector< SmartPointer< Surface > > SurfaceVector

Public Member Functions

- Segment ()
- ~Segment () override
- void AddSurface (SmartPointer< Surface > surface)
- SegmentHelper::BasicCodedEntry & GetAnatomicRegion ()
- SegmentHelper::BasicCodedEntry const & GetAnatomicRegion () const
- BasicCodedEntryVector & GetAnatomicRegionModifiers ()
- BasicCodedEntryVector const & GetAnatomicRegionModifiers () const
- SegmentHelper::BasicCodedEntry & GetPropertyCategory ()
- SegmentHelper::BasicCodedEntry const & GetPropertyCategory () const
- SegmentHelper::BasicCodedEntry & GetPropertyType ()
- SegmentHelper::BasicCodedEntry const & GetPropertyType () const
- BasicCodedEntryVector & GetPropertyTypeModifiers ()
- BasicCodedEntryVector const & GetPropertyTypeModifiers () const
- const char * GetSegmentAlgorithmName () const
- ALGOType GetSegmentAlgorithmType () const
- const char * GetSegmentDescription () const
- const char * GetSegmentLabel () const
- unsigned short GetSegmentNumber () const
- SmartPointer< Surface > GetSurface (const unsigned int idx=0) const
- unsigned long GetSurfaceCount ()
- SurfaceVector & GetSurfaces ()
- SurfaceVector const & GetSurfaces () const
- void SetAnatomicRegion (SegmentHelper::BasicCodedEntry const &BSE)
- void SetAnatomicRegionModifiers (BasicCodedEntryVector const &BSEV)
- void SetPropertyCategory (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyType (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyTypeModifiers (BasicCodedEntryVector const &BSEV)
- void SetSegmentAlgorithmName (const char *name)
- void SetSegmentAlgorithmType (ALGOType type)
- void SetSegmentAlgorithmType (const char *typeStr)
- void SetSegmentDescription (const char *description)
- void SetSegmentLabel (const char *label)
- void SetSegmentNumber (const unsigned short num)
- void SetSurfaceCount (const unsigned long nb)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static ALGOType GetALGOType (const char *type)
- static const char * GetALGOTypeString (ALGOType type)

Protected Attributes

- SegmentHelper::BasicCodedEntry AnatomicRegion
- BasicCodedEntryVector AnatomicRegionModifiers
- SegmentHelper::BasicCodedEntry PropertyCategory
- SegmentHelper::BasicCodedEntry PropertyType
- BasicCodedEntryVector PropertyTypeModifiers
- std::string SegmentAlgorithmName
- ALGOType SegmentAlgorithmType
- std::string SegmentDescription
- std::string SegmentLabel
- unsigned short SegmentNumber
- unsigned long SurfaceCount
- SurfaceVector Surfaces

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.268.1 Detailed Description

This class defines a segment.

It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

See also

PS 3.3 C.8.20.2 and C.8.23

12.268.2 Member Typedef Documentation

12.268.2.1 BasicCodedEntryVector

```
typedef std::vector< SegmentHelper::BasicCodedEntry > gdcm::Segment::BasicCodedEntryVector
```

12.268.2.2 SurfaceVector

```
typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector
```

12.268.3 Member Enumeration Documentation

12.268.3.1 ALGOType

```
enum gdcm::Segment::ALGOType
```

Enumerator

AUTOMATIC	
SEMIAUTOMATIC	
MANUAL	
ALGOType_END	

12.268.4 Constructor & Destructor Documentation

12.268.4.1 Segment()

```
gdcm::Segment::Segment ()
```

12.268.4.2 ~Segment()

```
gdcm::Segment::~Segment ()  [override]
```

12.268.5 Member Function Documentation

12.268.5.1 AddSurface()

```
void gdcm::Segment::AddSurface (
            SmartPointer< Surface > surface)
```

References gdcm::Object::SmartPointer.

12.268.5.2 GetALGOType()

```
ALGOType gdcm::Segment::GetALGOType (
            const char * type)  [static]
```

12.268.5.3 GetALGOTypeString()

```
const char * gdcm::Segment::GetALGOTypeString (
            ALGOType type)  [static]
```

12.268.5.4 GetAnatomicRegion() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Segment::GetAnatomicRegion ()
```

12.268.5.5 GetAnatomicRegion() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Segment::GetAnatomicRegion () const
```

12.268.5.6 GetAnatomicRegionModifiers() [1/2]

```
BasicCodedEntryVector & gdcm::Segment::GetAnatomicRegionModifiers ()
```

12.268.5.7 GetAnatomicRegionModifiers() [2/2]

```
BasicCodedEntryVector const & gdcm::Segment::GetAnatomicRegionModifiers () const
```

12.268.5.8 GetPropertyCategory() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Segment::GetPropertyCategory ()
```

12.268.5.9 GetPropertyCategory() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Segment::GetPropertyCategory () const
```

12.268.5.10 GetPropertyType() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Segment::GetPropertyType ()
```

12.268.5.11 GetPropertyType() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Segment::GetPropertyType () const
```

12.268.5.12 GetPropertyTypeModifiers() [1/2]

```
BasicCodedEntryVector & gdcm::Segment::GetPropertyTypeModifiers ()
```

12.268.5.13 GetPropertyTypeModifiers() [2/2]

```
BasicCodedEntryVector const & gdcm::Segment::GetPropertyTypeModifiers () const
```

12.268.5.14 GetSegmentAlgorithmName()

```
const char * gdcm::Segment::GetSegmentAlgorithmName () const
```

12.268.5.15 GetSegmentAlgorithmType()

```
ALGOType gdcm::Segment::GetSegmentAlgorithmType () const
```

12.268.5.16 GetSegmentDescription()

```
const char * gdcm::Segment::GetSegmentDescription () const
```

12.268.5.17 GetSegmentLabel()

```
const char * gdcm::Segment::GetSegmentLabel () const
```

12.268.5.18 GetSegmentNumber()

```
unsigned short gdcm::Segment::GetSegmentNumber () const
```

12.268.5.19 GetSurface()

```
SmartPointer< Surface > gdcm::Segment::GetSurface (
            const unsigned int idx = 0) const
```

References gdcm::Object::SmartPointer.

12.268.5.20 GetSurfaceCount()

```
unsigned long gdcm::Segment::GetSurfaceCount ()
```

12.268.5.21 GetSurfaces() [1/2]

```
SurfaceVector & gdcm::Segment::GetSurfaces ()
```

12.268.5.22 GetSurfaces() [2/2]

```
SurfaceVector const & gdcm::Segment::GetSurfaces () const
```

12.268.5.23 SetAnatomicRegion()

```
void gdcm::Segment::SetAnatomicRegion (
            SegmentHelper::BasicCodedEntry const & BSE)
```

12.268.5.24 SetAnatomicRegionModifiers()

```
void gdcm::Segment::SetAnatomicRegionModifiers (
            BasicCodedEntryVector const & BSEV)
```

12.268.5.25 SetPropertyCategory()

```
void gdcm::Segment::SetPropertyCategory (
            SegmentHelper::BasicCodedEntry const & BSE)
```

12.268.5.26 SetPropertyType()

```
void gdcm::Segment::SetPropertyType (
            SegmentHelper::BasicCodedEntry const & BSE)
```

12.268.5.27 SetPropertyTypeModifiers()

```
void gdcm::Segment::SetPropertyTypeModifiers (
            BasicCodedEntryVector const & BSEV)
```

12.268.5.28 SetSegmentAlgorithmName()

```
void gdcm::Segment::SetSegmentAlgorithmName (
            const char * name)
```

12.268.5.29 SetSegmentAlgorithmType() [1/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (
            ALGOType type)
```

12.268.5.30 SetSegmentAlgorithmType() [2/2]

```
void gdcm::Segment::SetSegmentAlgorithmType (
            const char * typeStr)
```

12.268.5.31 SetSegmentDescription()

```
void gdcm::Segment::SetSegmentDescription (
            const char * description)
```

12.268.5.32 SetSegmentLabel()

```
void gdcm::Segment::SetSegmentLabel (
            const char * label)
```

12.268.5.33 SetSegmentNumber()

```
void gdcm::Segment::SetSegmentNumber (
            const unsigned short num)
```

12.268.5.34 SetSurfaceCount()

```
void gdcm::Segment::SetSurfaceCount (
            const unsigned long nb)
```

12.268.6 Member Data Documentation

12.268.6.1 AnatomicRegion

```
SegmentHelper::BasicCodedEntry gdcm::Segment::AnatomicRegion  [protected]
```

12.268.6.2 AnatomicRegionModifiers

```
BasicCodedEntryVector gdcm::Segment::AnatomicRegionModifiers  [protected]
```

12.268.6.3 PropertyCategory

```
SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyCategory  [protected]
```

12.268.6.4 PropertyType

```
SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyType  [protected]
```

12.268.6.5 PropertyTypeModifiers

```
BasicCodedEntryVector gdcm::Segment::PropertyTypeModifiers [protected]
```

12.268.6.6 SegmentAlgorithmName

```
std::string gdcm::Segment::SegmentAlgorithmName [protected]
```

12.268.6.7 SegmentAlgorithmType

```
ALGOType gdcm::Segment::SegmentAlgorithmType [protected]
```

12.268.6.8 SegmentDescription

```
std::string gdcm::Segment::SegmentDescription [protected]
```

12.268.6.9 SegmentLabel

```
std::string gdcm::Segment::SegmentLabel [protected]
```

12.268.6.10 SegmentNumber

```
unsigned short gdcm::Segment::SegmentNumber [protected]
```

12.268.6.11 SurfaceCount

```
unsigned long gdcm::Segment::SurfaceCount [protected]
```

12.268.6.12 Surfaces

```
SurfaceVector gdcm::Segment::Surfaces [protected]
```

The documentation for this class was generated from the following file:

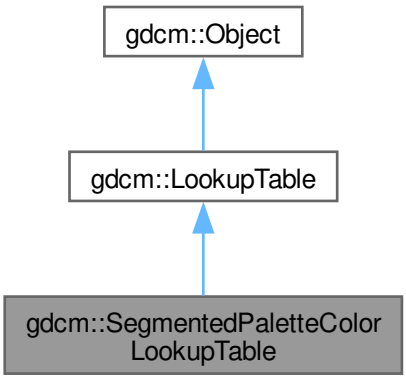
- gdcmSegment.h

12.269 gdcm::SegmentedPaletteColorLookupTable Class Reference

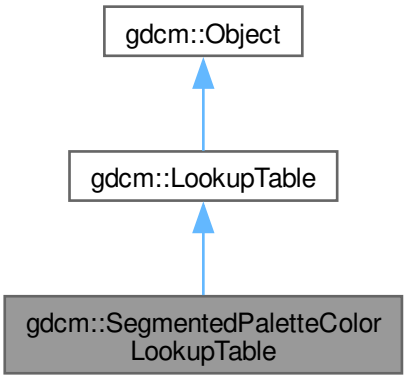
SegmentedPaletteColorLookupTable class.

```
#include <gdcmSegmentedPaletteColorLookupTable.h>
```

Inheritance diagram for gdcm::SegmentedPaletteColorLookupTable:



Collaboration diagram for gdcm::SegmentedPaletteColorLookupTable:



Public Member Functions

- SegmentedPaletteColorLookupTable ()
- ~SegmentedPaletteColorLookupTable () override
- void Print (std::ostream &) const override
- void SetLUT (LookupTableType type, const unsigned char *array, unsigned int length) override

  Initialize a SegmentedPaletteColorLookupTable.

Public Member Functions inherited from gdcm::LookupTable

- LookupTable ()
- LookupTable (LookupTable const &lut)
- ~LookupTable () override
- void Allocate (unsigned short bitsample=8)

  Allocate the LUT.
- void Clear ()

  Clear the LUT.
- bool Decode (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
- void Decode (std::istream &is, std::ostream &os) const

  Decode the LUT.
- bool Decode8 (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const

  Decode into RGB 8 bits space.
- unsigned short GetBitSample () const

  return the bit sample
- bool GetBufferAsRGBA (unsigned char *rgba) const

  return the LUT as RGBA buffer
- void GetLUT (LookupTableType type, unsigned char *array, unsigned int &length) const
- void GetLUTDescriptor (LookupTableType type, unsigned short &length, unsigned short &subscript, unsigned short &bitsize) const
- unsigned int GetLUTLength (LookupTableType type) const
- const unsigned char * GetPointer () const

  return a raw pointer to the LUT
- void InitializeBlueLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)
- bool Initialized () const

  return whether the LUT has been initialized
- void InitializeGreenLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)
- void InitializeLUT (LookupTableType type, unsigned short length, unsigned short subscript, unsigned short bitsize)

  Generic interface:
- void InitializeRedLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)

  RED / GREEN / BLUE specific:
- bool IsRGB8 () const

  Return whether 16 bits LUT is in RGB 8 bits space.
- void SetBlueLUT (const unsigned char *blue, unsigned int length)
- void SetGreenLUT (const unsigned char *green, unsigned int length)
- void SetRedLUT (const unsigned char *red, unsigned int length)
- bool WriteBufferAsRGBA (const unsigned char *rgba)

  Write the LUT as RGBA.

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Additional Inherited Members

Public Types inherited from gdcm::LookupTable

- enum LookupTableType {
  RED = 0 ,
  GREEN ,
  BLUE ,
  GRAY ,
  UNKNOWN }

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes inherited from gdcm::LookupTable

- unsigned short BitSample
- bool IncompleteLUT:1
- LookupTableInternal * Internal

12.269.1 Detailed Description

SegmentedPaletteColorLookupTable class.

12.269.2 Constructor & Destructor Documentation

12.269.2.1 SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ()
```

12.269.2.2 ~SegmentedPaletteColorLookupTable()

```
gdcm::SegmentedPaletteColorLookupTable::~SegmentedPaletteColorLookupTable () [override]
```

12.269.3 Member Function Documentation

12.269.3.1 Print()

```
void gdcm::SegmentedPaletteColorLookupTable::Print (
            std::ostream & ) const  [inline], [override], [virtual]
```

Reimplemented from gdcm::LookupTable.

12.269.3.2 SetLUT()

```
void gdcm::SegmentedPaletteColorLookupTable::SetLUT (
            LookupTableType type,
            const unsigned char * array,
            unsigned int length)  [override], [virtual]
```

Initialize a SegmentedPaletteColorLookupTable.

Reimplemented from gdcm::LookupTable.

The documentation for this class was generated from the following file:

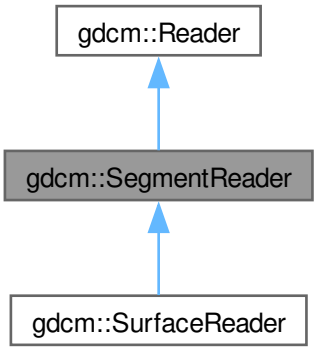
- gdcmSegmentedPaletteColorLookupTable.h

12.270 gdcm::SegmentReader Class Reference

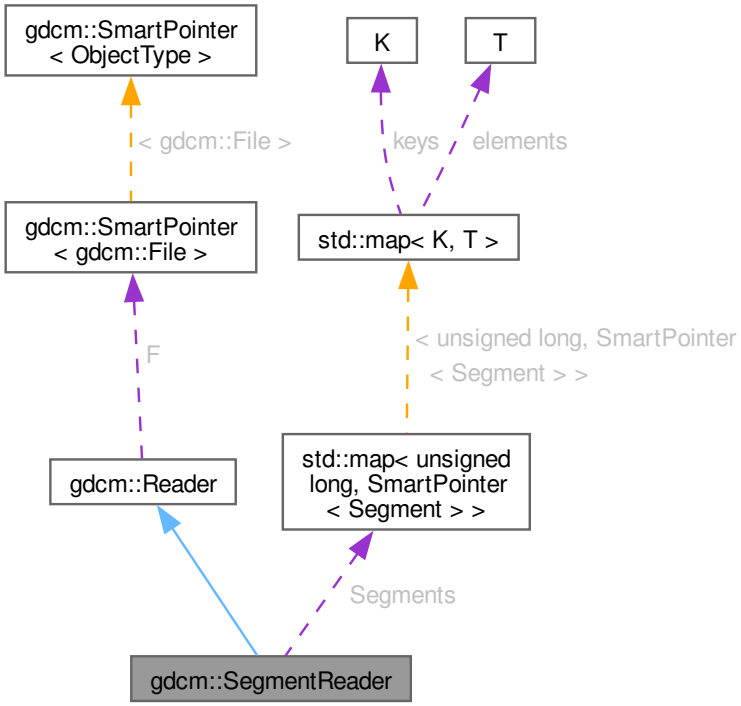
This class defines a segment reader.

```
#include <gdcmSegmentReader.h>
```

Inheritance diagram for gdcm::SegmentReader:



Collaboration diagram for gdcm::SegmentReader:



Public Types

- typedef std::vector< SmartPointer< Segment > > SegmentVector

Public Member Functions

- SegmentReader ()
- ~SegmentReader () override
- SegmentVector GetSegments ()
- SegmentVector GetSegments () const
- bool Read () override

  Read.

Public Member Functions inherited from gdcm::Reader

- Reader ()
- virtual ~Reader ()
- bool CanRead () const

- File & GetFile ()

  Set/Get File.
- const File & GetFile () const

  Set/Get File.
- size_t GetStreamCurrentPosition () const
- bool ReadSelectedPrivateTags (std::set< PrivateTag > const &ptags, bool readvalues=true)

  Will only read the specified selected private tags.
- bool ReadSelectedTags (std::set< Tag > const &tags, bool readvalues=true)

  Will only read the specified selected tags.
- bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())
- void SetFile (File &file)

  Set/Get File.
- void SetFileName (const char *filename_native)
- void SetStream (std::istream &input_stream)

  Set the open-ed stream directly.

Protected Types

- typedef std::map< unsigned long, SmartPointer< Segment > > SegmentMap

Protected Member Functions

- bool ReadSegment (const Item &segmentItem, const unsigned int idx)
- bool ReadSegments ()

Protected Member Functions inherited from gdcm::Reader

- std::istream * GetStreamPtr () const
- bool ReadDataSet ()
- bool ReadMetaInformation ()
- bool ReadPreamble ()

Protected Attributes

- SegmentMap Segments

Protected Attributes inherited from gdcm::Reader

- SmartPointer< File > F

12.270.1 Detailed Description

This class defines a segment reader.

It reads attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

12.270.2 Member Typedef Documentation

12.270.2.1 SegmentMap

```
typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap [protected]
```

12.270.2.2 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector
```

12.270.3 Constructor & Destructor Documentation

12.270.3.1 SegmentReader()

```
gdcm::SegmentReader::SegmentReader ()
```

12.270.3.2 ~SegmentReader()

```
gdcm::SegmentReader::~SegmentReader () [override]
```

12.270.4 Member Function Documentation

12.270.4.1 GetSegments() [1/2]

```
SegmentVector gdcm::SegmentReader::GetSegments ()
```

12.270.4.2 GetSegments() [2/2]

```
SegmentVector gdcm::SegmentReader::GetSegments () const
```

12.270.4.3 Read()

```
bool gdcm::SegmentReader::Read () [override], [virtual]
```

Read.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::SurfaceReader.

12.270.4.4 ReadSegment()

```
bool gdcm::SegmentReader::ReadSegment (
            const Item & segmentItem,
            const unsigned int idx) [protected]
```

12.270.4.5 ReadSegments()

```
bool gdcm::SegmentReader::ReadSegments () [protected]
```

12.270.5 Member Data Documentation

12.270.5.1 Segments

```
SegmentMap gdcm::SegmentReader::Segments [protected]
```

The documentation for this class was generated from the following file:

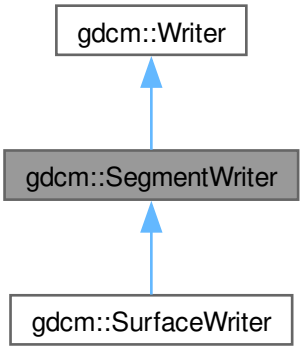
- gdcmSegmentReader.h

12.271 gdcm::SegmentWriter Class Reference

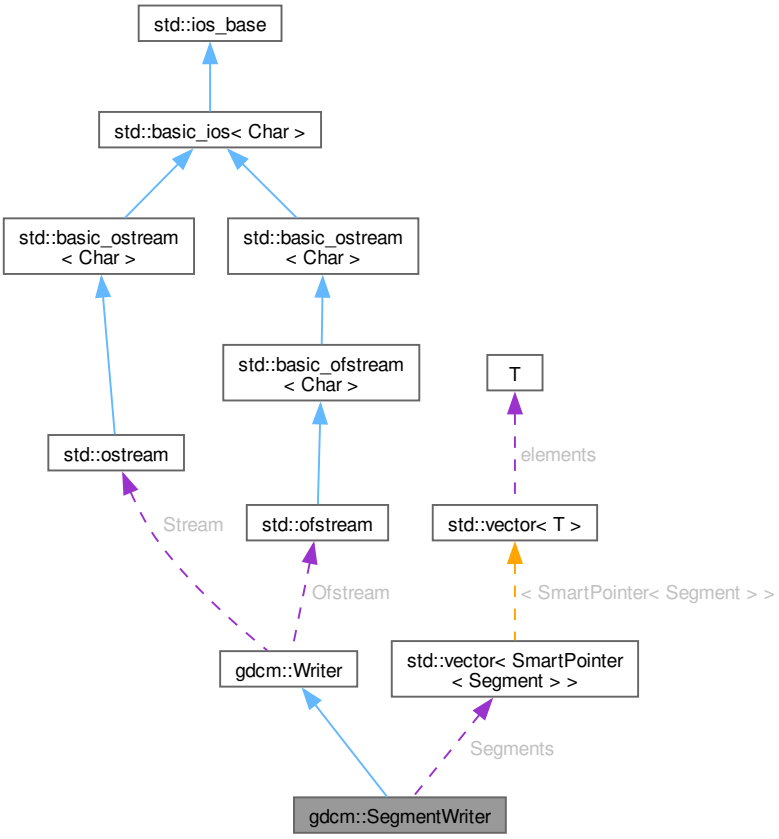
This class defines a segment writer.

```
#include <gdcmSegmentWriter.h>
```

Inheritance diagram for gdcm::SegmentWriter:



Collaboration diagram for gdcm::SegmentWriter:



Public Types

- typedef std::vector< SmartPointer< Segment > > SegmentVector

Public Member Functions

- SegmentWriter ()
- ~SegmentWriter () override
- void AddSegment (SmartPointer< Segment > segment)
- unsigned int GetNumberOfSegments () const
- SmartPointer< Segment > GetSegment (const unsigned int idx=0) const
- SegmentVector & GetSegments ()
- const SegmentVector & GetSegments () const
- void SetNumberOfSegments (const unsigned int size)
- void SetSegments (SegmentVector &segments)
- bool Write () override

  Write.

Public Member Functions inherited from gdcm::Writer

- Writer ()
- virtual ~Writer ()
- void CheckFileMetaInformationOff ()
- void CheckFileMetaInformationOn ()
- File & GetFile ()
- void SetCheckFileMetaInformation (bool b)

  Undocumented function, do not use (= leave default).
- void SetFile (const File &f)

  Set/Get the DICOM file (DataSet + Header).
- void SetFileName (const char *filename_native)

  Set the filename of DICOM file to write:
- void SetStream (std::ostream &output_stream)

  Set user ostream buffer.

Protected Member Functions

- bool PrepareWrite ()

Protected Member Functions inherited from gdcm::Writer

- bool GetCheckFileMetaInformation () const
- std::ostream * GetStreamPtr () const
- void SetWriteDataSetOnly (bool b)

Protected Attributes

- SegmentVector Segments

Protected Attributes inherited from gdcm::Writer

- std::ofstream * Ofstream
- std::ostream * Stream

12.271.1 Detailed Description

This class defines a segment writer.

It writes attributes of group 0x0062.

See also

PS 3.3 C.8.20.2 and C.8.23

12.271.2 Member Typedef Documentation

12.271.2.1 SegmentVector

```
typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector
```

12.271.3 Constructor & Destructor Documentation

12.271.3.1 SegmentWriter()

```
gdcm::SegmentWriter::SegmentWriter ()
```

12.271.3.2 ~SegmentWriter()

```
gdcm::SegmentWriter::~SegmentWriter ()  [override]
```

12.271.4 Member Function Documentation

12.271.4.1 AddSegment()

```
void gdcm::SegmentWriter::AddSegment (
            SmartPointer< Segment > segment)
```

12.271.4.2 GetNumberOfSegments()

```
unsigned int gdcm::SegmentWriter::GetNumberOfSegments () const
```

12.271.4.3 GetSegment()

```
SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (
            const unsigned int idx = 0) const
```

12.271.4.4 GetSegments() [1/2]

```
SegmentVector & gdcm::SegmentWriter::GetSegments ()
```

12.271.4.5 GetSegments() [2/2]

```
const SegmentVector & gdcm::SegmentWriter::GetSegments () const
```

12.271.4.6 PrepareWrite()

```
bool gdcm::SegmentWriter::PrepareWrite ()  [protected]
```

12.271.4.7 SetNumberOfSegments()

```
void gdcm::SegmentWriter::SetNumberOfSegments (
            const unsigned int size)
```

12.271.4.8 SetSegments()

```
void gdcm::SegmentWriter::SetSegments (
            SegmentVector & segments)
```

12.271.4.9 Write()

```
bool gdcm::SegmentWriter::Write ()  [override], [virtual]
```

Write.

Reimplemented from gdcm::Writer.

Reimplemented in gdcm::SurfaceWriter.

12.271.5 Member Data Documentation

12.271.5.1 Segments

```
SegmentVector gdcm::SegmentWriter::Segments  [protected]
```

The documentation for this class was generated from the following file:

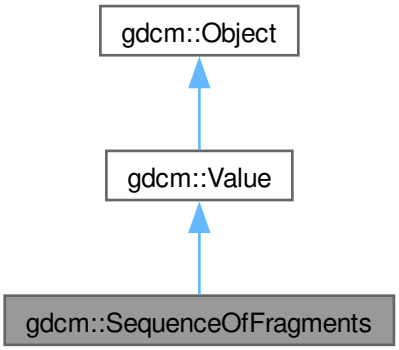
- gdcmSegmentWriter.h

12.272 gdcm::SequenceOfFragments Class Reference

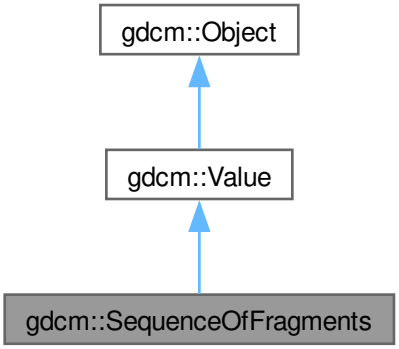
Class to represent a Sequence Of Fragments.

`#include <gdcmSequenceOfFragments.h>`

Inheritance diagram for gdcm::SequenceOfFragments:



Collaboration diagram for gdcm::SequenceOfFragments:



Public Types

- typedef FragmentVector::const_iterator ConstIterator
- typedef std::vector< Fragment > FragmentVector
- typedef FragmentVector::iterator Iterator
- typedef FragmentVector::size_type SizeType

Public Member Functions

- SequenceOfFragments ()

  constructor (UndefinedLength by default)
- void AddFragment (Fragment const &item)

  Appends a Fragment to the already added ones.
- Iterator Begin ()
- ConstIterator Begin () const
- void Clear () override

  Clear.
- unsigned long ComputeByteLength () const
- VL ComputeLength () const
- Iterator End ()
- ConstIterator End () const
- bool GetBuffer (char *buffer, unsigned long length) const
- bool GetFragBuffer (unsigned int fragNb, char *buffer, unsigned long &length) const
- const Fragment & GetFragment (SizeType num) const
- VL GetLength () const override

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfFragments () const
- BasicOffsetTable & GetTable ()
- const BasicOffsetTable & GetTable () const
- bool operator== (const Value &val) const override
- void Print (std::ostream &os) const override
- template<typename TSwap>
  std::istream & Read (std::istream &is, bool readvalues=true)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is, bool)
- void SetLength (VL length) override

  Sets the actual SQ length.
- template<typename TSwap>
  std::ostream const & Write (std::ostream &os) const
- bool WriteBuffer (std::ostream &os) const

Public Member Functions inherited from gdcm::Value

- Value ()=default
- ~Value () override=default

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Static Public Member Functions

- static SmartPointer< SequenceOfFragments > New ()

Additional Inherited Members

Protected Member Functions inherited from gdcm::Value

- virtual void SetLengthOnly (VL l)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.272.1 Detailed Description

Class to represent a Sequence Of Fragments.

Todo I do not enforce that Sequence of Fragments ends with a SQ end del

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, and MpegVideoInfo.cs.

12.272.2 Member Typedef Documentation

12.272.2.1 ConstIterator

```
typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator
```

12.272.2.2 FragmentVector

```
typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector
```

12.272.2.3 Iterator

```
typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator
```

12.272.2.4 SizeType

```
typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType
```

12.272.3 Constructor & Destructor Documentation

12.272.3.1 SequenceOfFragments()

```
gdcm::SequenceOfFragments::SequenceOfFragments () [inline]
```

constructor (UndefinedLength by default)

Referenced by New(), and operator==().

12.272.4 Member Function Documentation

12.272.4.1 AddFragment()

```
void gdcm::SequenceOfFragments::AddFragment (
            Fragment const & item)
```

Appends a Fragment to the already added ones.

12.272.4.2 Begin() [1/2]

```
Iterator gdcm::SequenceOfFragments::Begin () [inline]
```

Referenced by Print(), and Write().

12.272.4.3 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::Begin () const [inline]
```

12.272.4.4 Clear()

```
void gdcm::SequenceOfFragments::Clear () [override], [virtual]
```

Clear.

Implements gdcm::Value.

12.272.4.5 ComputeByteLength()

```
unsigned long gdcm::SequenceOfFragments::ComputeByteLength () const
```

12.272.4.6 ComputeLength()

```
VL gdcm::SequenceOfFragments::ComputeLength () const
```

12.272.4.7 End() [1/2]

```
Iterator gdcm::SequenceOfFragments::End ()  [inline]
```

Referenced by Print(), and Write().

12.272.4.8 End() [2/2]

```
ConstIterator gdcm::SequenceOfFragments::End () const  [inline]
```

12.272.4.9 GetBuffer()

```
bool gdcm::SequenceOfFragments::GetBuffer (
            char * buffer,
            unsigned long length) const
```

12.272.4.10 GetFragBuffer()

```
bool gdcm::SequenceOfFragments::GetFragBuffer (
            unsigned int fragNb,
            char * buffer,
            unsigned long & length) const
```

12.272.4.11 GetFragment()

```
const Fragment & gdcm::SequenceOfFragments::GetFragment (
            SizeType num) const
```

Examples

DecompressImage.cs, FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

12.272.4.12 GetLength()

```
VL gdcm::SequenceOfFragments::GetLength () const  [inline], [override], [virtual]
```

Returns the SQ length, as read from disk.

Implements gdcm::Value.

12.272.4.13 GetNumberOfFragments()

```
SizeType gdcm::SequenceOfFragments::GetNumberOfFragments () const
```

Examples

FixJAIBugJPEGLS.cxx.

12.272.4.14 GetTable() [1/2]

```
BasicOffsetTable & gdcm::SequenceOfFragments::GetTable ()  [inline]
```

12.272.4.15 GetTable() [2/2]

```
const BasicOffsetTable & gdcm::SequenceOfFragments::GetTable () const  [inline]
```

12.272.4.16 New()

```
SmartPointer< SequenceOfFragments > gdcm::SequenceOfFragments::New ()  [inline], [static]
```

Examples

DecompressImageMultiframe.cs, DecompressJPEGFile.cs, and MpegVideoInfo.cs.

References SequenceOfFragments().

12.272.4.17 operator==()

```
bool gdcm::SequenceOfFragments::operator== (
            const Value & val) const  [inline], [override], [virtual]
```

Implements gdcm::Value.

References SequenceOfFragments(), and gdcm::Value::Value().

12.272.4.18 Print()

```
void gdcm::SequenceOfFragments::Print (
            std::ostream & os) const  [inline], [override], [virtual]
```

Reimplemented from gdcm::Object.

References Begin(), and End().

12.272.4.19 Read()

```
template<typename TSwap>
std::istream & gdcm::SequenceOfFragments::Read (
            std::istream & is,
            bool readvalues = true)  [inline]
```

References ReadPreValue(), and ReadValue().

12.272.4.20 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::SequenceOfFragments::ReadPreValue (
            std::istream & is)  [inline]
```

References gdcmDebugMacro.

Referenced by Read().

12.272.4.21 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::SequenceOfFragments::ReadValue (
            std::istream & is,
            bool )  [inline]
```

References gdcmAssertAlwaysMacro, gdcmDebugMacro, gdcmWarningMacro, gdcm::Tag::GetElement(), gdcm::Tag::GetGroup(), gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVL(), gdcm::Fragment::Read(), gdcm::Fragment::ReadBacktrack(), and gdcm::Exception::what().

Referenced by Read().

12.272.4.22 SetLength()

```
void gdcm::SequenceOfFragments::SetLength (
            VL length)  [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements gdcm::Value.

12.272.4.23 Write()

```
template<typename TSwap>
std::ostream const & gdcm::SequenceOfFragments::Write (
            std::ostream & os) const  [inline]
```

References Begin(), End(), gdcm::Tag::Write(), and gdcm::VL::Write().

12.272.4.24 WriteBuffer()

```
bool gdcm::SequenceOfFragments::WriteBuffer (
            std::ostream & os) const
```

Examples

GetJPEGSamplePrecision.cxx.

The documentation for this class was generated from the following file:

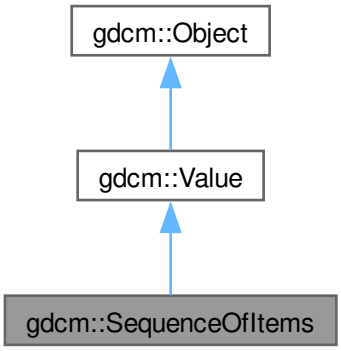
- gdcmSequenceOfFragments.h

12.273 gdcm::SequenceOfItems Class Reference

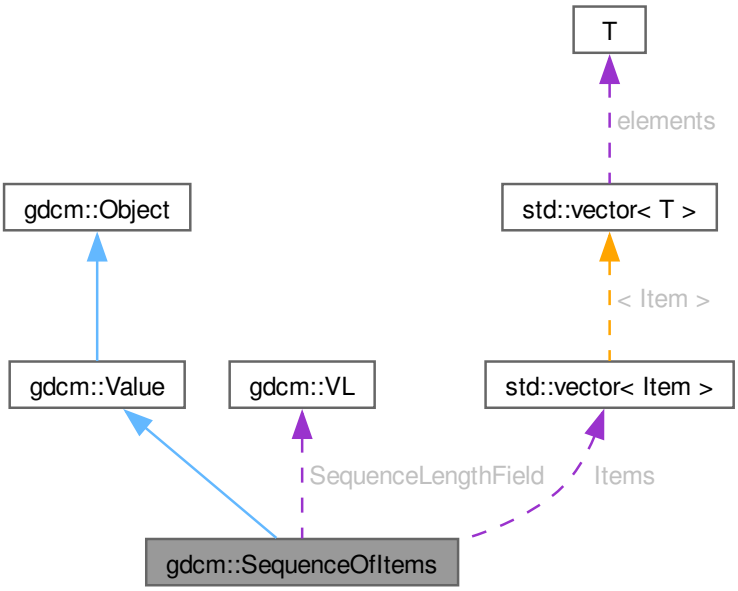
Class to represent a Sequence Of Items.

```
#include <gdcmSequenceOfItems.h>
```

Inheritance diagram for gdcm::SequenceOfItems:



Collaboration diagram for gdcm::SequenceOfItems:



Public Types

- typedef ItemVector::const_iterator ConstIterator
- typedef std::vector< Item > ItemVector
- typedef ItemVector::iterator Iterator
- typedef ItemVector::size_type SizeType

Public Member Functions

- SequenceOfItems ()

  constructor (UndefinedLength by default)
- void AddItem (Item const &item)

  Appends an Item to the already added ones.
- Item & AddNewUndefinedLengthItem ()

  Appends an Item to the already added ones.
- Iterator Begin ()
- ConstIterator Begin () const
- void Clear () override

  remove all items within the sequence
- template<typename TDE>
  VL ComputeLength () const
- Iterator End ()

- ConstIterator End () const
- bool FindDataElement (const Tag &t) const
- Item & GetItem (SizeType position)
- const Item & GetItem (SizeType position) const
- VL GetLength () const override

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfItems () const
- bool IsEmpty () const
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- SequenceOfItems & operator= (const SequenceOfItems &val)
- bool operator== (const Value &val) const override
- void Print (std::ostream &os) const override
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is, bool readvalues=true)
- bool RemoveItemByIndex (const SizeType index)
- void SetLength (VL length) override

  Sets the actual SQ length.
- void SetLengthToUndefined ()

  Properly set the Sequence of Item to be undefined length.
- void SetNumberOfItems (SizeType n)
- template<typename TDE, typename TSwap>
  std::ostream const & Write (std::ostream &os) const

Public Member Functions inherited from gdcm::Value

- Value ()=default
- ~Value () override=default

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Static Public Member Functions

- static SmartPointer< SequenceOfItems > New ()

Public Attributes

- ItemVector Items

  Vector of Sequence Items.
- VL SequenceLengthField

  Total length of the Sequence (or 0xffffffff) if undefined.

Additional Inherited Members

Protected Member Functions inherited from gdcm::Value

- virtual void SetLengthOnly (VL l)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.273.1 Detailed Description

Class to represent a Sequence Of Items.

(value representation : SQ)

- a Value Representation for Data Elements that contains a sequence of Data Sets.
- Sequence of Item allows for Nested Data Sets

See PS 3.5, 7.4.6 Data Element Type Within a Sequence

Note

SEQUENCE OF ITEMS (VALUE REPRESENTATION SQ) A Value Representation for Data Elements that contain a sequence of Data Sets. Sequence of Items allows for Nested Data Sets.

Examples

DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, and ReadExplicitLengthSQIVR.cxx.

12.273.2 Member Typedef Documentation

12.273.2.1 ConstIterator

```
typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator
```

12.273.2.2 ItemVector

```
typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector
```

12.273.2.3 Iterator

```
typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator
```

12.273.2.4 SizeType

```
typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType
```

Examples

DumpExamCard.cxx, and DumpGEMSMovieGroup.cxx.

12.273.3 Constructor & Destructor Documentation

12.273.3.1 SequenceOfItems()

```
gdcm::SequenceOfItems::SequenceOfItems () [inline]
```

constructor (UndefinedLength by default)

References SequenceLengthField.

Referenced by New(), operator=(), and operator==().

12.273.4 Member Function Documentation

12.273.4.1 AddItem()

```
void gdcm::SequenceOfItems::AddItem (
            Item const & item)
```

Appends an Item to the already added ones.

Examples

Extracting_All_Resolution.cxx.

12.273.4.2 AddNewUndefinedLengthItem()

```
Item & gdcm::SequenceOfItems::AddNewUndefinedLengthItem ()
```

Appends an Item to the already added ones.

12.273.4.3 Begin() [1/2]

```
Iterator gdcm::SequenceOfItems::Begin () [inline]
```

References Items.

12.273.4.4 Begin() [2/2]

```
ConstIterator gdcm::SequenceOfItems::Begin () const [inline]
```

References Items.

12.273.4.5 Clear()

```
void gdcm::SequenceOfItems::Clear () [override], [virtual]
```

remove all items within the sequence

Implements gdcm::Value.

12.273.4.6 ComputeLength()

```
template<typename TDE>
VL gdcm::SequenceOfItems::ComputeLength () const
```

12.273.4.7 End() [1/2]

```
Iterator gdcm::SequenceOfItems::End () [inline]
```

References Items.

12.273.4.8 End() [2/2]

```
ConstIterator gdcm::SequenceOfItems::End () const [inline]
```

References Items.

12.273.4.9 FindDataElement()

```
bool gdcm::SequenceOfItems::FindDataElement (
            const Tag & t) const
```

12.273.4.10 GetItem() [1/2]

```
Item & gdcm::SequenceOfItems::GetItem (
            SizeType position)
```

12.273.4.11 GetItem() [2/2]

```
const Item & gdcm::SequenceOfItems::GetItem (
            SizeType position) const
```

Examples

DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, and GetSequenceUltrasound.cxx.

12.273.4.12 GetLength()

```
VL gdcm::SequenceOfItems::GetLength () const  [inline], [override], [virtual]
```

Returns the SQ length, as read from disk.

Implements gdcm::Value.

References SequenceLengthField.

Referenced by Read().

12.273.4.13 GetNumberOfItems()

```
SizeType gdcm::SequenceOfItems::GetNumberOfItems () const  [inline]
```

Examples

DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, ExtractEncryptedContent.cxx, and GetSequenceUltrasound.cxx.

References Items.

12.273.4.14 IsEmpty()

```
bool gdcm::SequenceOfItems::IsEmpty () const  [inline]
```

References Items.

12.273.4.15 IsUndefinedLength()

```
bool gdcm::SequenceOfItems::IsUndefinedLength () const  [inline]
```

return if Value Length if of undefined length

References SequenceLengthField.

12.273.4.16 New()

```
SmartPointer< SequenceOfItems > gdcm::SequenceOfItems::New ()  [inline], [static]
```

Examples

NewSequence.cs.

References SequenceOfItems().

12.273.4.17 operator=()

```
SequenceOfItems & gdcm::SequenceOfItems::operator= (
            const SequenceOfItems & val)  [inline]
```

References SequenceOfItems(), Items, and SequenceLengthField.

12.273.4.18 operator==()

```
bool gdcm::SequenceOfItems::operator== (
            const Value & val) const  [inline], [override], [virtual]
```

Implements gdcm::Value.

References SequenceOfItems(), gdcm::Value::Value(), Items, and SequenceLengthField.

12.273.4.19 Print()

```
void gdcm::SequenceOfItems::Print (
            std::ostream & os) const  [inline], [override], [virtual]
```

Reimplemented from gdcm::Object.

References Items, and SequenceLengthField.

12.273.4.20 Read()

```
template<typename TDE, typename TSwap>
std::istream & gdcm::SequenceOfItems::Read (
            std::istream & is,
            bool readvalues = true)  [inline]
```

References gdcm::Item::Clear(), gdcmDebugMacro, gdcmWarningMacro, gdcm::Exception::GetDescription(), GetLength(), gdcm::Item::GetNestedDataSet(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVL(), Items, gdcm::Item::Read(), SequenceLengthField, and gdcm::DataSet::Size().

12.273.4.21 RemoveItemByIndex()

```
bool gdcm::SequenceOfItems::RemoveItemByIndex (
            const SizeType index)
```

Remove an Item as specified by its index, if index > size, false is returned Index starts at 1 not 0

12.273.4.22 SetLength()

```
void gdcm::SequenceOfItems::SetLength (
            VL length)  [inline], [override], [virtual]
```

Sets the actual SQ length.

Implements gdcm::Value.

References SequenceLengthField.

12.273.4.23 SetLengthToUndefined()

```
void gdcm::SequenceOfItems::SetLengthToUndefined ()
```

Properly set the Sequence of Item to be undefined length.

12.273.4.24 SetNumberOfItems()

```
void gdcm::SequenceOfItems::SetNumberOfItems (
            SizeType n)  [inline]
```

References Items.

12.273.4.25 Write()

```
template<typename TDE, typename TSwap>
std::ostream const & gdcm::SequenceOfItems::Write (
            std::ostream & os) const  [inline]
```

References Items, SequenceLengthField, gdcm::Tag::Write(), and gdcm::VL::Write().

12.273.5 Member Data Documentation

12.273.5.1 Items

```
ItemVector gdcm::SequenceOfItems::Items
```

Vector of Sequence Items.

Referenced by Begin(), Begin(), End(), End(), GetNumberOfItems(), IsEmpty(), operator=(), operator==(), Print(), Read(), SetNumberOfItems(), and Write().

12.273.5.2 SequenceLengthField

```
VL gdcm::SequenceOfItems::SequenceLengthField
```

Total length of the Sequence (or 0xffffffff) if undefined.

Referenced by SequenceOfItems(), GetLength(), IsUndefinedLength(), operator=(), operator==(), Print(), Read(), SetLength(), and Write().

The documentation for this class was generated from the following file:

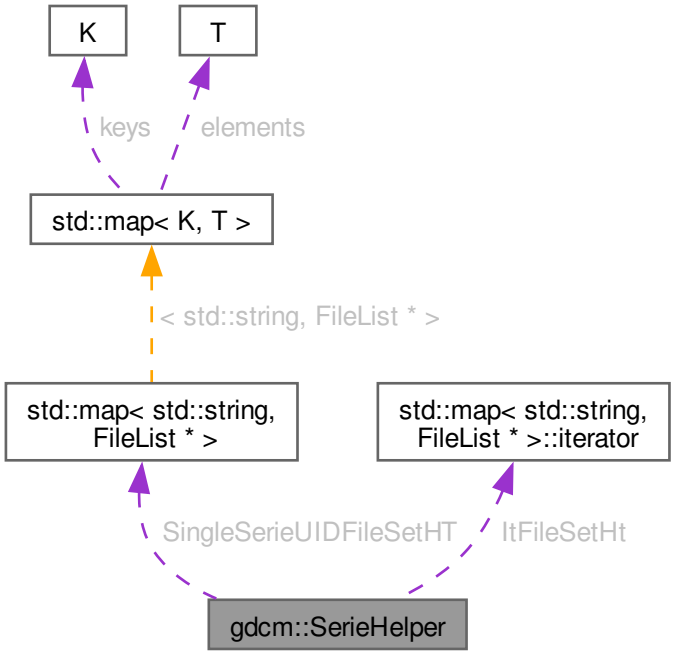
- gdcmSequenceOfItems.h

12.274 gdcm::SerieHelper Class Reference

SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

```
#include <gdcmSerieHelper.h>
```

Collaboration diagram for gdcm::SerieHelper:



Public Member Functions

- SerieHelper ()
- ~SerieHelper ()
- void AddRestriction (const std::string &tag)
- void AddRestriction (uint16_t group, uint16_t elem, std::string const &value, int op)
- void Clear ()
- void CreateDefaultUniqueSeriesIdentifier ()
- std::string CreateUniqueSeriesIdentifier (File *inFile)
- FileList * GetFirstSingleSerieUIDFileSet ()
- FileList * GetNextSingleSerieUIDFileSet ()
- void OrderFileList (FileList *fileSet)
- void SetDirectory (std::string const &dir, bool recursive=false)
- void SetLoadMode (int)
- void SetUseSeriesDetails (bool useSeriesDetails)

Protected Types

- using Rule
- typedef std::vector< Rule > SerieRestrictions
- typedef std::map< std::string, FileList * > SingleSerieUIDFileSetmap

Protected Member Functions

- bool AddFile (FileWithName &header)
- void AddFileName (std::string const &filename)
- void AddRestriction (const Tag &tag)
- bool FileNameOrdering (FileList *fileList)
- bool ImageNumberOrdering (FileList *fileList)
- bool ImagePositionPatientOrdering (FileList *fileSet)
- bool UserOrdering (FileList *fileSet)

Protected Attributes

- SingleSerieUIDFileSetmap::iterator ItFileSetHt
- SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT

12.274.1 Detailed Description

SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

Instead see ImageHelper or IPPSorter

12.274.2 Member Typedef Documentation

12.274.2.1 Rule

```
using gdcm::SerieHelper::Rule  [protected]
```

Initial value:

```
struct RuleStructure{
   uint16_t group;
   uint16_t elem;
   std::string value;
   int op;
 }
```

12.274.2.2 SerieRestrictions

```
typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions  [protected]
```

12.274.2.3 SingleSerieUIDFileSetmap

```
typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap  [protected]
```

12.274.3 Constructor & Destructor Documentation

12.274.3.1 SerieHelper()

```
gdcm::SerieHelper::SerieHelper ()
```

12.274.3.2 ~SerieHelper()

```
gdcm::SerieHelper::~SerieHelper ()
```

12.274.4 Member Function Documentation

12.274.4.1 AddFile()

```
bool gdcm::SerieHelper::AddFile (
            FileWithName & header)  [protected]
```

12.274.4.2 AddFileName()

```
void gdcm::SerieHelper::AddFileName (
            std::string const & filename)  [protected]
```

12.274.4.3 AddRestriction() [1/3]

```
void gdcm::SerieHelper::AddRestriction (
            const std::string & tag)
```

12.274.4.4 AddRestriction() [2/3]

```
void gdcm::SerieHelper::AddRestriction (
            const Tag & tag)  [protected]
```

12.274.4.5 AddRestriction() [3/3]

```
void gdcm::SerieHelper::AddRestriction (
            uint16_t group,
            uint16_t elem,
            std::string const & value,
            int op)
```

12.274.4.6 Clear()

```
void gdcm::SerieHelper::Clear ()
```

12.274.4.7 CreateDefaultUniqueSeriesIdentifier()

```
void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ()
```

12.274.4.8 CreateUniqueSeriesIdentifier()

```
std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (
            File * inFile)
```

12.274.4.9 FileNameOrdering()

```
bool gdcm::SerieHelper::FileNameOrdering (
            FileList * fileList)  [protected]
```

12.274.4.10 GetFirstSingleSerieUIDFileSet()

```
FileList * gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ()
```

12.274.4.11 GetNextSingleSerieUIDFileSet()

```
FileList * gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ()
```

12.274.4.12 ImageNumberOrdering()

```
bool gdcm::SerieHelper::ImageNumberOrdering (
            FileList * fileList)  [protected]
```

12.274.4.13 ImagePositionPatientOrdering()

```
bool gdcm::SerieHelper::ImagePositionPatientOrdering (
            FileList * fileSet)  [protected]
```

12.274.4.14 OrderFileList()

```
void gdcm::SerieHelper::OrderFileList (
            FileList * fileSet)
```

12.274.4.15 SetDirectory()

```
void gdcm::SerieHelper::SetDirectory (
            std::string const & dir,
            bool recursive = false)
```

12.274.4.16 SetLoadMode()

```
void gdcm::SerieHelper::SetLoadMode (
            int )  [inline]
```

12.274.4.17 SetUseSeriesDetails()

```
void gdcm::SerieHelper::SetUseSeriesDetails (
            bool useSeriesDetails)
```

12.274.4.18 UserOrdering()

```
bool gdcm::SerieHelper::UserOrdering (
            FileList * fileSet)  [protected]
```

12.274.5 Member Data Documentation

12.274.5.1 ItFileSetHt

```
SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt  [protected]
```

12.274.5.2 SingleSerieUIDFileSetHT

```
SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT  [protected]
```

The documentation for this class was generated from the following file:

- gdcmSerieHelper.h

12.275 gdcm::Series Class Reference

Series.

```
#include <gdcmSeries.h>
```

Public Member Functions

- Series ()=default

12.275.1 Detailed Description

Series.

12.275.2 Constructor & Destructor Documentation

12.275.2.1 Series()

```
gdcm::Series::Series () [default]
```

The documentation for this class was generated from the following file:

- gdcmSeries.h

12.276 gdcm::network::ServiceClassApplicationInformation Class Reference

```
#include <gdcmServiceClassApplicationInformation.h>
```

Public Member Functions

- ServiceClassApplicationInformation ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetTuple (uint8_t levelofsupport, uint8_t levelofdigitalsig, uint8_t elementcoercion)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.276.1 Detailed Description

PS 3.4 Table B.3-1 SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

12.276.2 Constructor & Destructor Documentation

12.276.2.1 ServiceClassApplicationInformation()

```
gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ()
```

12.276.3 Member Function Documentation

12.276.3.1 Print()

```
void gdcm::network::ServiceClassApplicationInformation::Print (
            std::ostream & os) const
```

12.276.3.2 Read()

```
std::istream & gdcm::network::ServiceClassApplicationInformation::Read (
            std::istream & is)
```

12.276.3.3 SetTuple()

```
void gdcm::network::ServiceClassApplicationInformation::SetTuple (
            uint8_t levelofsupport,
            uint8_t levelofdigitalsig,
            uint8_t elementcoercion)
```

12.276.3.4 Size()

```
size_t gdcm::network::ServiceClassApplicationInformation::Size () const
```

12.276.3.5 Write()

```
const std::ostream & gdcm::network::ServiceClassApplicationInformation::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

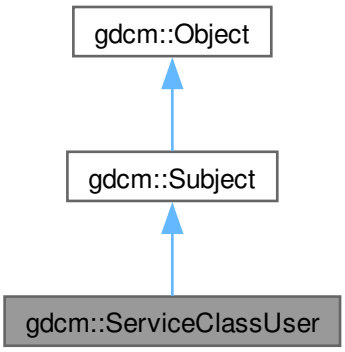
- gdcmServiceClassApplicationInformation.h

12.277 gdcm::ServiceClassUser Class Reference

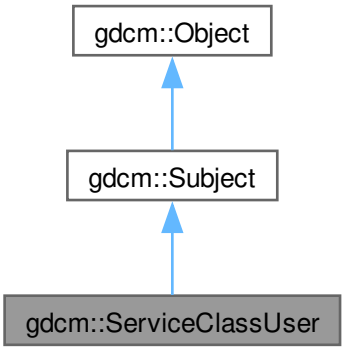
ServiceClassUser.

```
#include <gdcmServiceClassUser.h>
```

Inheritance diagram for gdcm::ServiceClassUser:



Collaboration diagram for gdcm::ServiceClassUser:



Public Member Functions

- ServiceClassUser ()
- ServiceClassUser (const ServiceClassUser &)=delete
- ~ServiceClassUser () override
- const char * GetAETitle () const
- const char * GetCalledAETitle () const
- double GetTimeout () const
- bool InitializeConnection ()
- bool IsPresentationContextAccepted (const PresentationContext &pc) const

  Return if the passed in presentation was accepted during association negotiation.
- void operator= (const ServiceClassUser &)=delete
- bool SendEcho ()

  C-ECHO.
- bool SendFind (const BaseRootQuery *query, std::vector< DataSet > &retDatasets)

  C-FIND a query, return result are in retDatasets.
- bool SendMove (const BaseRootQuery *query, const char *outputdir)

  Execute a C-MOVE, based on query, return files are written in outputdir.
- bool SendMove (const BaseRootQuery *query, std::vector< DataSet > &retDatasets)

  Execute a C-MOVE, based on query, returned dataset are Implicit.
- bool SendMove (const BaseRootQuery *query, std::vector< File > &retFile)

  Execute a C-MOVE, based on query, returned Files are stored in vector.
- bool SendStore (const char *filename)

  Execute a C-STORE on file on disk, named filename.
- bool SendStore (DataSet const &ds)

  Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.
- bool SendStore (File const &file)
- void SetAETitle (const char *aetitle)

  set calling ae title
- void SetCalledAETitle (const char *aetitle)

  set called ae title
- void SetHostname (const char *hostname)

  Set the name of the called hostname (hostname or IP address).
- void SetPort (uint16_t port)

  Set port of remote host (called application).
- void SetPortSCP (uint16_t portscp)

  Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE).
- void SetPresentationContexts (std::vector< PresentationContext > const &pcs)

  Set the Presentation Context used for the Association.
- void SetTimeout (double t)

  set/get Timeout
- bool StartAssociation ()

  Start the association. Need to call SetPresentationContexts before.
- bool StopAssociation ()

  Stop the running association.

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static SmartPointer< ServiceClassUser > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.277.1 Detailed Description

ServiceClassUser.

Examples

CStoreQtProgress.cxx.

12.277.2 Constructor & Destructor Documentation

12.277.2.1 ServiceClassUser() [1/2]

```
gdcm::ServiceClassUser::ServiceClassUser ()
```

Construct a SCU with default:

- hostname = localhost
- port = 104

Referenced by ServiceClassUser(), New(), and operator=().

12.277.2.2 ~ServiceClassUser()

```
gdcm::ServiceClassUser::~ServiceClassUser ()  [override]
```

12.277.2.3 ServiceClassUser() [2/2]

```
gdcm::ServiceClassUser::ServiceClassUser (
            const ServiceClassUser & )  [delete]
```

References ServiceClassUser().

12.277.3 Member Function Documentation

12.277.3.1 GetAETitle()

```
const char * gdcm::ServiceClassUser::GetAETitle () const
```

12.277.3.2 GetCalledAETitle()

```
const char * gdcm::ServiceClassUser::GetCalledAETitle () const
```

12.277.3.3 GetTimeout()

```
double gdcm::ServiceClassUser::GetTimeout () const
```

12.277.3.4 InitializeConnection()

```
bool gdcm::ServiceClassUser::InitializeConnection ()
```

Will try to connect This will setup the actual timeout used during the whole connection time. Need to call SetTimeout first

Examples

CStoreQtProgress.cxx.

12.277.3.5 IsPresentationContextAccepted()

```
bool gdcm::ServiceClassUser::IsPresentationContextAccepted (
            const PresentationContext & pc) const
```

Return if the passed in presentation was accepted during association negotiation.

12.277.3.6 New()

```
SmartPointer< ServiceClassUser > gdcm::ServiceClassUser::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

References ServiceClassUser().

12.277.3.7 operator=()

```
void gdcm::ServiceClassUser::operator= (
            const ServiceClassUser & ) [delete]
```

References ServiceClassUser().

12.277.3.8 SendEcho()

```
bool gdcm::ServiceClassUser::SendEcho ()
```

C-ECHO.

12.277.3.9 SendFind()

```
bool gdcm::ServiceClassUser::SendFind (
            const BaseRootQuery * query,
            std::vector< DataSet > & retDatasets)
```

C-FIND a query, return result are in retDatasets.

12.277.3.10 SendMove() [1/3]

```
bool gdcm::ServiceClassUser::SendMove (
            const BaseRootQuery * query,
            const char * outputdir)
```

Execute a C-MOVE, based on query, return files are written in outputdir.

12.277.3.11 SendMove() [2/3]

```
bool gdcm::ServiceClassUser::SendMove (
            const BaseRootQuery * query,
            std::vector< DataSet > & retDatasets)
```

Execute a C-MOVE, based on query, returned dataset are Implicit.

12.277.3.12 SendMove() [3/3]

```
bool gdcm::ServiceClassUser::SendMove (
            const BaseRootQuery * query,
            std::vector< File > & retFile)
```

Execute a C-MOVE, based on query, returned Files are stored in vector.

12.277.3.13 SendStore() [1/3]

```
bool gdcm::ServiceClassUser::SendStore (
            const char * filename)
```

Execute a C-STORE on file on disk, named filename.

Examples

CStoreQtProgress.cxx.

12.277.3.14 SendStore() [2/3]

```
bool gdcm::ServiceClassUser::SendStore (
            DataSet const & ds)
```

Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.

12.277.3.15 SendStore() [3/3]

```
bool gdcm::ServiceClassUser::SendStore (
            File const & file)
```

Execute a C-STORE on a File, the transfer syntax used for the query is based on the file.

12.277.3.16 SetAETitle()

```
void gdcm::ServiceClassUser::SetAETitle (
            const char * aetitle)
```

set calling ae title

12.277.3.17 SetCalledAETitle()

```
void gdcm::ServiceClassUser::SetCalledAETitle (
            const char * aetitle)
```

set called ae title

Examples

CStoreQtProgress.cxx.

12.277.3.18 SetHostname()

```
void gdcm::ServiceClassUser::SetHostname (
            const char * hostname)
```

Set the name of the called hostname (hostname or IP address).

Examples

CStoreQtProgress.cxx.

12.277.3.19 SetPort()

```
void gdcm::ServiceClassUser::SetPort (
            uint16_t port)
```

Set port of remote host (called application).

Examples

CStoreQtProgress.cxx.

12.277.3.20 SetPortSCP()

```
void gdcm::ServiceClassUser::SetPortSCP (
            uint16_t portscp)
```

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE).

12.277.3.21 SetPresentationContexts()

```
void gdcm::ServiceClassUser::SetPresentationContexts (
            std::vector< PresentationContext > const & pcs)
```

Set the Presentation Context used for the Association.

Examples

CStoreQtProgress.cxx.

12.277.3.22 SetTimeout()

```
void gdcm::ServiceClassUser::SetTimeout (
            double t)
```

set/get Timeout

Examples

CStoreQtProgress.cxx.

12.277.3.23 StartAssociation()

```
bool gdcm::ServiceClassUser::StartAssociation ()
```

Start the association. Need to call SetPresentationContexts before.

Examples

CStoreQtProgress.cxx.

12.277.3.24 StopAssociation()

```
bool gdcm::ServiceClassUser::StopAssociation ()
```

Stop the running association.

Examples

CStoreQtProgress.cxx.

The documentation for this class was generated from the following file:

- gdcmServiceClassUser.h

12.278 gdcm::SHA1 Class Reference

Class for SHA1.

```
#include <gdcmSHA1.h>
```

Public Member Functions

- SHA1 ()
- SHA1 (const SHA1 &)=delete
- ~SHA1 ()
- void operator= (const SHA1 &)=delete

Static Public Member Functions

- static bool Compute (const char *buffer, unsigned long buf_len, char digest_str[20 *2+1])
- static bool ComputeFile (const char *filename, char digest_str[20 *2+1])

12.278.1 Detailed Description

Class for SHA1.

Warning

this class is able to pick from one implementation:

1. the one from OpenSSL (when GDCM_USE_SYSTEM_OPENSSL is turned ON)

In all other cases it will return an error

12.278.2 Constructor & Destructor Documentation

12.278.2.1 SHA1() [1/2]

```
gdcm::SHA1::SHA1 ()
```

Referenced by SHA1(), and operator=().

12.278.2.2 ~SHA1()

```
gdcm::SHA1::~SHA1 ()
```

12.278.2.3 SHA1() [2/2]

```
gdcm::SHA1::SHA1 (
            const SHA1 & )  [delete]
```

References SHA1().

12.278.3 Member Function Documentation

12.278.3.1 Compute()

```
bool gdcm::SHA1::Compute (
            const char * buffer,
            unsigned long buf_len,
            char digest_str[20 *2+1])  [static]
```

12.278.3.2 ComputeFile()

```
bool gdcm::SHA1::ComputeFile (
            const char * filename,
            char digest_str[20 *2+1])  [static]
```

12.278.3.3 operator=()

```
void gdcm::SHA1::operator= (
            const SHA1 & )  [delete]
```

References SHA1().

The documentation for this class was generated from the following file:

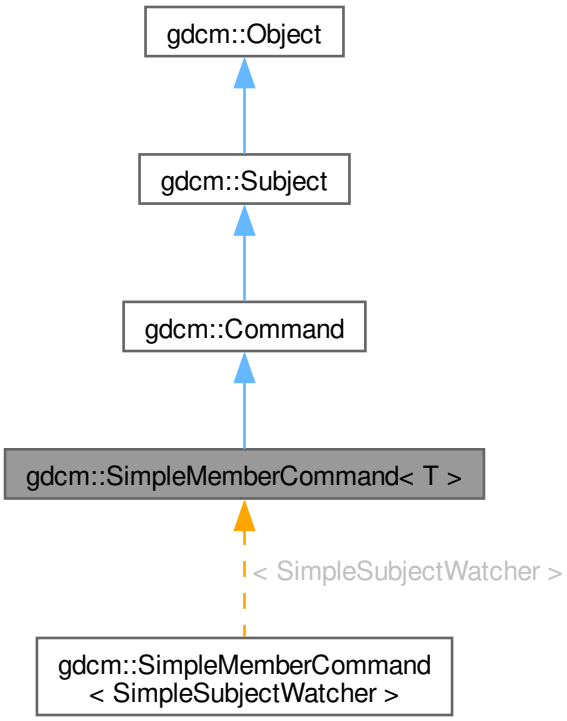
- gdcmSHA1.h

12.279 gdcm::SimpleMemberCommand< T > Class Template Reference

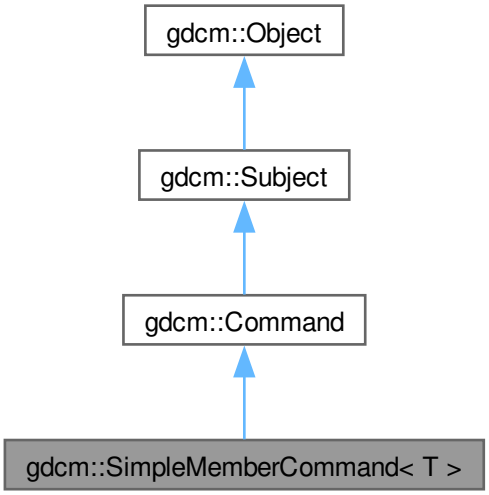
Command subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```

Inheritance diagram for gdcm::SimpleMemberCommand< T >:



Collaboration diagram for gdcm::SimpleMemberCommand< T >:



Public Types

- typedef SimpleMemberCommand Self
- typedef void(T::* TMemberFunctionPointer) ()

Public Member Functions

- SimpleMemberCommand (const Self &)=delete
- void Execute (const Subject *, const Event &) override
- void Execute (Subject *, const Event &) override
- void operator= (const Self &)=delete
- void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)

Public Member Functions inherited from gdcm::Command

- Command (const Command &)=delete
- void operator= (const Command &)=delete

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static SmartPointer< SimpleMemberCommand > New ()

Protected Member Functions

- SimpleMemberCommand ()
- ~SimpleMemberCommand () override=default

Protected Member Functions inherited from gdcm::Command

- Command ()
- ~Command () override

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes

- TMemberFunctionPointer m_MemberFunction
- T * m_This

12.279.1 Detailed Description

template<typename T>
class gdcm::SimpleMemberCommand< T >

Command subclass that calls a pointer to a member function.

SimpleMemberCommand calls a pointer to a member function with no arguments.

12.279.2 Member Typedef Documentation

12.279.2.1 Self

```
template<typename T>
typedef SimpleMemberCommand gdcm::SimpleMemberCommand< T >::Self
```

Standard class typedefs.

12.279.2.2 TMemberFunctionPointer

```
template<typename T>
typedef void(T::* gdcm::SimpleMemberCommand< T >::TMemberFunctionPointer) ()
```

A method callback.

12.279.3 Constructor & Destructor Documentation

12.279.3.1 SimpleMemberCommand() [1/2]

```
template<typename T>
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand (
            const Self & )  [delete]
```

12.279.3.2 SimpleMemberCommand() [2/2]

```
template<typename T>
gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ()  [inline], [protected]
```

12.279.3.3 ~SimpleMemberCommand()

```
template<typename T>
gdcm::SimpleMemberCommand< T >::~SimpleMemberCommand ()  [override], [protected], [default]
```

12.279.4 Member Function Documentation

12.279.4.1 Execute() [1/2]

```
template<typename T>
void gdcm::SimpleMemberCommand< T >::Execute (
            const Subject * caller,
            const Event & event)  [inline], [override], [virtual]
```

Abstract method that defines the action to be taken by the command. This variant is expected to be used when requests comes from a const Object

Implements gdcm::Command.

12.279.4.2 Execute() [2/2]

```
template<typename T>
void gdcm::SimpleMemberCommand< T >::Execute (
            Subject * ,
            const Event & )  [inline], [override], [virtual]
```

Invoke the callback function.

Implements gdcm::Command.

12.279.4.3 New()

```
template<typename T>
SmartPointer< SimpleMemberCommand > gdcm::SimpleMemberCommand< T >::New ()  [inline], [static]
```

Run-time type information (and related methods). Method for creation through the object factory.

12.279.4.4 operator=()

```
template<typename T>
void gdcm::SimpleMemberCommand< T >::operator= (
            const Self & )  [delete]
```

12.279.4.5 SetCallbackFunction()

```
template<typename T>
void gdcm::SimpleMemberCommand< T >::SetCallbackFunction (
            T * object,
            TMemberFunctionPointer memberFunction)  [inline]
```

Specify the callback function.

12.279.5 Member Data Documentation

12.279.5.1 m_MemberFunction

```
template<typename T>
TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction [protected]
```

12.279.5.2 m_This

```
template<typename T>
T* gdcm::SimpleMemberCommand< T >::m_This [protected]
```

The documentation for this class was generated from the following file:

- gdcmCommand.h

12.280 gdcm::SimpleSubjectWatcher Class Reference

SimpleSubjectWatcher.

```
#include <gdcmSimpleSubjectWatcher.h>
```

Public Member Functions

- SimpleSubjectWatcher (const SimpleSubjectWatcher &)=delete
- SimpleSubjectWatcher (Subject *s, const char *comment="")
- virtual ~SimpleSubjectWatcher ()
- void operator= (const SimpleSubjectWatcher &)=delete

Protected Member Functions

- virtual void EndFilter ()
- virtual void ShowAbort ()
- virtual void ShowAnonymization (Subject *caller, const Event &evt)
- virtual void ShowData (Subject *caller, const Event &evt)
- virtual void ShowDataSet (Subject *caller, const Event &evt)
- virtual void ShowFileName (Subject *caller, const Event &evt)
- virtual void ShowIteration ()
- virtual void ShowProgress (Subject *caller, const Event &evt)
- virtual void StartFilter ()
- void TestAbortOff ()
- void TestAbortOn ()

12.280.1 Detailed Description

SimpleSubjectWatcher.

This is a typical Subject Watcher class. It will observe all events.

Examples

BasicAnonymizer.cs, CStoreQtProgress.cxx, Cleaner.cs, ClinicalTrialIdentificationWorkflow.cs, FileChangeTS.cs, FileChangeTSLossy.cs, ScanDirectory.cs, and SimpleScanner.cxx.

12.280.2 Constructor & Destructor Documentation

12.280.2.1 SimpleSubjectWatcher() [1/2]

```
gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (
            Subject * s,
            const char * comment = "")
```

Referenced by SimpleSubjectWatcher(), and operator=().

12.280.2.2 ~SimpleSubjectWatcher()

```
virtual gdcm::SimpleSubjectWatcher::~SimpleSubjectWatcher ()  [virtual]
```

12.280.2.3 SimpleSubjectWatcher() [2/2]

```
gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (
            const SimpleSubjectWatcher & )  [delete]
```

References SimpleSubjectWatcher().

12.280.3 Member Function Documentation

12.280.3.1 EndFilter()

```
virtual void gdcm::SimpleSubjectWatcher::EndFilter ()  [protected], [virtual]
```

12.280.3.2 operator=()

```
void gdcm::SimpleSubjectWatcher::operator= (
            const SimpleSubjectWatcher & )  [delete]
```

References SimpleSubjectWatcher().

12.280.3.3 ShowAbort()

```
virtual void gdcm::SimpleSubjectWatcher::ShowAbort ()  [protected], [virtual]
```

12.280.3.4 ShowAnonymization()

```
virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (
            Subject * caller,
            const Event & evt)  [protected], [virtual]
```

12.280.3.5 ShowData()

```
virtual void gdcm::SimpleSubjectWatcher::ShowData (
            Subject * caller,
            const Event & evt)  [protected], [virtual]
```

12.280.3.6 ShowDataSet()

```
virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (
            Subject * caller,
            const Event & evt)  [protected], [virtual]
```

12.280.3.7 ShowFileName()

```
virtual void gdcm::SimpleSubjectWatcher::ShowFileName (
            Subject * caller,
            const Event & evt)  [protected], [virtual]
```

Examples

SimpleScanner.cxx.

12.280.3.8 ShowIteration()

```
virtual void gdcm::SimpleSubjectWatcher::ShowIteration ()  [protected], [virtual]
```

12.280.3.9 ShowProgress()

```
virtual void gdcm::SimpleSubjectWatcher::ShowProgress (
            Subject * caller,
            const Event & evt)  [protected], [virtual]
```

12.280.3.10 StartFilter()

```
virtual void gdcm::SimpleSubjectWatcher::StartFilter () [protected], [virtual]
```

12.280.3.11 TestAbortOff()

```
void gdcm::SimpleSubjectWatcher::TestAbortOff () [protected]
```

12.280.3.12 TestAbortOn()

```
void gdcm::SimpleSubjectWatcher::TestAbortOn () [protected]
```

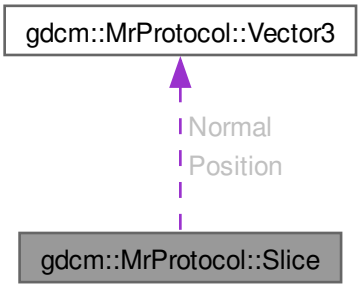
The documentation for this class was generated from the following file:

- gdcmSimpleSubjectWatcher.h

12.281 gdcm::MrProtocol::Slice Struct Reference

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for gdcm::MrProtocol::Slice:



Public Attributes

- Vector3 Normal
- Vector3 Position

12.281.1 Member Data Documentation

12.281.1.1 Normal

```
Vector3 gdcm::MrProtocol::Slice::Normal
```

12.281.1.2 Position

```
Vector3 gdcm::MrProtocol::Slice::Position
```

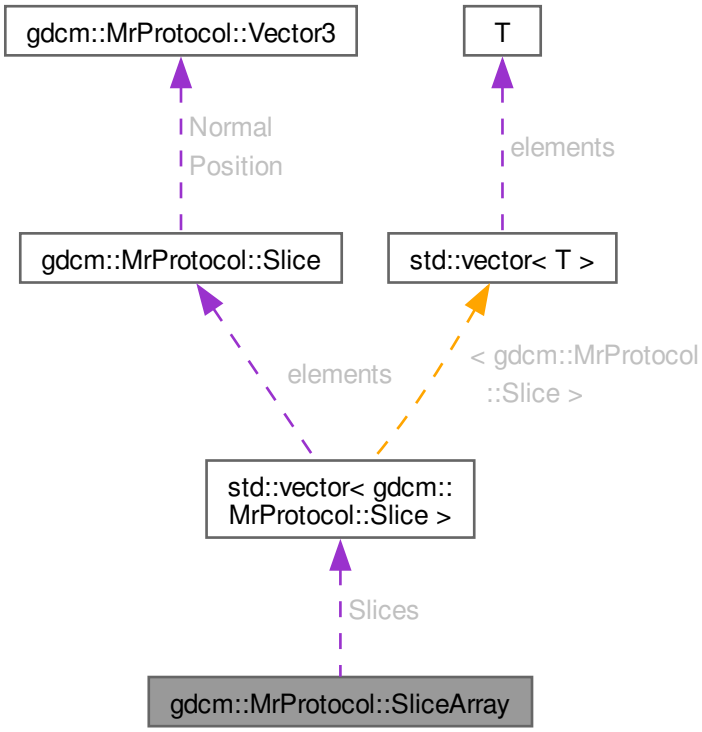
The documentation for this struct was generated from the following file:

- gdcmMrProtocol.h

12.282 gdcm::MrProtocol::SliceArray Struct Reference

```
#include <gdcmMrProtocol.h>
```

Collaboration diagram for gdcm::MrProtocol::SliceArray:



Public Attributes

- std::vector< Slice > Slices

12.282.1 Member Data Documentation

12.282.1.1 Slices

`std::vector< Slice > gdcm::MrProtocol::SliceArray::Slices`

The documentation for this struct was generated from the following file:

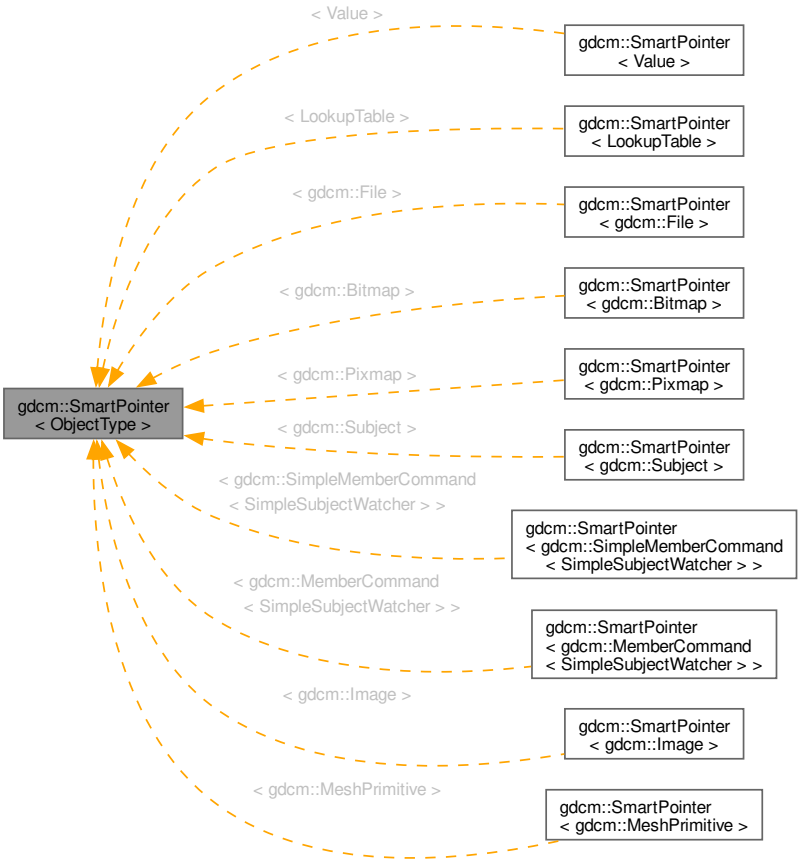
- gdcmMrProtocol.h

12.283 gdcm::SmartPointer< ObjectType > Class Template Reference

Class for Smart Pointer.

`#include <gdcmSmartPointer.h>`

Inheritance diagram for gdcm::SmartPointer< ObjectType >:



Public Member Functions

- SmartPointer ()
- SmartPointer (const SmartPointer< ObjectType > &p)
- SmartPointer (ObjectType *p)
- SmartPointer (ObjectType const &p)
- ~SmartPointer ()
- ObjectType * GetPointer () const

  Explicit function to retrieve the pointer.
- operator ObjectType * () const

  Return pointer to object.
- ObjectType & operator* () const
- ObjectType * operator-> () const

  Overload operator ->.
- SmartPointer & operator= (ObjectType *r)

  Overload operator assignment.
- SmartPointer & operator= (ObjectType const &r)
- SmartPointer & operator= (SmartPointer const &r)

  Overload operator assignment.

12.283.1 Detailed Description

template<class ObjectType>
class gdcm::SmartPointer< ObjectType >

Class for Smart Pointer.

Will only work for subclass of gdcm::Object See tr1/shared_ptr for a more general approach (not invasive) #include <tr1/memory> { shared_ptr<Bla> b(new Bla); }

Note

Class partly based on post by Bill Hubauer: `http://groups.google.com/group/comp.lang.c++/msg/173ddc38a827a930`

See also

`http://www.davethehat.com/articles/smartp.htm`

and itk::SmartPointer

Examples

CStoreQtProgress.cxx, ChangeSequenceUltrasound.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, DumpVisusChange.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_ FixBrokenJ2K.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, SimpleScanner.cxx, gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.283.2 Constructor & Destructor Documentation

12.283.2.1 SmartPointer() [1/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer ()  [inline]
```

12.283.2.2 SmartPointer() [2/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
            const SmartPointer< ObjectType > & p)  [inline]
```

12.283.2.3 SmartPointer() [3/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
            ObjectType * p)  [inline]
```

12.283.2.4 SmartPointer() [4/4]

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::SmartPointer (
            ObjectType const & p)  [inline]
```

12.283.2.5 ~SmartPointer()

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::~SmartPointer ()  [inline]
```

12.283.3 Member Function Documentation

12.283.3.1 GetPointer()

```
template<class ObjectType>
ObjectType * gdcm::SmartPointer< ObjectType >::GetPointer () const  [inline]
```

Explicit function to retrieve the pointer.

12.283.3.2 operator ObjectType *()

```
template<class ObjectType>
gdcm::SmartPointer< ObjectType >::operator ObjectType * () const  [inline]
```

Return pointer to object.

12.283.3.3 operator*()

```
template<class ObjectType>
ObjectType & gdcm::SmartPointer< ObjectType >::operator* () const  [inline]
```

12.283.3.4 operator->()

```
template<class ObjectType>
ObjectType * gdcm::SmartPointer< ObjectType >::operator-> () const  [inline]
```

Overload operator ->.

12.283.3.5 operator=() [1/3]

```
template<class ObjectType>
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
            ObjectType * r)  [inline]
```

Overload operator assignment.

12.283.3.6 operator=() [2/3]

```
template<class ObjectType>
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
            ObjectType const & r)  [inline]
```

12.283.3.7 operator=() [3/3]

```
template<class ObjectType>
SmartPointer & gdcm::SmartPointer< ObjectType >::operator= (
            SmartPointer< ObjectType > const & r)  [inline]
```

Overload operator assignment.

Referenced by gdcm::SmartPointer< Value >::operator=(), and gdcm::SmartPointer< Value >::operator=().

The documentation for this class was generated from the following files:

- gdcmObject.h
- gdcmSmartPointer.h

12.284 gdcm::network::SOPClassExtendedNegociationSub Class Reference

SOPClassExtendedNegociationSub.

```
#include <gdcmSOPClassExtendedNegociationSub.h>
```

Public Member Functions

- SOPClassExtendedNegociationSub ()
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetTuple (const char *uid, uint8_t levelofsupport=3, uint8_t levelofdigitalsig=0, uint8_t elementcoercion=2)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.284.1 Detailed Description

SOPClassExtendedNegociationSub.

PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-↩ ASSOCIATE-AC)

12.284.2 Constructor & Destructor Documentation

12.284.2.1 SOPClassExtendedNegociationSub()

```
gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ()
```

12.284.3 Member Function Documentation

12.284.3.1 Print()

```
void gdcm::network::SOPClassExtendedNegociationSub::Print (
            std::ostream & os) const
```

12.284.3.2 Read()

```
std::istream & gdcm::network::SOPClassExtendedNegociationSub::Read (
            std::istream & is)
```

12.284.3.3 SetTuple()

```
void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (
            const char * uid,
            uint8_t levelofsupport = 3,
            uint8_t levelofdigitalsig = 0,
            uint8_t elementcoercion = 2)
```

12.284.3.4 Size()

```
size_t gdcm::network::SOPClassExtendedNegociationSub::Size () const
```

12.284.3.5 Write()

```
const std::ostream & gdcm::network::SOPClassExtendedNegociationSub::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

- gdcmSOPClassExtendedNegociationSub.h

12.285 gdcm::SOPClassUIDToIOD Class Reference

Class convert a class SOP Class UID into IOD.

```
#include <gdcmSOPClassUIDToIOD.h>
```

Public Types

- typedef const char ∗ const(SOPClassUIDToIODType)[2]

Static Public Member Functions

- static const char ∗ GetIOD (UIDs const &uid)
- static const char ∗ GetIODFromSOPClassUID (const char ∗sopclassuid)
- static unsigned int GetNumberOfSOPClassToIOD ()

  Return the number of SOP Class UID listed internally.
- static const char ∗ GetSOPClassUIDFromIOD (const char ∗iod)
- static SOPClassUIDToIODType & GetSOPClassUIDToIOD (unsigned int i)
- static SOPClassUIDToIODType ∗ GetSOPClassUIDToIODs ()

12.285.1 Detailed Description

Class convert a class SOP Class UID into IOD.

Reference PS 3.4 Table B.5-1 STANDARD SOP CLASSES

12.285.2 Member Typedef Documentation

12.285.2.1 const

```
typedef const char * gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]
```

12.285.3 Member Function Documentation

12.285.3.1 GetIOD()

```
const char * gdcm::SOPClassUIDToIOD::GetIOD (
            UIDs const & uid)  [static]
```

Return the associated IOD based on a SOP Class UID uid (there is a one-to-one mapping from SOP Class UID to matching IOD)

Examples

GenerateStandardSOPClasses.cxx.

12.285.3.2 GetIODFromSOPClassUID()

```
const char * gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (
            const char * sopclassuid)  [static]
```

12.285.3.3 GetNumberOfSOPClassToIOD()

```
unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ()  [static]
```

Return the number of SOP Class UID listed internally.

12.285.3.4 GetSOPClassUIDFromIOD()

```
const char * gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (
            const char * iod)  [static]
```

12.285.3.5 GetSOPClassUIDToIOD()

```
SOPClassUIDToIODType & gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (
            unsigned int i)  [static]
```

12.285.3.6 GetSOPClassUIDToIODs()

```
SOPClassUIDToIODType * gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ()  [static]
```

The documentation for this class was generated from the following file:

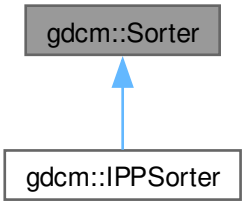
- gdcmSOPClassUIDToIOD.h

12.286 gdcm::Sorter Class Reference

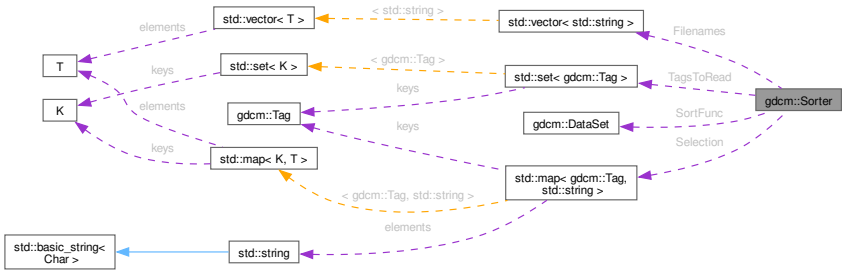
Sorter.

```
#include <gdcmSorter.h>
```

Inheritance diagram for gdcm::Sorter:



Collaboration diagram for gdcm::Sorter:



Public Types

- typedef bool(∗ SortFunction) (DataSet const &, DataSet const &)

  Set the sort function which compares one dataset to the other.

Public Member Functions

- Sorter ()
- virtual ∼Sorter ()
- bool AddSelect (Tag const &tag, const char ∗value)

  UNSUPPORTED FOR NOW.
- const std::vector< std::string > & GetFilenames () const
- void Print (std::ostream &os) const

  Print.
- void SetSortFunction (SortFunction f)
- void SetTagsToRead (std::set< Tag > const &tags)
- virtual bool Sort (std::vector< std::string > const &filenames)

  Typically the output of Directory::GetFilenames().
- virtual bool StableSort (std::vector< std::string > const &filenames)

Protected Types

- typedef std::map< Tag, std::string > SelectionMap

Protected Attributes

- std::vector< std::string > Filenames
- std::map< Tag, std::string > Selection
- SortFunction SortFunc
- std::set< Tag > TagsToRead

Friends

- std::ostream & operator<< (std::ostream &_os, const Sorter &s)

12.286.1 Detailed Description

Sorter.

General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction

Warning

implementation details. For now there is no cache mechanism. Which means that every time you call Sort, all files specified as input parameter are *read*

See also

Scanner

Examples

SortImage.cxx, SortImage2.cs, and VolumeSorter.cxx.

12.286.2 Member Typedef Documentation

12.286.2.1 SelectionMap

```
typedef std::map<Tag,std::string> gdcm::Sorter::SelectionMap  [protected]
```

12.286.2.2 SortFunction

```
typedef bool(* gdcm::Sorter::SortFunction) (DataSet const &, DataSet const &)
```

Set the sort function which compares one dataset to the other.

12.286.3 Constructor & Destructor Documentation

12.286.3.1 Sorter()

```
gdcm::Sorter::Sorter ()
```

Referenced by operator<<.

12.286.3.2 ~Sorter()

```
virtual gdcm::Sorter::~Sorter ()  [virtual]
```

12.286.4 Member Function Documentation

12.286.4.1 AddSelect()

```
bool gdcm::Sorter::AddSelect (
            Tag const & tag,
            const char * value)
```

UNSUPPORTED FOR NOW.

12.286.4.2 GetFilenames()

```
const std::vector< std::string > & gdcm::Sorter::GetFilenames () const  [inline]
```

Return the list of filenames as sorted by the specific algorithm used. Empty by default (before Sort() is called)

Examples

Compute3DSpacing.cxx, SortImage.cxx, VolumeSorter.cxx, gdcmorthoplanes.cxx, and reslicesphere.cxx.

References Filenames.

12.286.4.3 Print()

```
void gdcm::Sorter::Print (
            std::ostream & os) const
```

Print.

Examples

SortImage.cxx, VolumeSorter.cxx, and gdcmorthoplanes.cxx.

Referenced by operator<<.

12.286.4.4 SetSortFunction()

```
void gdcm::Sorter::SetSortFunction (
            SortFunction f)
```

Examples

SortImage.cxx, SortImage2.cs, and VolumeSorter.cxx.

12.286.4.5 SetTagsToRead()

```
void gdcm::Sorter::SetTagsToRead (
            std::set< Tag > const & tags)
```

Specify a set of tags to be read in during the sort procedure. By default this set is empty, in which case the entire image, including pixel data, is read in.

12.286.4.6 Sort()

```
virtual bool gdcm::Sorter::Sort (
            std::vector< std::string > const & filenames)  [virtual]
```

Typically the output of Directory::GetFilenames().

Reimplemented in gdcm::IPPSorter.

Examples

SortImage.cxx.

12.286.4.7 StableSort()

```
virtual bool gdcm::Sorter::StableSort (
            std::vector< std::string > const & filenames)  [virtual]
```

Examples

SortImage.cxx, and VolumeSorter.cxx.

12.286.5 Friends And Related Symbol Documentation

12.286.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Sorter & s)  [friend]
```

References Sorter(), and Print().

12.286.6 Member Data Documentation

12.286.6.1 Filenames

```
std::vector<std::string> gdcm::Sorter::Filenames  [protected]
```

Referenced by GetFilenames().

12.286.6.2 Selection

```
std::map<Tag,std::string> gdcm::Sorter::Selection  [protected]
```

12.286.6.3 SortFunc

```
SortFunction gdcm::Sorter::SortFunc  [protected]
```

12.286.6.4 TagsToRead

```
std::set<Tag> gdcm::Sorter::TagsToRead  [protected]
```

The documentation for this class was generated from the following file:

- gdcmSorter.h

12.287 gdcm::Spacing Class Reference

Class for Spacing.

```
#include <gdcmSpacing.h>
```

Public Types

- enum SpacingType {
  DETECTOR = 0 ,
  MAGNIFIED ,
  CALIBRATED ,
  UNKNOWN }

Public Member Functions

- Spacing ()
- ~Spacing ()=default

Static Public Member Functions

- static Attribute< 0x28, 0x34 > ComputePixelAspectRatioFromPixelSpacing (const Attribute< 0x28, 0x30 > &pixelspacing)

12.287.1 Detailed Description

Class for Spacing.

It all began with a mail to WG6:

Subject: Imager Pixel Spacing vs Pixel Spacing Body: [Apologies for the duplicate post, namely to David Clunie & OFFIS team]

I have been trying to understand CP-586 in the following two cases:

On the one hand:

- DISCIMG/IMAGES/CRIMAGE taken from `http://dclunie.com/images/pixelspacingtestimages.zip`

And on the other hand:

- `http://gdcm.sourceforge.net/thingies/cr_pixelspacing.dcm`

If I understand correctly the CP, one is required to use Pixel Spacing for measurement ('true size' print) instead of Imager Pixel Spacing, since the two attributes are present and Pixel Spacing is different from Imager Pixel Spacing.

If this is correct, then the test data DISCIMG/IMAGES/CRIMAGE is incorrect. If this is incorrect (ie. I need to use Imager Pixel Spacing), then the display of cr_pixelspacing.dcm for measurement will be incorrect.

Could someone please let me know what am I missing here? I could not find any information in any header that would allow me to differentiate those.

Thank you for your time,

Ref: `http://lists.nema.org/scripts/lyris.pl?sub=488573&id=400720477`

See PS 3.3-2008, Table C.7-11b IMAGE PIXEL MACRO ATTRIBUTES

Ratio of the vertical size and horizontal size of the pixels in the image specified by a pair of integer values where the first value is the vertical pixel size, and the second value is the horizontal pixel size. Required if the aspect ratio values do not have a ratio of 1:1 and the physical pixel spacing is not specified by Pixel Spacing (0028,0030), or Imager Pixel Spacing (0018,1164) or Nominal Scanned Pixel Spacing (0018,2010), either for the entire Image or per-frame in a Functional Group Macro. See C.7.6.3.1.7.

PS 3.3-2008 10.7.1.3 Pixel Spacing Value Order and Valid Values All pixel spacing related attributes shall have non-zero values, except when there is only a single row or column or pixel of data present, in which case the corresponding value may be zero.

Ref: `http://gdcm.sourceforge.net/wiki/index.php/Imager_Pixel_Spacing`

12.287.2 Member Enumeration Documentation

12.287.2.1 SpacingType

`enum gdcm::Spacing::SpacingType`

Enumerator

DETECTOR	
MAGNIFIED	
CALIBRATED	
UNKNOWN	

12.287.3 Constructor & Destructor Documentation

12.287.3.1 Spacing()

`gdcm::Spacing::Spacing ()`

12.287.3.2 ~Spacing()

```
gdcm::Spacing::~Spacing () [default]
```

12.287.4 Member Function Documentation

12.287.4.1 ComputePixelAspectRatioFromPixelSpacing()

```
Attribute< 0x28, 0x34 > gdcm::Spacing::ComputePixelAspectRatioFromPixelSpacing (
            const Attribute< 0x28, 0x30 > & pixelspacing) [static]
```

The documentation for this class was generated from the following file:

- gdcmSpacing.h

12.288 gdcm::Spectroscopy Class Reference

Spectroscopy class.

```
#include <gdcmSpectroscopy.h>
```

Public Member Functions

- Spectroscopy ()=default

12.288.1 Detailed Description

Spectroscopy class.

12.288.2 Constructor & Destructor Documentation

12.288.2.1 Spectroscopy()

```
gdcm::Spectroscopy::Spectroscopy () [default]
```

The documentation for this class was generated from the following file:

- gdcmSpectroscopy.h

12.289 gdcm::SplitMosaicFilter Class Reference

SplitMosaicFilter class.

```
#include <gdcmSplitMosaicFilter.h>
```

Public Member Functions

- SplitMosaicFilter ()
- ~SplitMosaicFilter ()
- bool ComputeMOSAICDimensions (unsigned int dims[3])
- bool ComputeMOSAICImagePositionPatient (double pos[3], const double ipp[6], const double dircos[6], const double pixelspacing[3], const unsigned int image_dims[3], const unsigned int mosaic_dims[3], bool inverted)

  Extract the value for ImagePositionPatient.
- bool ComputeMOSAICSliceNormal (double dims[3], bool &inverted)

  Extract the value for SliceNormalVector (CSA header).
- bool ComputeMOSAICSlicePosition (double pos[3], bool inverted)
- File & GetFile ()
- const File & GetFile () const
- Image & GetImage ()
- const Image & GetImage () const
- void SetFile (const File &f)
- void SetImage (const Image &image)
- bool Split ()

  Split the SIEMENS MOSAIC image.

Static Public Member Functions

- static bool GetAcquisitionSize (unsigned int size[2], DataSet const &ds)

  Get the Acquisition Matrix (non zero value):
- static unsigned int GetNumberOfImagesInMosaic (File const &file)

  Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.

12.289.1 Detailed Description

SplitMosaicFilter class.

Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA Image Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture

Warning

when private attributes are not found, the acquisition matrix is used to compute the NumberOfImagesInMosaic. This means trailing black slices will be considered in the volume (instead of discarded). CSA 0029,1010 is needed for correct NumberOfImagesInMosaic CSA 0029,1020 is needed to compute the correct origin without above info default are taken (may not be accurate).

12.289.2 Constructor & Destructor Documentation

12.289.2.1 SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::SplitMosaicFilter ()
```

12.289.2.2 ∼SplitMosaicFilter()

```
gdcm::SplitMosaicFilter::∼SplitMosaicFilter ()
```

12.289.3 Member Function Documentation

12.289.3.1 ComputeMOSAICDimensions()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (
            unsigned int dims[3])
```

Compute the new dimensions according to private information stored in the MOSAIC header.

12.289.3.2 ComputeMOSAICImagePositionPatient()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICImagePositionPatient (
            double pos[3],
            const double ipp[6],
            const double dircos[6],
            const double pixelspacing[3],
            const unsigned int image_dims[3],
            const unsigned int mosaic_dims[3],
            bool inverted)
```

Extract the value for ImagePositionPatient.

12.289.3.3 ComputeMOSAICSliceNormal()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSliceNormal (
            double dims[3],
            bool & inverted)
```

Extract the value for SliceNormalVector (CSA header).

12.289.3.4 ComputeMOSAICSlicePosition()

```
bool gdcm::SplitMosaicFilter::ComputeMOSAICSlicePosition (
            double pos[3],
            bool inverted)
```

Extract the value for ImagePositionPatient (requires inverted flag) Deprecated

12.289.3.5 GetAcquisitionSize()

```
bool gdcm::SplitMosaicFilter::GetAcquisitionSize (
            unsigned int size[2],
            DataSet const & ds)  [static]
```

Get the Acquisition Matrix (non zero value):

12.289.3.6 GetFile() [1/2]

```
File & gdcm::SplitMosaicFilter::GetFile ()  [inline]
```

12.289.3.7 GetFile() [2/2]

```
const File & gdcm::SplitMosaicFilter::GetFile () const  [inline]
```

12.289.3.8 GetImage() [1/2]

```
Image & gdcm::SplitMosaicFilter::GetImage ()  [inline]
```

12.289.3.9 GetImage() [2/2]

```
const Image & gdcm::SplitMosaicFilter::GetImage () const  [inline]
```

12.289.3.10 GetNumberOfImagesInMosaic()

```
unsigned int gdcm::SplitMosaicFilter::GetNumberOfImagesInMosaic (
            File const & file)  [static]
```

Return the value for NumberOfImagesInMosaic, or compute it from Acquisition Size.

12.289.3.11 SetFile()

```
void gdcm::SplitMosaicFilter::SetFile (
            const File & f)  [inline]
```

12.289.3.12 SetImage()

```
void gdcm::SplitMosaicFilter::SetImage (
            const Image & image)
```

12.289.3.13 Split()

```
bool gdcm::SplitMosaicFilter::Split ()
```

Split the SIEMENS MOSAIC image.

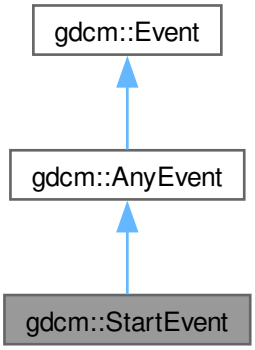
The documentation for this class was generated from the following file:

- gdcmSplitMosaicFilter.h

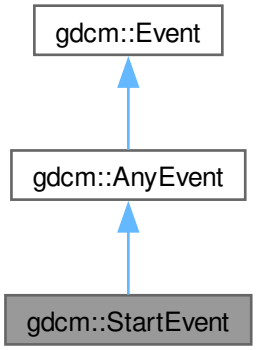
12.290 gdcm::StartEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::StartEvent:



Collaboration diagram for gdcm::StartEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()
- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

- gdcmEvent.h

12.291 gdcm::static_assert_test< x > Struct Template Reference

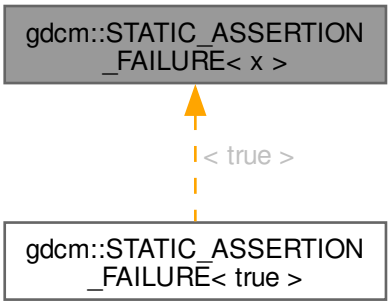
```
#include <gdcmStaticAssert.h>
```

The documentation for this struct was generated from the following file:

- gdcmStaticAssert.h

12.292 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference

Inheritance diagram for gdcm::STATIC_ASSERTION_FAILURE< x >:



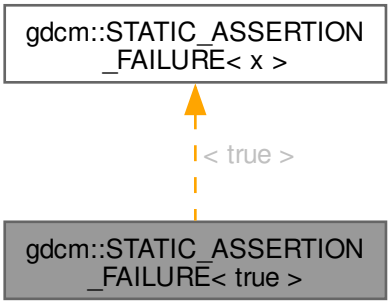
The documentation for this struct was generated from the following file:

- gdcmStaticAssert.h

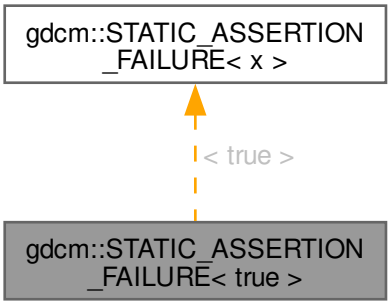
12.293 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Reference

```
#include <gdcmStaticAssert.h>
```

Inheritance diagram for gdcm::STATIC_ASSERTION_FAILURE< true >:



Collaboration diagram for gdcm::STATIC_ASSERTION_FAILURE< true >:



Public Types

- enum { value = 1 }

12.293.1 Member Enumeration Documentation

12.293.1.1 anonymous enum

```
anonymous enum
```

Enumerator

value	

The documentation for this struct was generated from the following file:

- gdcmStaticAssert.h

12.294 gdcm::StreamImageReader Class Reference

StreamImageReader.

```
#include <gdcmStreamImageReader.h>
```

Public Member Functions

- StreamImageReader ()
- virtual ~StreamImageReader ()
- bool CanReadImage () const
- void DefinePixelExtent (uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1)
- uint32_t DefineProperBufferLength () const
- std::vector< unsigned int > GetDimensionsValueForResolution (unsigned int)
- File const & GetFile () const
- bool Read (char *inReadBuffer, const std::size_t &inBufferLength)
- virtual bool ReadImageInformation ()
- void SetFileName (const char *inFileName)
- void SetStream (std::istream &inStream)

12.294.1 Detailed Description

StreamImageReader.

Note

its role is to convert the DICOM DataSet into a Image representation via an ITK streaming (ie, multithreaded) interface Image is different from Pixmap has it has a position and a direction in Space. Currently, this class is thread safe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See also

Image

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

12.294.2 Constructor & Destructor Documentation

12.294.2.1 StreamImageReader()

```
gdcm::StreamImageReader::StreamImageReader ()
```

12.294.2.2 ~StreamImageReader()

```
virtual gdcm::StreamImageReader::~StreamImageReader ()  [virtual]
```

12.294.3 Member Function Documentation

12.294.3.1 CanReadImage()

```
bool gdcm::StreamImageReader::CanReadImage () const
```

Only RAW images are currently readable by the stream reader. As more streaming codecs are added, then this function will be updated to reflect those changes. Calling this function prior to reading will ensure that only streamable files are streamed. Make sure to call ReadImageInformation prior to calling this function.

Examples

StreamImageReaderTest.cxx.

12.294.3.2 DefinePixelExtent()

```
void gdcm::StreamImageReader::DefinePixelExtent (
            uint16_t inXMin,
            uint16_t inXMax,
            uint16_t inYMin,
            uint16_t inYMax,
            uint16_t inZMin = 0,
            uint16_t inZMax = 1)
```

Defines an image extent for the Read function. DICOM states that an image can have no more than 2^16 pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation.

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

12.294.3.3 DefineProperBufferLength()

```
uint32_t gdcm::StreamImageReader::DefineProperBufferLength () const
```

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. Call this function to determine the size of the char∗ buffer that will need to be passed in to ReadImageSubregion(). If the return is 0, then that means that the pixel extent was not defined prior

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

12.294.3.4 GetDimensionsValueForResolution()

```
std::vector< unsigned int > gdcm::StreamImageReader::GetDimensionsValueForResolution (
            unsigned int )
```

12.294.3.5 GetFile()

```
File const & gdcm::StreamImageReader::GetFile () const
```

Returns the dataset read by ReadImageInformation Couple this with the ImageHelper to get statistics about the image, like pixel extent, to be able to initialize buffers for reading

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

12.294.3.6 Read()

```
bool gdcm::StreamImageReader::Read (
            char * inReadBuffer,
            const std::size_t & inBufferLength)
```

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from char∗ to std::ostream (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use ImageReader instead.

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

12.294.3.7 ReadImageInformation()

```
virtual bool gdcm::StreamImageReader::ReadImageInformation () [virtual]
```

Set the spacing and dimension information for the set filename. returns false if the file is not initialized or not an image, with the pixel (7fe0,0010) tag.

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

12.294.3.8 SetFileName()

```
void gdcm::StreamImageReader::SetFileName (
            const char * inFileName)
```

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal Reader class to be able to get non-pixel image information.

Examples

ExtractOneFrame.cs, and StreamImageReaderTest.cxx.

12.294.3.9 SetStream()

```
void gdcm::StreamImageReader::SetStream (
            std::istream & inStream)
```

The documentation for this class was generated from the following file:

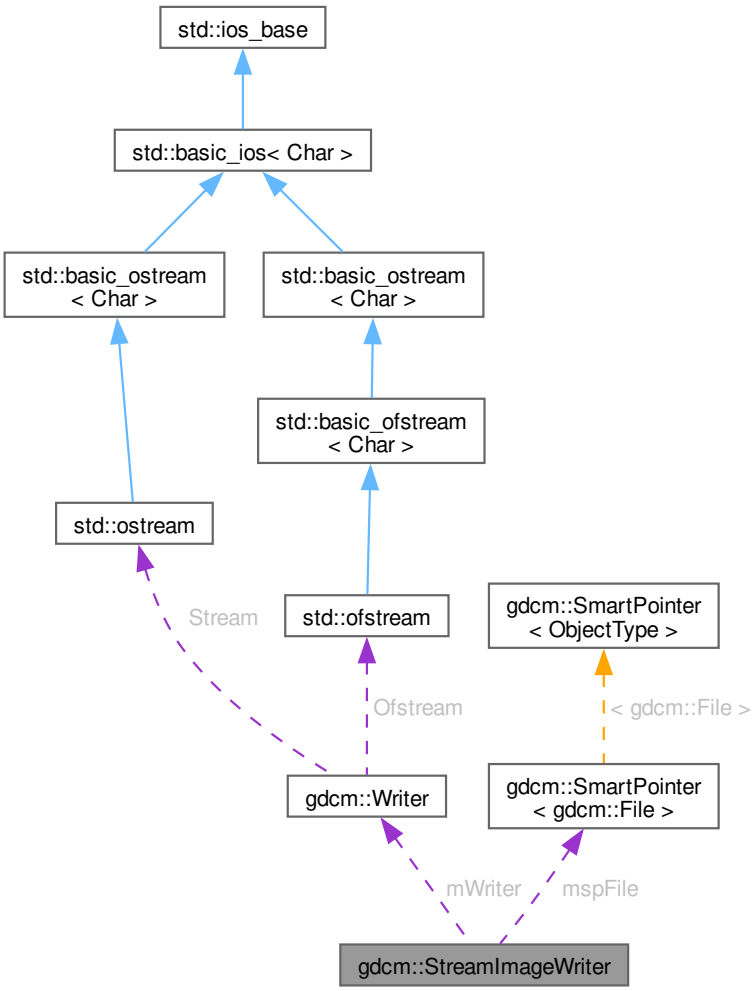
- gdcmStreamImageReader.h

12.295 gdcm::StreamImageWriter Class Reference

StreamImageReader.

```
#include <gdcmStreamImageWriter.h>
```

Collaboration diagram for gdcm::StreamImageWriter:



Public Member Functions

- StreamImageWriter ()
- virtual ~StreamImageWriter ()
- bool CanWriteFile () const
- void DefinePixelExtent (uint16_t inXMin, uint16_t inXMax, uint16_t inYMin, uint16_t inYMax, uint16_t inZMin=0, uint16_t inZMax=1)
- uint32_t DefineProperBufferLength ()
- void SetFile (const File &inFile)
- void SetFileName (const char *inFileName)
- void SetStream (std::ostream &inStream)
- bool Write (void *inWriteBuffer, const std::size_t &inBufferLength)
- virtual bool WriteImageInformation ()

Protected Member Functions

- virtual bool WriteImageSubregionRAW (char ∗inWriteBuffer, const std::size_t &inBufferLength)
- int WriteRawHeader (RAWCodec ∗inCodec, std::ostream ∗inStream)

Protected Attributes

- int mElementOffsets
- int mElementOffsets1
- SmartPointer< File > mspFile
- Writer mWriter
- uint16_t mXMax
- uint16_t mXMin
- uint16_t mYMax
- uint16_t mYMin
- uint16_t mZMax
- uint16_t mZMin

12.295.1 Detailed Description

StreamImageReader.

Note

its role is to convert the DICOM DataSet into a Image representation via an ITK streaming (ie, multithreaded) interface Image is different from Pixmap has it has a position and a direction in Space. Currently, this class is threadsafe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See also

Image

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.295.2 Constructor & Destructor Documentation

12.295.2.1 StreamImageWriter()

```
gdcm::StreamImageWriter::StreamImageWriter ()
```

12.295.2.2 ∼StreamImageWriter()

```
virtual gdcm::StreamImageWriter::~StreamImageWriter () [virtual]
```

12.295.3 Member Function Documentation

12.295.3.1 CanWriteFile()

```
bool gdcm::StreamImageWriter::CanWriteFile () const
```

This function determines if a file can even be written using the streaming writer unlike the reader, can be called before WriteImageInformation, but must be called after SetFile.

Examples

Extracting_All_Resolution.cxx, and Fake_Image_Using_Stream_Image_Writer.cxx.

12.295.3.2 DefinePixelExtent()

```
void gdcm::StreamImageWriter::DefinePixelExtent (
            uint16_t inXMin,
            uint16_t inXMax,
            uint16_t inYMin,
            uint16_t inYMax,
            uint16_t inZMin = 0,
            uint16_t inZMax = 1)
```

Defines an image extent for the Read function. DICOM states that an image can have no more than 2^16 pixels per edge (as of 2009) In this case, the pixel extents ignore the direction cosines entirely, and assumes that the origin of the image is at location 0,0 (regardless of the definition in space per the tags). So, if the first 100 pixels of the first row are to be read in, this function should be called with DefinePixelExtent(0, 100, 0, 1), regardless of pixel size or orientation. 15 nov 2010: added z dimension, defaults to being 1 plane large

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.295.3.3 DefineProperBufferLength()

```
uint32_t gdcm::StreamImageWriter::DefineProperBufferLength ()
```

Paying attention to the pixel format and so forth, define the proper buffer length for the user. The return amount is in bytes. If the return is 0, then that means that the pixel extent was not defined prior this return is for RAW inputs which are then encoded by the writer, but are used to ensure that the writer gets the proper buffer size

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.295.3.4 SetFile()

```
void gdcm::StreamImageWriter::SetFile (
            const File & inFile)
```

Set the image information to be written to disk that is everything but the pixel information: (7fe0,0010) PixelData

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.295.3.5 SetFileName()

```
void gdcm::StreamImageWriter::SetFileName (
            const char * inFileName)
```

One of either SetFileName or SetStream must be called prior to any other functions. These initialize an internal Reader class to be able to get non-pixel image information.

12.295.3.6 SetStream()

```
void gdcm::StreamImageWriter::SetStream (
            std::ostream & inStream)
```

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.295.3.7 Write()

```
bool gdcm::StreamImageWriter::Write (
            void * inWriteBuffer,
            const std::size_t & inBufferLength)
```

Read the DICOM image. There are three reasons for failure:

1. The extent is not set
2. the conversion from void∗ to std::ostream (internally) fails
3. the given buffer isn't large enough to accommodate the desired pixel extent. This method has been implemented to look similar to the metaimageio in itk MUST have an extent defined, or else Read will return false. If no particular extent is required, use ImageReader instead.

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.295.3.8 WriteImageInformation()

```
virtual bool gdcm::StreamImageWriter::WriteImageInformation ()  [virtual]
```

Write the header information to disk, and a bunch of zeros for the actual pixel information Of course, if we're doing a non-compressed format, that works but if it's compressed, we have to force the ordering of chunks that are written.

Examples

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.295.3.9 WriteImageSubregionRAW()

```
virtual bool gdcm::StreamImageWriter::WriteImageSubregionRAW (
            char * inWriteBuffer,
            const std::size_t & inBufferLength)  [protected], [virtual]
```

Using the min, max, etc set by DefinePixelExtent, this will fill the given buffer Make sure to call DefinePixelExtent and to initialize the buffer with the amount given by DefineProperBufferLength prior to calling this. reads by the RAW codec; other codecs are added once implemented

12.295.3.10 WriteRawHeader()

```
int gdcm::StreamImageWriter::WriteRawHeader (
            RAWCodec * inCodec,
            std::ostream * inStream)  [protected]
```

when writing a raw file, we know the full extent, and can just write the first 12 bytes out (the tag, the VR, and the size) when we do compressed files, we'll do it in chunks, as described in 2009-3, part 5, Annex A, section 4. Pass the raw codec so that in the rare case of a bigendian explicit raw, the first 12 bytes written out should still be kosher. returns -1 if there's any failure, or the complete offset (12 bytes) if it works. Those 12 bytes are then added to the position in order to determine where to write.

12.295.4 Member Data Documentation

12.295.4.1 mElementOffsets

```
int gdcm::StreamImageWriter::mElementOffsets  [protected]
```

The result of WriteRawHeader (or another header, when that's implemented) This result is saved so that the first N bytes aren't constantly being rewritten for each chunk that's passed in. For compressed data, the offset table will require rewrites of data.

12.295.4.2 mElementOffsets1

```
int gdcm::StreamImageWriter::mElementOffsets1  [protected]
```

12.295.4.3 mspFile

```
SmartPointer<File> gdcm::StreamImageWriter::mspFile [protected]
```

12.295.4.4 mWriter

```
Writer gdcm::StreamImageWriter::mWriter [protected]
```

12.295.4.5 mXMax

```
uint16_t gdcm::StreamImageWriter::mXMax [protected]
```

12.295.4.6 mXMin

```
uint16_t gdcm::StreamImageWriter::mXMin [protected]
```

12.295.4.7 mYMax

```
uint16_t gdcm::StreamImageWriter::mYMax [protected]
```

12.295.4.8 mYMin

```
uint16_t gdcm::StreamImageWriter::mYMin [protected]
```

12.295.4.9 mZMax

```
uint16_t gdcm::StreamImageWriter::mZMax [protected]
```

12.295.4.10 mZMin

```
uint16_t gdcm::StreamImageWriter::mZMin [protected]
```

The documentation for this class was generated from the following file:

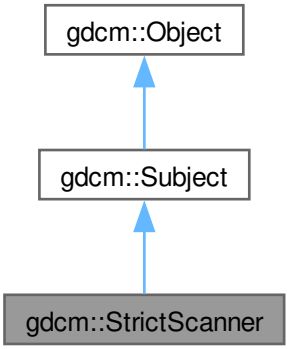
- gdcmStreamImageWriter.h

12.296 gdcm::StrictScanner Class Reference

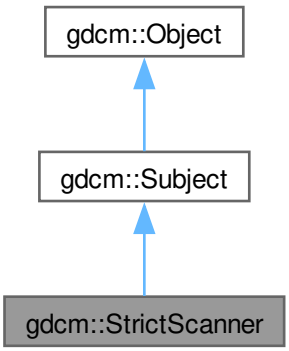
StrictScanner.

```
#include <gdcmStrictScanner.h>
```

Inheritance diagram for gdcm::StrictScanner:



Collaboration diagram for gdcm::StrictScanner:



Classes

- struct ltstr

Public Types

- typedef MappingType::const_iterator ConstIterator
- typedef std::map< const char *, TagToValue, ltstr > MappingType
- typedef std::map< Tag, const char * > TagToValue
- typedef TagToValue::value_type TagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- StrictScanner ()
- ~StrictScanner () override
- void AddPrivateTag (PrivateTag const &t)
- void AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- void AddTag (Tag const &t)

  Add a tag that will need to be read. Those are root level skip tags.
- ConstIterator Begin () const
- void ClearSkipTags ()
- void ClearTags ()
- ConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const
- Directory::FilenamesType GetKeys () const
- TagToValue const & GetMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- TagToValue const & GetMappingFromTagToValue (Tag const &t, const char *value) const

  See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.
- MappingType const & GetMappings () const

  Mappings are the mapping from a particular tag to the map, mapping filename to value:
- Directory::FilenamesType GetOrderedValues (Tag const &t) const
- const char * GetValue (const char *filename, Tag const &t) const
- ValuesType const & GetValues () const

  Get all the values found (in lexicographic order).
- ValuesType GetValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const override

  Print result.
- void PrintTable (std::ostream &os) const
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Static Public Member Functions

- static SmartPointer< StrictScanner > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- void ProcessPublicTag (StringFilter &sf, const char *filename)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Friends

- std::ostream & operator<< (std::ostream &_os, const StrictScanner &s)

12.296.1 Detailed Description

StrictScanner.

This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

12.296.2 Member Typedef Documentation

12.296.2.1 ConstIterator

```
typedef MappingType::const_iterator gdcm::StrictScanner::ConstIterator
```

12.296.2.2 MappingType

```
typedef std::map<const char *,TagToValue, ltstr> gdcm::StrictScanner::MappingType
```

12.296.2.3 TagToValue

```
typedef std::map<Tag, const char*> gdcm::StrictScanner::TagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char * and not std::string since we are pointing to existing std::string (hold in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

Examples

SimpleScanner.cxx.

12.296.2.4 TagToValueValueType

```
typedef TagToValue::value_type gdcm::StrictScanner::TagToValueValueType
```

12.296.2.5 ValuesType

```
typedef std::set< std::string > gdcm::StrictScanner::ValuesType
```

12.296.3 Constructor & Destructor Documentation

12.296.3.1 StrictScanner()

```
gdcm::StrictScanner::StrictScanner () [inline]
```

Referenced by New(), and operator<<.

12.296.3.2 ~StrictScanner()

```
gdcm::StrictScanner::~StrictScanner () [override]
```

12.296.4 Member Function Documentation

12.296.4.1 AddPrivateTag()

```
void gdcm::StrictScanner::AddPrivateTag (
            PrivateTag const & t)
```

12.296.4.2 AddSkipTag()

```
void gdcm::StrictScanner::AddSkipTag (
            Tag const & t)
```

Add a tag that will need to be skipped. Those are root level skip tags.

12.296.4.3 AddTag()

```
void gdcm::StrictScanner::AddTag (
            Tag const & t)
```

Add a tag that will need to be read. Those are root level skip tags.

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

12.296.4.4 Begin()

```
ConstIterator gdcm::StrictScanner::Begin () const  [inline]
```

12.296.4.5 ClearSkipTags()

```
void gdcm::StrictScanner::ClearSkipTags ()
```

12.296.4.6 ClearTags()

```
void gdcm::StrictScanner::ClearTags ()
```

12.296.4.7 End()

```
ConstIterator gdcm::StrictScanner::End () const  [inline]
```

12.296.4.8 GetAllFilenamesFromTagToValue()

```
Directory::FilenamesType gdcm::StrictScanner::GetAllFilenamesFromTagToValue (
            Tag const & t,
            const char * valueref) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

12.296.4.9 GetFilenameFromTagToValue()

```
const char * gdcm::StrictScanner::GetFilenameFromTagToValue (
            Tag const & t,
            const char * valueref) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

12.296.4.10 GetFilenames()

```
Directory::FilenamesType const & gdcm::StrictScanner::GetFilenames () const  [inline]
```

12.296.4.11 GetKeys()

```
Directory::FilenamesType gdcm::StrictScanner::GetKeys () const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

12.296.4.12 GetMapping()

```
TagToValue const & gdcm::StrictScanner::GetMapping (
            const char * filename) const
```

Get the std::map mapping filenames to value for file 'filename'.

Examples

SimpleScanner.cxx.

12.296.4.13 GetMappingFromTagToValue()

```
TagToValue const & gdcm::StrictScanner::GetMappingFromTagToValue (
            Tag const & t,
            const char * value) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

12.296.4.14 GetMappings()

```
MappingType const & gdcm::StrictScanner::GetMappings () const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

12.296.4.15 GetOrderedValues()

```
Directory::FilenamesType gdcm::StrictScanner::GetOrderedValues (
            Tag const & t) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

12.296.4.16 GetValue()

```
const char * gdcm::StrictScanner::GetValue (
            const char * filename,
            Tag const & t) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

12.296.4.17 GetValues() [1/2]

```
ValuesType const & gdcm::StrictScanner::GetValues () const  [inline]
```

Get all the values found (in lexicographic order).

12.296.4.18 GetValues() [2/2]

```
ValuesType gdcm::StrictScanner::GetValues (
            Tag const & t) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

12.296.4.19 IsKey()

```
bool gdcm::StrictScanner::IsKey (
            const char * filename) const
```

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

12.296.4.20 New()

```
SmartPointer< StrictScanner > gdcm::StrictScanner::New () [inline], [static]
```

for wrapped language: instantiate a reference counted object

Examples

ScanDirectory.cs.

References StrictScanner().

12.296.4.21 Print()

```
void gdcm::StrictScanner::Print (
            std::ostream & os) const [override], [virtual]
```

Print result.

Reimplemented from gdcm::Object.

Referenced by operator<<.

12.296.4.22 PrintTable()

```
void gdcm::StrictScanner::PrintTable (
            std::ostream & os) const
```

12.296.4.23 ProcessPublicTag()

```
void gdcm::StrictScanner::ProcessPublicTag (
            StringFilter & sf,
            const char * filename) [protected]
```

12.296.4.24 Scan()

```
bool gdcm::StrictScanner::Scan (
            Directory::FilenamesType const & filenames)
```

Start the scan !

Examples

ScanDirectory.cs, and SimpleScanner.cxx.

12.296.5 Friends And Related Symbol Documentation

12.296.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const StrictScanner & s)  [friend]
```

References StrictScanner(), and Print().

The documentation for this class was generated from the following file:

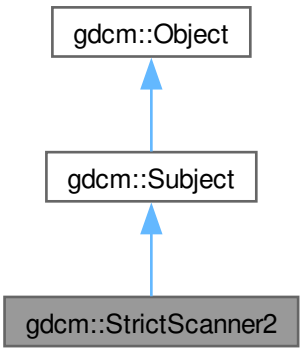
- gdcmStrictScanner.h

12.297 gdcm::StrictScanner2 Class Reference

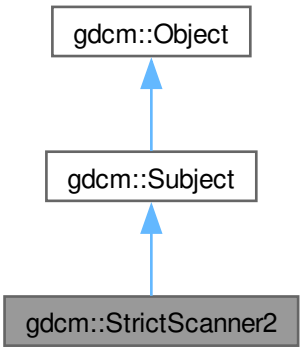
StrictScanner2.

```
#include <gdcmStrictScanner2.h>
```

Inheritance diagram for gdcm::StrictScanner2:



Collaboration diagram for gdcm::StrictScanner2:



Classes

- struct ltstr

Public Types

- typedef PrivateMappingType::const_iterator PrivateConstIterator
- typedef std::map< const char *, PrivateTagToValue, ltstr > PrivateMappingType
- typedef std::map< PrivateTag, const char * > PrivateTagToValue
- typedef PrivateTagToValue::value_type PrivateTagToValueValueType
- typedef PublicMappingType::const_iterator PublicConstIterator
- typedef std::map< const char *, PublicTagToValue, ltstr > PublicMappingType
- typedef std::map< Tag, const char * > PublicTagToValue
- typedef PublicTagToValue::value_type PublicTagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- StrictScanner2 ()
- ~StrictScanner2 () override
- bool AddPrivateTag (PrivateTag const &pt)
- bool AddPublicTag (Tag const &t)

  Add a tag that will need to be read. Those are root level tags.
- bool AddSkipTag (Tag const &t)

  Add a tag that will need to be skipped. Those are root level skip tags.
- PublicConstIterator Begin () const
- void ClearPrivateTags ()
- void ClearPublicTags ()

- void ClearSkipTags ()
- PublicConstIterator End () const
- Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue (PrivateTag const &pt, const char *valueref) const
- Directory::FilenamesType GetAllFilenamesFromPublicTagToValue (Tag const &t, const char *valueref) const
- const char * GetFilenameFromPrivateTagToValue (PrivateTag const &pt, const char *valueref) const
- const char * GetFilenameFromPublicTagToValue (Tag const &t, const char *valueref) const
- Directory::FilenamesType const & GetFilenames () const

  Return the list of filenames.
- Directory::FilenamesType GetKeys () const
- PrivateTagToValue const & GetMappingFromPrivateTagToValue (PrivateTag const &pt, const char *value) const
- PublicTagToValue const & GetMappingFromPublicTagToValue (Tag const &t, const char *value) const
- PrivateTagToValue const & GetPrivateMapping (const char *filename) const
- PrivateMappingType const & GetPrivateMappings () const
- Directory::FilenamesType GetPrivateOrderedValues (PrivateTag const &pt) const
- const char * GetPrivateValue (const char *filename, PrivateTag const &t) const
- ValuesType GetPrivateValues (PrivateTag const &pt) const
- PublicTagToValue const & GetPublicMapping (const char *filename) const

  Get the std::map mapping filenames to value for file 'filename'.
- PublicMappingType const & GetPublicMappings () const
- Directory::FilenamesType GetPublicOrderedValues (Tag const &t) const
- const char * GetPublicValue (const char *filename, Tag const &t) const
- ValuesType GetPublicValues (Tag const &t) const

  Get all the values found (in lexicographic order) associated with Tag 't'.
- ValuesType const & GetValues () const

  Get all the values found (in lexicographic order).
- bool IsKey (const char *filename) const
- void Print (std::ostream &os) const override

  Print result.
- void PrintTable (std::ostream &os, bool header=false) const

  Print result as CSV table.
- PrivateConstIterator PrivateBegin () const
- PrivateConstIterator PrivateEnd () const
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Static Public Member Functions

- static SmartPointer< StrictScanner2 > New ()

  for wrapped language: instantiate a reference counted object

Protected Member Functions

- void ProcessPrivateTag (StringFilter &sf, const char *filename)
- void ProcessPublicTag (StringFilter &sf, const char *filename)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Friends

- std::ostream & operator<< (std::ostream &_os, const StrictScanner2 &s)

12.297.1 Detailed Description

StrictScanner2.

This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of StringFilter

Warning

IMPORTANT In case of file where tags are not ordered (illegal as per DICOM specification), the output will be missing information

Note

implementation details. All values are stored in a std::set of std::string. Then the address of the cstring underlying the std::string is used in the std::map.

This class implement the Subject/Observer pattern trigger the following events:

- ProgressEvent
- StartEvent
- EndEvent

12.297.2 Member Typedef Documentation

12.297.2.1 PrivateConstIterator

```
typedef PrivateMappingType::const_iterator gdcm::StrictScanner2::PrivateConstIterator
```

12.297.2.2 PrivateMappingType

```
typedef std::map<const char *, PrivateTagToValue, ltstr> gdcm::StrictScanner2::PrivateMappingType
```

12.297.2.3 PrivateTagToValue

```
typedef std::map<PrivateTag, const char *> gdcm::StrictScanner2::PrivateTagToValue
```

12.297.2.4 PrivateTagToValueValueType

```
typedef PrivateTagToValue::value_type gdcm::StrictScanner2::PrivateTagToValueValueType
```

12.297.2.5 PublicConstIterator

```
typedef PublicMappingType::const_iterator gdcm::StrictScanner2::PublicConstIterator
```

12.297.2.6 PublicMappingType

```
typedef std::map<const char *, PublicTagToValue, ltstr> gdcm::StrictScanner2::PublicMappingType
```

12.297.2.7 PublicTagToValue

```
typedef std::map<Tag, const char *> gdcm::StrictScanner2::PublicTagToValue
```

struct to map a filename to a value Implementation note: all std::map in this class will be using const char $*$ and not std::string since we are pointing to existing std::string (held in a std::vector) this avoid an extra copy of the byte array. Tag are used as Tag class since sizeof(tag) <= sizeof(pointer)

12.297.2.8 PublicTagToValueValueType

```
typedef PublicTagToValue::value_type gdcm::StrictScanner2::PublicTagToValueValueType
```

12.297.2.9 ValuesType

```
typedef std::set<std::string> gdcm::StrictScanner2::ValuesType
```

12.297.3 Constructor & Destructor Documentation

12.297.3.1 StrictScanner2()

```
gdcm::StrictScanner2::StrictScanner2 () [inline]
```

Referenced by New(), and operator<<.

12.297.3.2 ~StrictScanner2()

```
gdcm::StrictScanner2::~StrictScanner2 () [override]
```

12.297.4 Member Function Documentation

12.297.4.1 AddPrivateTag()

```
bool gdcm::StrictScanner2::AddPrivateTag (
            PrivateTag const & pt)
```

12.297.4.2 AddPublicTag()

```
bool gdcm::StrictScanner2::AddPublicTag (
            Tag const & t)
```

Add a tag that will need to be read. Those are root level tags.

12.297.4.3 AddSkipTag()

```
bool gdcm::StrictScanner2::AddSkipTag (
            Tag const & t)
```

Add a tag that will need to be skipped. Those are root level skip tags.

12.297.4.4 Begin()

```
PublicConstIterator gdcm::StrictScanner2::Begin () const [inline]
```

12.297.4.5 ClearPrivateTags()

```
void gdcm::StrictScanner2::ClearPrivateTags ()
```

12.297.4.6 ClearPublicTags()

```
void gdcm::StrictScanner2::ClearPublicTags ()
```

12.297.4.7 ClearSkipTags()

```
void gdcm::StrictScanner2::ClearSkipTags ()
```

12.297.4.8 End()

```
PublicConstIterator gdcm::StrictScanner2::End () const  [inline]
```

12.297.4.9 GetAllFilenamesFromPrivateTagToValue()

```
Directory::FilenamesType gdcm::StrictScanner2::GetAllFilenamesFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * valueref) const
```

12.297.4.10 GetAllFilenamesFromPublicTagToValue()

```
Directory::FilenamesType gdcm::StrictScanner2::GetAllFilenamesFromPublicTagToValue (
            Tag const & t,
            const char * valueref) const
```

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valueref'

12.297.4.11 GetFilenameFromPrivateTagToValue()

```
const char * gdcm::StrictScanner2::GetFilenameFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * valueref) const
```

12.297.4.12 GetFilenameFromPublicTagToValue()

```
const char * gdcm::StrictScanner2::GetFilenameFromPublicTagToValue (
            Tag const & t,
            const char * valueref) const
```

Will loop over all files and return the first file where value match the reference value 'valueref'

12.297.4.13 GetFilenames()

```
Directory::FilenamesType const & gdcm::StrictScanner2::GetFilenames () const  [inline]
```

Return the list of filenames.

12.297.4.14 GetKeys()

```
Directory::FilenamesType gdcm::StrictScanner2::GetKeys () const
```

Return the list of filename that are key in the internal map, which means those filename were properly parsed

12.297.4.15 GetMappingFromPrivateTagToValue()

```
PrivateTagToValue const & gdcm::StrictScanner2::GetMappingFromPrivateTagToValue (
            PrivateTag const & pt,
            const char * value) const
```

12.297.4.16 GetMappingFromPublicTagToValue()

```
PublicTagToValue const & gdcm::StrictScanner2::GetMappingFromPublicTagToValue (
            Tag const & t,
            const char * value) const
```

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed

12.297.4.17 GetPrivateMapping()

```
PrivateTagToValue const & gdcm::StrictScanner2::GetPrivateMapping (
            const char * filename) const
```

12.297.4.18 GetPrivateMappings()

```
PrivateMappingType const & gdcm::StrictScanner2::GetPrivateMappings () const  [inline]
```

12.297.4.19 GetPrivateOrderedValues()

```
Directory::FilenamesType gdcm::StrictScanner2::GetPrivateOrderedValues (
            PrivateTag const & pt) const
```

12.297.4.20 GetPrivateValue()

```
const char * gdcm::StrictScanner2::GetPrivateValue (
            const char * filename,
            PrivateTag const & t) const
```

12.297.4.21 GetPrivateValues()

```
ValuesType gdcm::StrictScanner2::GetPrivateValues (
            PrivateTag const & pt) const
```

Get all the values found (in lexicographic order) associated with PrivateTag 'pt'

12.297.4.22 GetPublicMapping()

```
PublicTagToValue const & gdcm::StrictScanner2::GetPublicMapping (
            const char * filename) const
```

Get the std::map mapping filenames to value for file 'filename'.

12.297.4.23 GetPublicMappings()

```
PublicMappingType const & gdcm::StrictScanner2::GetPublicMappings () const  [inline]
```

Mappings are the mapping from a particular tag to the map, mapping filename to value:

12.297.4.24 GetPublicOrderedValues()

```
Directory::FilenamesType gdcm::StrictScanner2::GetPublicOrderedValues (
            Tag const & t) const
```

Get all the values found (in a vector) associated with Tag 't' This function is identical to GetValues, but is accessible from the wrapped layer (python, C#, java)

12.297.4.25 GetPublicValue()

```
const char * gdcm::StrictScanner2::GetPublicValue (
            const char * filename,
            Tag const & t) const
```

Retrieve the value found for tag: t associated with file: filename This is meant for a single short call. If multiple calls (multiple tags) should be done, prefer the GetMapping function, and then reuse the TagToValue hash table.

Warning

Tag 't' should have been added via AddTag() prior to the Scan() call !

12.297.4.26 GetPublicValues()

```
ValuesType gdcm::StrictScanner2::GetPublicValues (
            Tag const & t) const
```

Get all the values found (in lexicographic order) associated with Tag 't'.

12.297.4.27 GetValues()

```
ValuesType const & gdcm::StrictScanner2::GetValues () const  [inline]
```

Get all the values found (in lexicographic order).

12.297.4.28 IsKey()

```
bool gdcm::StrictScanner2::IsKey (
            const char * filename) const
```

Check if filename is a key in the Mapping table. returns true only of file can be found, which means the file was indeed a DICOM file that could be processed

12.297.4.29 New()

```
SmartPointer< StrictScanner2 > gdcm::StrictScanner2::New ()  [inline], [static]
```

for wrapped language: instantiate a reference counted object

References StrictScanner2().

12.297.4.30 Print()

```
void gdcm::StrictScanner2::Print (
            std::ostream & os) const  [override], [virtual]
```

Print result.

Reimplemented from gdcm::Object.

Referenced by operator<<.

12.297.4.31 PrintTable()

```
void gdcm::StrictScanner2::PrintTable (
            std::ostream & os,
            bool header = false) const
```

Print result as CSV table.

12.297.4.32 PrivateBegin()

```
PrivateConstIterator gdcm::StrictScanner2::PrivateBegin () const  [inline]
```

12.297.4.33 PrivateEnd()

```
PrivateConstIterator gdcm::StrictScanner2::PrivateEnd () const  [inline]
```

12.297.4.34 ProcessPrivateTag()

```
void gdcm::StrictScanner2::ProcessPrivateTag (
            StringFilter & sf,
            const char * filename)  [protected]
```

12.297.4.35 ProcessPublicTag()

```
void gdcm::StrictScanner2::ProcessPublicTag (
            StringFilter & sf,
            const char * filename)  [protected]
```

12.297.4.36 Scan()

```
bool gdcm::StrictScanner2::Scan (
            Directory::FilenamesType const & filenames)
```

Start the scan !

12.297.5 Friends And Related Symbol Documentation

12.297.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const StrictScanner2 & s)  [friend]
```

References StrictScanner2(), and Print().

The documentation for this class was generated from the following file:

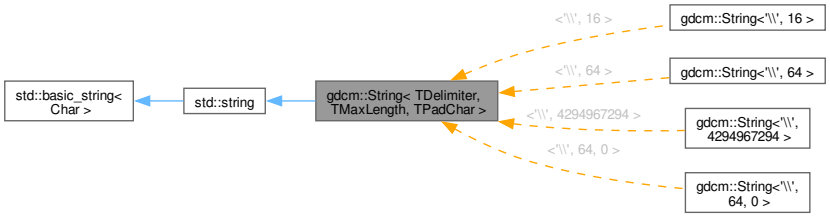
- gdcmStrictScanner2.h

12.298 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

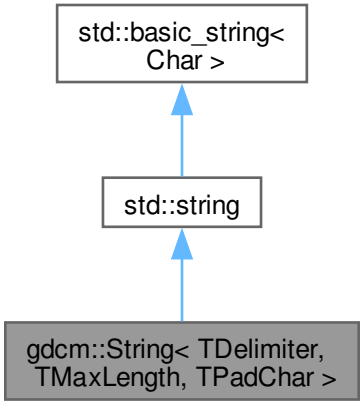
String.

```
#include <gdcmString.h>
```

Inheritance diagram for gdcm::String< TDelimiter, TMaxLength, TPadChar >:



Collaboration diagram for gdcm::String< TDelimiter, TMaxLength, TPadChar >:



Public Types

- typedef std::string::const_iterator const_iterator
- typedef std::string::const_reference const_reference
- typedef std::string::const_reverse_iterator const_reverse_iterator
- typedef std::string::difference_type difference_type

- typedef std::string::iterator iterator
- typedef std::string::pointer pointer
- typedef std::string::reference reference
- typedef std::string::reverse_iterator reverse_iterator
- typedef std::string::size_type size_type
- typedef std::string::value_type value_type

Public Member Functions

- String ()

  String constructors.
- String (const std::string &s, size_type pos=0, size_type n=npos)
- String (const value_type *s)
- String (const value_type *s, size_type n)
- bool IsValid () const

  return if string is valid
- operator const char * () const

  WARNING: Trailing \0 might be lost in this operation:
- std::string Trim () const
- gdcm::String< TDelimiter, TMaxLength, TPadChar > Truncate () const

Static Public Member Functions

- static std::string Trim (const char *input)

12.298.1 Detailed Description

template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
class gdcm::String< TDelimiter, TMaxLength, TPadChar >

String.

Note

TDelimiter template parameter is used to separate multiple String (VM1 >) TMaxLength is only a hint. No one actually respect the max length TPadChar is the string padding (0 or space)

12.298.2 Member Typedef Documentation

12.298.2.1 const_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↩
iterator
```

12.298.2.2 const_reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_↩
reference
```

12.298.2.3 const_reverse_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::const_reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >↩
::const_reverse_iterator
```

12.298.2.4 difference_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::difference_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::difference↩
_type
```

12.298.2.5 iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator
```

12.298.2.6 pointer

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::pointer gdcm::String< TDelimiter, TMaxLength, TPadChar >::pointer
```

12.298.2.7 reference

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::reference
```

12.298.2.8 reverse_iterator

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::reverse↩
_iterator
```

12.298.2.9 size_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::size_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::size_type
```

12.298.2.10 value_type

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type
```

12.298.3 Constructor & Destructor Documentation

12.298.3.1 String() [1/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ()  [inline]
```

String constructors.

12.298.3.2 String() [2/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
            const value_type * s)  [inline]
```

12.298.3.3 String() [3/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
            const value_type * s,
            size_type n)  [inline]
```

12.298.3.4 String() [4/4]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (
            const std::string & s,
            size_type pos = 0,
            size_type n = npos)  [inline]
```

12.298.4 Member Function Documentation

12.298.4.1 IsValid()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
bool gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid () const  [inline]
```

return if string is valid

Referenced by gdcm::String<'\\', 16 >::Truncate().

12.298.4.2 operator const char *()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar >::operator const char * () const  [inline]
```

WARNING: Trailing \0 might be lost in this operation:

12.298.4.3 Trim() [1/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim () const  [inline]
```

Trim function is required to return a std::string object, otherwise we could not create a gdcm::String object with an odd number of bytes...

12.298.4.4 Trim() [2/2]

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim (
            const char * input)  [inline], [static]
```

12.298.4.5 Truncate()

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
gdcm::String< TDelimiter, TMaxLength, TPadChar > gdcm::String< TDelimiter, TMaxLength, TPadChar
>::Truncate () const  [inline]
```

The documentation for this class was generated from the following file:

- gdcmString.h

12.299 gdcm::StringFilter Class Reference

StringFilter.

```
#include <gdcmStringFilter.h>
```

Public Member Functions

- StringFilter ()
- ~StringFilter ()
- bool ExecuteQuery (std::string const &query, std::string &value) const
- std::string FromString (const Tag &t, const char *value, size_t len)

  Convert to string the char array defined by the pair (value,len).
- File & GetFile ()
- const File & GetFile () const
- void SetDicts (const Dicts &dicts)

  Allow user to pass in there own dicts.
- void SetFile (const File &f)

  Set/Get File.
- std::string ToString (const DataElement &de) const
- std::string ToString (const PrivateTag &t) const
- std::string ToString (const Tag &t) const

  Directly from a Tag:
- std::pair< std::string, std::string > ToStringPair (const DataElement &de) const
- std::pair< std::string, std::string > ToStringPair (const Tag &t) const

  Directly from a Tag:
- void UseDictAlways (bool)

Protected Member Functions

- bool ExecuteQuery (std::string const &query, DataSet const &ds, std::string &value) const
- std::pair< std::string, std::string > ToStringPair (const Tag &t, DataSet const &ds) const

12.299.1 Detailed Description

StringFilter.

StringFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language

Examples

DumpVisusChange.cxx, ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

12.299.2 Constructor & Destructor Documentation

12.299.2.1 StringFilter()

```
gdcm::StringFilter::StringFilter ()
```

12.299.2.2 ~StringFilter()

```
gdcm::StringFilter::~StringFilter ()
```

12.299.3 Member Function Documentation

12.299.3.1 ExecuteQuery() [1/2]

```
bool gdcm::StringFilter::ExecuteQuery (
            std::string const & query,
            DataSet const & ds,
            std::string & value) const  [protected]
```

12.299.3.2 ExecuteQuery() [2/2]

```
bool gdcm::StringFilter::ExecuteQuery (
            std::string const & query,
            std::string & value) const
```

Execute the XPATH query to find a value (as string) return false when attribute is not found (or an error in the XPATH query) You need to make sure that your XPATH query is syntactically correct

12.299.3.3 FromString()

```
std::string gdcm::StringFilter::FromString (
            const Tag & t,
            const char * value,
            size_t len)
```

Convert to string the char array defined by the pair (value,len).

12.299.3.4 GetFile() [1/2]

```
File & gdcm::StringFilter::GetFile ()  [inline]
```

12.299.3.5 GetFile() [2/2]

```
const File & gdcm::StringFilter::GetFile () const  [inline]
```

12.299.3.6 SetDicts()

```
void gdcm::StringFilter::SetDicts (
            const Dicts & dicts)
```

Allow user to pass in there own dicts.

12.299.3.7 SetFile()

```
void gdcm::StringFilter::SetFile (
            const File & f)  [inline]
```

Set/Get File.

Examples

DumpVisusChange.cxx, ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

12.299.3.8 ToString() [1/3]

```
std::string gdcm::StringFilter::ToString (
            const DataElement & de) const
```

Convert to string the ByteValue contained in a DataElement. The DataElement must be coming from the actual DataSet associated with File (see SetFile).

Examples

DumpVisusChange.cxx, ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

12.299.3.9 ToString() [2/3]

```
std::string gdcm::StringFilter::ToString (
            const PrivateTag & t) const
```

12.299.3.10 ToString() [3/3]

```
std::string gdcm::StringFilter::ToString (
            const Tag & t) const
```

Directly from a Tag:

12.299.3.11 ToStringPair() [1/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
            const DataElement & de) const
```

Convert to string the ByteValue contained in a DataElement the returned elements are: pair.first : the name as found in the dictionary of DataElement pari.second : the value encoded into a string (US,UL...) are properly converted

Examples

ReadAndPrintAttributes.cxx.

12.299.3.12 ToStringPair() [2/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
            const Tag & t) const
```

Directly from a Tag:

12.299.3.13 ToStringPair() [3/3]

```
std::pair< std::string, std::string > gdcm::StringFilter::ToStringPair (
            const Tag & t,
            DataSet const & ds) const  [protected]
```

12.299.3.14 UseDictAlways()

```
void gdcm::StringFilter::UseDictAlways (
            bool )  [inline]
```

The documentation for this class was generated from the following file:

- gdcmStringFilter.h

12.300 gdcm::Study Class Reference

Study.

```
#include <gdcmStudy.h>
```

Public Member Functions

- Study ()=default

12.300.1 Detailed Description

Study.

12.300.2 Constructor & Destructor Documentation

12.300.2.1 Study()

```
gdcm::Study::Study ()  [default]
```

The documentation for this class was generated from the following file:

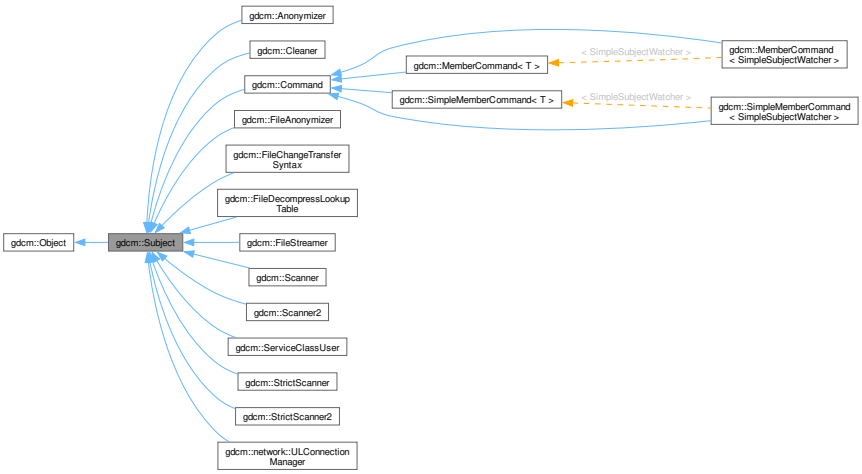
- gdcmStudy.h

12.301 gdcm::Subject Class Reference

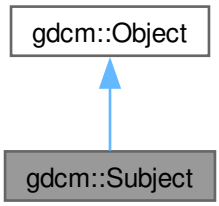
Subject.

```
#include <gdcmSubject.h>
```

Inheritance diagram for gdcm::Subject:



Collaboration diagram for gdcm::Subject:



Public Member Functions

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)

- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.301.1 Detailed Description

Subject.

See also

Command Event

Examples

BasicAnonymizer.cs, Cleaner.cs, ClinicalTrialIdentificationWorkflow.cs, ScanDirectory.cs, and SimpleScanner.cxx.

12.301.2 Constructor & Destructor Documentation

12.301.2.1 Subject()

```
gdcm::Subject::Subject ()
```

Referenced by gdcm::Command::Execute(), and gdcm::Command::Execute().

12.301.2.2 ~Subject()

```
gdcm::Subject::~Subject () [override]
```

12.301.3 Member Function Documentation

12.301.3.1 AddObserver() [1/2]

```
unsigned long gdcm::Subject::AddObserver (
            const Event & event,
            Command * )
```

Allow people to add/remove/invoke observers (callbacks) to any GDCM object. This is an implementation of the subject/observer design pattern. An observer is added by specifying an event to respond to and an gdcm::Command to execute. It returns an unsigned long tag which can be used later to remove the event or retrieve the command. The memory for the Command becomes the responsibility of this object, so don't pass the same instance of a command to two different objects

12.301.3.2 AddObserver() [2/2]

```
unsigned long gdcm::Subject::AddObserver (
            const Event & event,
            Command * ) const
```

12.301.3.3 GetCommand()

```
Command * gdcm::Subject::GetCommand (
            unsigned long tag)
```

Get the command associated with the given tag. NOTE: This returns a pointer to a Command, but it is safe to assign this to a Command::Pointer. Since Command inherits from LightObject, at this point in the code, only a pointer or a reference to the Command can be used.

12.301.3.4 HasObserver()

```
bool gdcm::Subject::HasObserver (
            const Event & event) const
```

Return true if an observer is registered for this event.

12.301.3.5 InvokeEvent() [1/2]

```
void gdcm::Subject::InvokeEvent (
            const Event & )
```

Call Execute on all the Commands observing this event id.

12.301.3.6 InvokeEvent() [2/2]

```
void gdcm::Subject::InvokeEvent (
            const Event & ) const
```

Call Execute on all the Commands observing this event id. The actions triggered by this call doesn't modify this object.

12.301.3.7 RemoveAllObservers()

```
void gdcm::Subject::RemoveAllObservers ()
```

Remove all observers .

12.301.3.8 RemoveObserver()

```
void gdcm::Subject::RemoveObserver (
            unsigned long tag)
```

Remove the observer with this tag value.

The documentation for this class was generated from the following file:

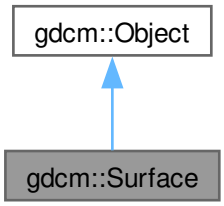
- gdcmSubject.h

12.302 gdcm::Surface Class Reference

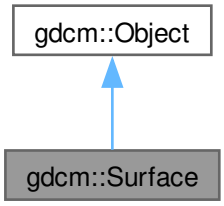
This class defines a SURFACE IE.

```
#include <gdcmSurface.h>
```

Inheritance diagram for gdcm::Surface:



Collaboration diagram for gdcm::Surface:



Public Types

- enum STATES {
  NO = 0 ,
  YES ,
  UNKNOWN ,
  STATES_END }
- enum VIEWType {
  SURFACE = 0 ,
  WIREFRAME ,
  POINTS ,
  VIEWType_END }

  Enumeration for Recommended Presentation Type.

Public Member Functions

- Surface ()
- ~Surface () override
- SegmentHelper::BasicCodedEntry & GetAlgorithmFamily ()
- SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily () const
- const char * GetAlgorithmName () const
- const char * GetAlgorithmVersion () const
- const float * GetAxisOfRotation () const
- const float * GetCenterOfRotation () const
- STATES GetFiniteVolume () const
- STATES GetManifold () const
- float GetMaximumPointDistance () const
- float GetMeanPointDistance () const
- MeshPrimitive & GetMeshPrimitive ()
- MeshPrimitive const & GetMeshPrimitive () const
- unsigned long GetNumberOfSurfacePoints () const
- unsigned long GetNumberOfVectors () const
- DataElement & GetPointCoordinatesData ()

- const DataElement & GetPointCoordinatesData () const
- const float ∗ GetPointPositionAccuracy () const
- const float ∗ GetPointsBoundingBoxCoordinates () const
- SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm ()
- SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm () const
- const unsigned short ∗ GetRecommendedDisplayCIELabValue () const
- unsigned short GetRecommendedDisplayCIELabValue (const unsigned int idx) const
- unsigned short GetRecommendedDisplayGrayscaleValue () const
- float GetRecommendedPresentationOpacity () const
- VIEWType GetRecommendedPresentationType () const
- const char ∗ GetSurfaceComments () const
- unsigned long GetSurfaceNumber () const
- bool GetSurfaceProcessing () const
- const char ∗ GetSurfaceProcessingDescription () const
- float GetSurfaceProcessingRatio () const
- const float ∗ GetVectorAccuracy () const
- DataElement & GetVectorCoordinateData ()
- const DataElement & GetVectorCoordinateData () const
- unsigned short GetVectorDimensionality () const
- void SetAlgorithmFamily (SegmentHelper::BasicCodedEntry const &BSE)
- void SetAlgorithmName (const char ∗str)
- void SetAlgorithmVersion (const char ∗str)
- void SetAxisOfRotation (const float ∗axis)
- void SetCenterOfRotation (const float ∗center)
- void SetFiniteVolume (STATES state)
- void SetManifold (STATES state)
- void SetMaximumPointDistance (float maximum)
- void SetMeanPointDistance (float average)
- void SetMeshPrimitive (MeshPrimitive &mp)
- void SetNumberOfSurfacePoints (const unsigned long nb)
- void SetNumberOfVectors (const unsigned long nb)
- void SetPointCoordinatesData (DataElement const &de)
- void SetPointPositionAccuracy (const float ∗accuracies)
- void SetPointsBoundingBoxCoordinates (const float ∗coordinates)
- void SetProcessingAlgorithm (SegmentHelper::BasicCodedEntry const &BSE)
- void SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > &vl)
- void SetRecommendedDisplayCIELabValue (const unsigned short vl, const unsigned int idx=0)
- void SetRecommendedDisplayCIELabValue (const unsigned short vl[3])
- void SetRecommendedDisplayGrayscaleValue (const unsigned short vl)
- void SetRecommendedPresentationOpacity (const float opacity)
- void SetRecommendedPresentationType (VIEWType type)
- void SetSurfaceComments (const char ∗comment)
- void SetSurfaceNumber (const unsigned long nb)
- void SetSurfaceProcessing (bool b)
- void SetSurfaceProcessingDescription (const char ∗description)
- void SetSurfaceProcessingRatio (const float ratio)
- void SetVectorAccuracy (const float ∗accuracy)
- void SetVectorCoordinateData (DataElement const &de)
- void SetVectorDimensionality (const unsigned short dim)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Static Public Member Functions

- static STATES GetSTATES (const char *state)
- static const char * GetSTATESString (STATES state)
- static VIEWType GetVIEWType (const char *type)
- static const char * GetVIEWTypeString (VIEWType type)

Additional Inherited Members

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

12.302.1 Detailed Description

This class defines a SURFACE IE.

This members are taken from required surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

12.302.2 Member Enumeration Documentation

12.302.2.1 STATES

`enum gdcm::Surface::STATES`

Enumerator

NO	

YES	
UNKNOWN	
STATES_END	

12.302.2.2 VIEWType

`enum gdcm::Surface::VIEWType`

Enumeration for Recommended Presentation Type.

See also

Tag(0x0066, 0x000D) and PS 3.3 C.27.1.1.3

Enumerator

SURFACE	
WIREFRAME	
POINTS	
VIEWType_END	

12.302.3 Constructor & Destructor Documentation

12.302.3.1 Surface()

`gdcm::Surface::Surface ()`

12.302.3.2 ~Surface()

`gdcm::Surface::~Surface () [override]`

12.302.4 Member Function Documentation

12.302.4.1 GetAlgorithmFamily() [1/2]

`SegmentHelper::BasicCodedEntry & gdcm::Surface::GetAlgorithmFamily ()`

12.302.4.2 GetAlgorithmFamily() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Surface::GetAlgorithmFamily () const
```

12.302.4.3 GetAlgorithmName()

```
const char * gdcm::Surface::GetAlgorithmName () const
```

12.302.4.4 GetAlgorithmVersion()

```
const char * gdcm::Surface::GetAlgorithmVersion () const
```

12.302.4.5 GetAxisOfRotation()

```
const float * gdcm::Surface::GetAxisOfRotation () const
```

Note

Pointer is null if undefined

12.302.4.6 GetCenterOfRotation()

```
const float * gdcm::Surface::GetCenterOfRotation () const
```

Note

Pointer is null if undefined

12.302.4.7 GetFiniteVolume()

```
STATES gdcm::Surface::GetFiniteVolume () const
```

12.302.4.8 GetManifold()

```
STATES gdcm::Surface::GetManifold () const
```

12.302.4.9 GetMaximumPointDistance()

```
float gdcm::Surface::GetMaximumPointDistance () const
```

12.302.4.10 GetMeanPointDistance()

```
float gdcm::Surface::GetMeanPointDistance () const
```

12.302.4.11 GetMeshPrimitive() [1/2]

```
MeshPrimitive & gdcm::Surface::GetMeshPrimitive ()
```

12.302.4.12 GetMeshPrimitive() [2/2]

```
MeshPrimitive const & gdcm::Surface::GetMeshPrimitive () const
```

12.302.4.13 GetNumberOfSurfacePoints()

```
unsigned long gdcm::Surface::GetNumberOfSurfacePoints () const
```

12.302.4.14 GetNumberOfVectors()

```
unsigned long gdcm::Surface::GetNumberOfVectors () const
```

12.302.4.15 GetPointCoordinatesData() [1/2]

```
DataElement & gdcm::Surface::GetPointCoordinatesData ()
```

12.302.4.16 GetPointCoordinatesData() [2/2]

```
const DataElement & gdcm::Surface::GetPointCoordinatesData () const
```

12.302.4.17 GetPointPositionAccuracy()

```
const float * gdcm::Surface::GetPointPositionAccuracy () const
```

Note

Pointer is null if undefined

12.302.4.18 GetPointsBoundingBoxCoordinates()

```
const float * gdcm::Surface::GetPointsBoundingBoxCoordinates () const
```

Note

Pointer is null if undefined

12.302.4.19 GetProcessingAlgorithm() [1/2]

```
SegmentHelper::BasicCodedEntry & gdcm::Surface::GetProcessingAlgorithm ()
```

12.302.4.20 GetProcessingAlgorithm() [2/2]

```
SegmentHelper::BasicCodedEntry const & gdcm::Surface::GetProcessingAlgorithm () const
```

12.302.4.21 GetRecommendedDisplayCIELabValue() [1/2]

```
const unsigned short * gdcm::Surface::GetRecommendedDisplayCIELabValue () const
```

12.302.4.22 GetRecommendedDisplayCIELabValue() [2/2]

```
unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (
            const unsigned int idx) const
```

12.302.4.23 GetRecommendedDisplayGrayscaleValue()

```
unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue () const
```

12.302.4.24 GetRecommendedPresentationOpacity()

```
float gdcm::Surface::GetRecommendedPresentationOpacity () const
```

12.302.4.25 GetRecommendedPresentationType()

```
VIEWType gdcm::Surface::GetRecommendedPresentationType () const
```

12.302.4.26 GetSTATES()

```
STATES gdcm::Surface::GetSTATES (
            const char * state)  [static]
```

12.302.4.27 GetSTATESString()

```
const char * gdcm::Surface::GetSTATESString (
            STATES state)  [static]
```

12.302.4.28 GetSurfaceComments()

```
const char * gdcm::Surface::GetSurfaceComments () const
```

12.302.4.29 GetSurfaceNumber()

```
unsigned long gdcm::Surface::GetSurfaceNumber () const
```

12.302.4.30 GetSurfaceProcessing()

```
bool gdcm::Surface::GetSurfaceProcessing () const
```

12.302.4.31 GetSurfaceProcessingDescription()

```
const char * gdcm::Surface::GetSurfaceProcessingDescription () const
```

12.302.4.32 GetSurfaceProcessingRatio()

```
float gdcm::Surface::GetSurfaceProcessingRatio () const
```

12.302.4.33 GetVectorAccuracy()

```
const float * gdcm::Surface::GetVectorAccuracy () const
```

12.302.4.34 GetVectorCoordinateData() [1/2]

```
DataElement & gdcm::Surface::GetVectorCoordinateData ()
```

12.302.4.35 GetVectorCoordinateData() [2/2]

```
const DataElement & gdcm::Surface::GetVectorCoordinateData () const
```

12.302.4.36 GetVectorDimensionality()

```
unsigned short gdcm::Surface::GetVectorDimensionality () const
```

12.302.4.37 GetVIEWType()

```
VIEWType gdcm::Surface::GetVIEWType (
            const char * type)  [static]
```

12.302.4.38 GetVIEWTypeString()

```
const char * gdcm::Surface::GetVIEWTypeString (
            VIEWType type)  [static]
```

12.302.4.39 SetAlgorithmFamily()

```
void gdcm::Surface::SetAlgorithmFamily (
            SegmentHelper::BasicCodedEntry const & BSE)
```

12.302.4.40 SetAlgorithmName()

```
void gdcm::Surface::SetAlgorithmName (
            const char * str)
```

12.302.4.41 SetAlgorithmVersion()

```
void gdcm::Surface::SetAlgorithmVersion (
            const char * str)
```

12.302.4.42 SetAxisOfRotation()

```
void gdcm::Surface::SetAxisOfRotation (
            const float * axis)
```

12.302.4.43 SetCenterOfRotation()

```
void gdcm::Surface::SetCenterOfRotation (
            const float * center)
```

12.302.4.44 SetFiniteVolume()

```
void gdcm::Surface::SetFiniteVolume (
            STATES state)
```

12.302.4.45 SetManifold()

```
void gdcm::Surface::SetManifold (
            STATES state)
```

12.302.4.46 SetMaximumPointDistance()

```
void gdcm::Surface::SetMaximumPointDistance (
            float maximum)
```

12.302.4.47 SetMeanPointDistance()

```
void gdcm::Surface::SetMeanPointDistance (
            float average)
```

12.302.4.48 SetMeshPrimitive()

```
void gdcm::Surface::SetMeshPrimitive (
            MeshPrimitive & mp)
```

References gdcm::Object::SmartPointer.

12.302.4.49 SetNumberOfSurfacePoints()

```
void gdcm::Surface::SetNumberOfSurfacePoints (
            const unsigned long nb)
```

12.302.4.50 SetNumberOfVectors()

```
void gdcm::Surface::SetNumberOfVectors (
            const unsigned long nb)
```

12.302.4.51 SetPointCoordinatesData()

```
void gdcm::Surface::SetPointCoordinatesData (
            DataElement const & de)
```

12.302.4.52 SetPointPositionAccuracy()

```
void gdcm::Surface::SetPointPositionAccuracy (
            const float * accuracies)
```

12.302.4.53 SetPointsBoundingBoxCoordinates()

```
void gdcm::Surface::SetPointsBoundingBoxCoordinates (
            const float * coordinates)
```

12.302.4.54 SetProcessingAlgorithm()

```
void gdcm::Surface::SetProcessingAlgorithm (
            SegmentHelper::BasicCodedEntry const & BSE)
```

12.302.4.55 SetRecommendedDisplayCIELabValue() [1/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const std::vector< unsigned short > & vl)
```

12.302.4.56 SetRecommendedDisplayCIELabValue() [2/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const unsigned short vl,
            const unsigned int idx = 0)
```

12.302.4.57 SetRecommendedDisplayCIELabValue() [3/3]

```
void gdcm::Surface::SetRecommendedDisplayCIELabValue (
            const unsigned short vl[3])
```

12.302.4.58 SetRecommendedDisplayGrayscaleValue()

```
void gdcm::Surface::SetRecommendedDisplayGrayscaleValue (
            const unsigned short vl)
```

12.302.4.59 SetRecommendedPresentationOpacity()

```
void gdcm::Surface::SetRecommendedPresentationOpacity (
            const float opacity)
```

12.302.4.60 SetRecommendedPresentationType()

```
void gdcm::Surface::SetRecommendedPresentationType (
            VIEWType type)
```

12.302.4.61 SetSurfaceComments()

```
void gdcm::Surface::SetSurfaceComments (
            const char * comment)
```

12.302.4.62 SetSurfaceNumber()

```
void gdcm::Surface::SetSurfaceNumber (
            const unsigned long nb)
```

12.302.4.63 SetSurfaceProcessing()

```
void gdcm::Surface::SetSurfaceProcessing (
            bool b)
```

12.302.4.64 SetSurfaceProcessingDescription()

```
void gdcm::Surface::SetSurfaceProcessingDescription (
            const char * description)
```

12.302.4.65 SetSurfaceProcessingRatio()

```
void gdcm::Surface::SetSurfaceProcessingRatio (
            const float ratio)
```

12.302.4.66 SetVectorAccuracy()

```
void gdcm::Surface::SetVectorAccuracy (
            const float * accuracy)
```

12.302.4.67 SetVectorCoordinateData()

```
void gdcm::Surface::SetVectorCoordinateData (
            DataElement const & de)
```

12.302.4.68 SetVectorDimensionality()

```
void gdcm::Surface::SetVectorDimensionality (
            const unsigned short dim)
```

The documentation for this class was generated from the following file:

- gdcmSurface.h

12.303 gdcm::SurfaceHelper Class Reference

SurfaceHelper.

```
#include <gdcmSurfaceHelper.h>
```

Public Types

- typedef std::vector< unsigned short > ColorArray

Static Public Member Functions

- template<typename U>
  static std::vector< float > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U range↩Max=255)

  Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename T, typename U>
  static std::vector< T > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U range↩Max=255)

  Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename T, typename U>
  static ColorArray RGBToRecommendedDisplayCIELab (const std::vector< T > &RGB, const U rangeMax=255)

  Convert a RGB color into DICOM CIE-Lab (ready to write).
- template<typename T, typename U>
  static unsigned short RGBToRecommendedDisplayGrayscale (const std::vector< T > &RGB, const U range↩Max=255)

  Convert a RGB color into DICOM grayscale (ready to write).

12.303.1 Detailed Description

SurfaceHelper.

Helper class for Surface object

12.303.2 Member Typedef Documentation

12.303.2.1 ColorArray

```
typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray
```

12.303.3 Member Function Documentation

12.303.3.1 RecommendedDisplayCIELabToRGB() [1/2]

```
template<typename U>
std::vector< float > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (
            const ColorArray & CIELab,
            const U rangeMax = 255)  [static]
```

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

CIELab	DICOM CIE-Lab array.
rangeMax	Max value of the RGB range.

Template Parameters

U	Type of rangeMax value.

References RecommendedDisplayCIELabToRGB().

12.303.3.2 RecommendedDisplayCIELabToRGB() [2/2]

```
template<typename T, typename U>
std::vector< T > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB (
            const ColorArray & CIELab,
            const U rangeMax = 255)  [static]
```

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

CIELab	DICOM CIE-Lab array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of CIELab components.
U	Type of rangeMax value.

Referenced by RecommendedDisplayCIELabToRGB().

12.303.3.3 RGBToRecommendedDisplayCIELab()

```
template<typename T, typename U>
SurfaceHelper::ColorArray gdcm::SurfaceHelper::RGBToRecommendedDisplayCIELab (
            const std::vector< T > & RGB,
            const U rangeMax = 255)  [static]
```

Convert a RGB color into DICOM CIE-Lab (ready to write).

See also

PS 3.3 C.10.7.1.1

Parameters

RGB	RGB array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of RGB components.
U	Type of rangeMax value.

12.303.3.4 RGBToRecommendedDisplayGrayscale()

```
template<typename T, typename U>
unsigned short gdcm::SurfaceHelper::RGBToRecommendedDisplayGrayscale (
            const std::vector< T > & RGB,
            const U rangeMax = 255)  [static]
```

Convert a RGB color into DICOM grayscale (ready to write).

See also

PS 3.3 C.27.1 tag(0062,000C)

Parameters

RGB	RGB array.
rangeMax	Max value of the RGB range.

Template Parameters

T	Type of RGB components.
U	Type of rangeMax value.

The documentation for this class was generated from the following file:

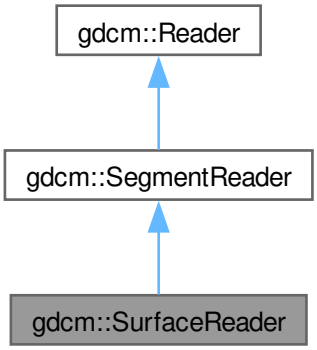
- gdcmSurfaceHelper.h

12.304 gdcm::SurfaceReader Class Reference

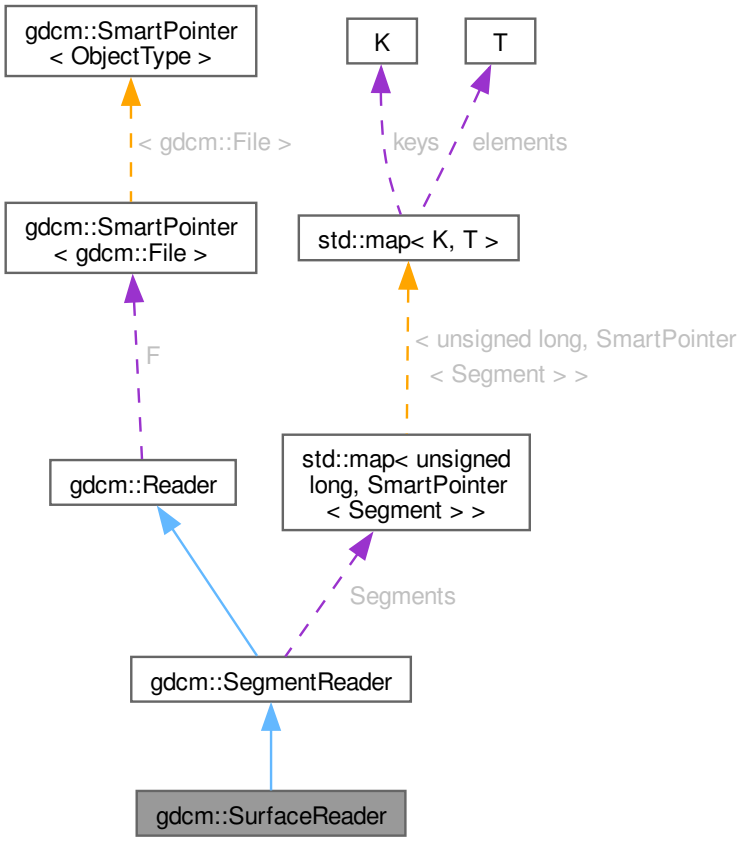
This class defines a SURFACE IE reader.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for gdcm::SurfaceReader:



Collaboration diagram for gdcm::SurfaceReader:



Public Member Functions

- SurfaceReader ()
- ~SurfaceReader () override
- unsigned long GetNumberOfSurfaces () const
- bool Read () override

  Read.

Public Member Functions inherited from gdcm::SegmentReader

- SegmentReader ()
- ~SegmentReader () override
- SegmentVector GetSegments ()
- SegmentVector GetSegments () const

Public Member Functions inherited from gdcm::Reader

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- File & GetFile ()

  Set/Get File.
- const File & GetFile () const

  Set/Get File.
- size_t GetStreamCurrentPosition () const
- bool ReadSelectedPrivateTags (std::set< PrivateTag > const &ptags, bool readvalues=true)

  Will only read the specified selected private tags.
- bool ReadSelectedTags (std::set< Tag > const &tags, bool readvalues=true)

  Will only read the specified selected tags.
- bool ReadUpToTag (const Tag &tag, std::set< Tag > const &skiptags=std::set< Tag >())
- void SetFile (File &file)

  Set/Get File.
- void SetFileName (const char *filename_native)
- void SetStream (std::istream &input_stream)

  Set the open-ed stream directly.

Protected Member Functions

- bool ReadPointMacro (SmartPointer< Surface > surface, const DataSet &surfaceDS)
- bool ReadSurface (const Item &surfaceItem, const unsigned long idx)
- bool ReadSurfaces ()

Protected Member Functions inherited from gdcm::SegmentReader

- bool ReadSegment (const Item &segmentItem, const unsigned int idx)
- bool ReadSegments ()

Protected Member Functions inherited from gdcm::Reader

- std::istream * GetStreamPtr () const
- bool ReadDataSet ()
- bool ReadMetaInformation ()
- bool ReadPreamble ()

Additional Inherited Members

Public Types inherited from gdcm::SegmentReader

- typedef std::vector< SmartPointer< Segment > > SegmentVector

Protected Types inherited from gdcm::SegmentReader

- typedef std::map< unsigned long, SmartPointer< Segment > > SegmentMap

Protected Attributes inherited from gdcm::SegmentReader

- SegmentMap Segments

Protected Attributes inherited from gdcm::Reader

- SmartPointer< File > F

12.304.1 Detailed Description

This class defines a SURFACE IE reader.

It reads surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

12.304.2 Constructor & Destructor Documentation

12.304.2.1 SurfaceReader()

```
gdcm::SurfaceReader::SurfaceReader ()
```

12.304.2.2 ~SurfaceReader()

```
gdcm::SurfaceReader::~SurfaceReader ()  [override]
```

12.304.3 Member Function Documentation

12.304.3.1 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces () const
```

12.304.3.2 Read()

```
bool gdcm::SurfaceReader::Read () [override], [virtual]
```

Read.

Reimplemented from gdcm::SegmentReader.

12.304.3.3 ReadPointMacro()

```
bool gdcm::SurfaceReader::ReadPointMacro (
            SmartPointer< Surface > surface,
            const DataSet & surfaceDS) [protected]
```

12.304.3.4 ReadSurface()

```
bool gdcm::SurfaceReader::ReadSurface (
            const Item & surfaceItem,
            const unsigned long idx) [protected]
```

12.304.3.5 ReadSurfaces()

```
bool gdcm::SurfaceReader::ReadSurfaces () [protected]
```

The documentation for this class was generated from the following file:

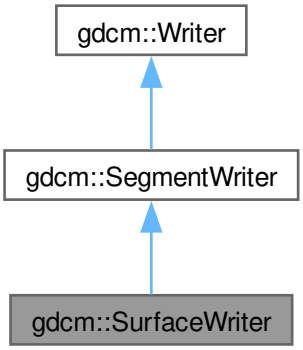
- gdcmSurfaceReader.h

12.305 gdcm::SurfaceWriter Class Reference

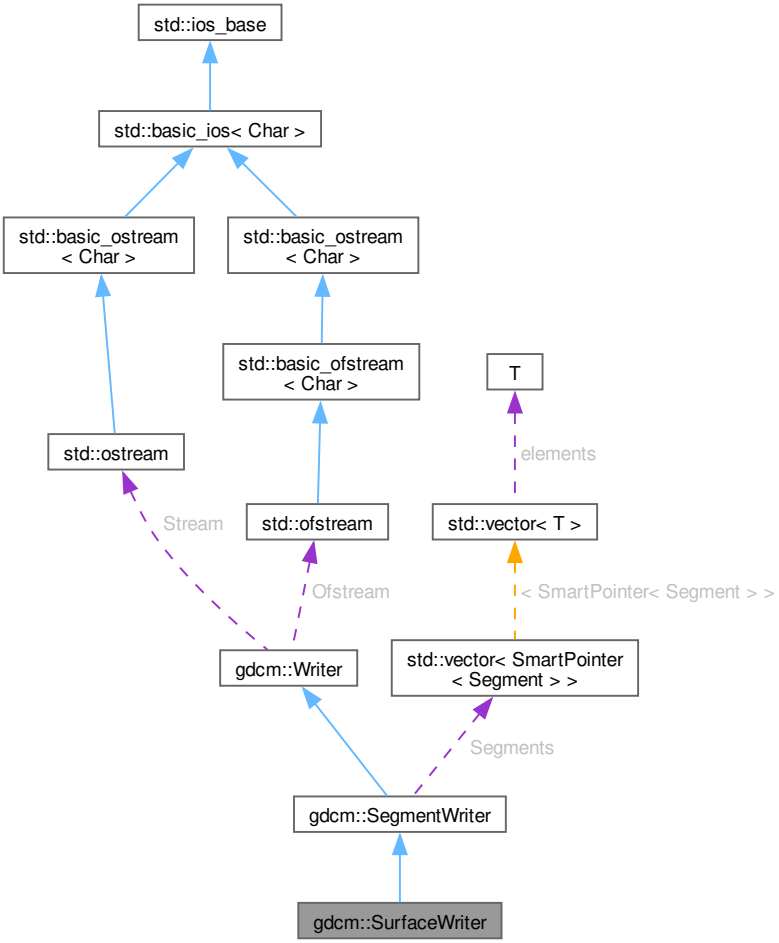
This class defines a SURFACE IE writer.

```
#include <gdcmSurfaceWriter.h>
```

Inheritance diagram for gdcm::SurfaceWriter:



Collaboration diagram for gdcm::SurfaceWriter:



Public Member Functions

- SurfaceWriter ()
- ~SurfaceWriter () override
- unsigned long GetNumberOfSurfaces ()
- void SetNumberOfSurfaces (const unsigned long nb)
- bool Write () override

  Write.

Public Member Functions inherited from gdcm::SegmentWriter

- SegmentWriter ()
- ~SegmentWriter () override

- void AddSegment (SmartPointer< Segment > segment)
- unsigned int GetNumberOfSegments () const
- SmartPointer< Segment > GetSegment (const unsigned int idx=0) const
- SegmentVector & GetSegments ()
- const SegmentVector & GetSegments () const
- void SetNumberOfSegments (const unsigned int size)
- void SetSegments (SegmentVector &segments)

Public Member Functions inherited from gdcm::Writer

- Writer ()
- virtual ~Writer ()
- void CheckFileMetaInformationOff ()
- void CheckFileMetaInformationOn ()
- File & GetFile ()
- void SetCheckFileMetaInformation (bool b)

  Undocumented function, do not use (= leave default).
- void SetFile (const File &f)

  Set/Get the DICOM file (DataSet + Header).
- void SetFileName (const char *filename_native)

  Set the filename of DICOM file to write:
- void SetStream (std::ostream &output_stream)

  Set user ostream buffer.

Protected Member Functions

- void ComputeNumberOfSurfaces ()
- bool PrepareWrite ()
- bool PrepareWritePointMacro (SmartPointer< Surface > surface, DataSet &surfaceDS, const TransferSyntax &ts)

Protected Member Functions inherited from gdcm::SegmentWriter

- bool PrepareWrite ()

Protected Member Functions inherited from gdcm::Writer

- bool GetCheckFileMetaInformation () const
- std::ostream * GetStreamPtr () const
- void SetWriteDataSetOnly (bool b)

Protected Attributes

- unsigned long NumberOfSurfaces

Protected Attributes inherited from gdcm::SegmentWriter

- SegmentVector Segments

Protected Attributes inherited from gdcm::Writer

- std::ofstream * Ofstream
- std::ostream * Stream

Additional Inherited Members

Public Types inherited from gdcm::SegmentWriter

- typedef std::vector< SmartPointer< Segment > > SegmentVector

12.305.1 Detailed Description

This class defines a SURFACE IE writer.

It writes surface mesh module attributes.

See also

PS 3.3 A.1.2.18 , A.57 and C.27

12.305.2 Constructor & Destructor Documentation

12.305.2.1 SurfaceWriter()

```
gdcm::SurfaceWriter::SurfaceWriter ()
```

12.305.2.2 ~SurfaceWriter()

```
gdcm::SurfaceWriter::~SurfaceWriter () [override]
```

12.305.3 Member Function Documentation

12.305.3.1 ComputeNumberOfSurfaces()

```
void gdcm::SurfaceWriter::ComputeNumberOfSurfaces () [protected]
```

12.305.3.2 GetNumberOfSurfaces()

```
unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ()
```

12.305.3.3 PrepareWrite()

```
bool gdcm::SurfaceWriter::PrepareWrite ()  [protected]
```

12.305.3.4 PrepareWritePointMacro()

```
bool gdcm::SurfaceWriter::PrepareWritePointMacro (
            SmartPointer< Surface > surface,
            DataSet & surfaceDS,
            const TransferSyntax & ts)  [protected]
```

12.305.3.5 SetNumberOfSurfaces()

```
void gdcm::SurfaceWriter::SetNumberOfSurfaces (
            const unsigned long nb)
```

12.305.3.6 Write()

```
bool gdcm::SurfaceWriter::Write ()  [override], [virtual]
```

Write.

Reimplemented from gdcm::SegmentWriter.

12.305.4 Member Data Documentation

12.305.4.1 NumberOfSurfaces

```
unsigned long gdcm::SurfaceWriter::NumberOfSurfaces  [protected]
```

The documentation for this class was generated from the following file:

- gdcmSurfaceWriter.h

12.306 gdcm::SwapCode Class Reference

SwapCode representation.

```
#include <gdcmSwapCode.h>
```

Public Types

- enum SwapCodeType {
  Unknown = 0 ,
  LittleEndian = 1234 ,
  BigEndian = 4321 ,
  BadLittleEndian = 3412 ,
  BadBigEndian = 2143 }

Public Member Functions

- SwapCode (SwapCodeType sc=Unknown)
- operator SwapCode::SwapCodeType () const

Static Public Member Functions

- static const char * GetSwapCodeString (SwapCode const &sc)

Static Protected Member Functions

- static int GetIndex (SwapCode const &sc)

Friends

- std::ostream & operator<< (std::ostream &os, const SwapCode &sc)

12.306.1 Detailed Description

SwapCode representation.

Examples

TestByteSwap.cxx.

12.306.2 Member Enumeration Documentation

12.306.2.1 SwapCodeType

`enum gdcm::SwapCode::SwapCodeType`

Enumerator

Unknown	
LittleEndian	
BigEndian	
BadLittleEndian	
BadBigEndian	

12.306.3 Constructor & Destructor Documentation

12.306.3.1 SwapCode()

```
gdcm::SwapCode::SwapCode (
            SwapCodeType sc = Unknown)  [inline]
```

References Unknown.

Referenced by GetIndex(), GetSwapCodeString(), and operator<<.

12.306.4 Member Function Documentation

12.306.4.1 GetIndex()

```
int gdcm::SwapCode::GetIndex (
            SwapCode const & sc)  [static], [protected]
```

References SwapCode().

12.306.4.2 GetSwapCodeString()

```
const char * gdcm::SwapCode::GetSwapCodeString (
            SwapCode const & sc)  [static]
```

References SwapCode(), and operator<<.

Referenced by operator<<.

12.306.4.3 operator SwapCode::SwapCodeType()

```
gdcm::SwapCode::operator SwapCode::SwapCodeType () const  [inline]
```

12.306.5 Friends And Related Symbol Documentation

12.306.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const SwapCode & sc)  [friend]
```

References SwapCode(), and GetSwapCodeString().

Referenced by GetSwapCodeString().

The documentation for this class was generated from the following file:

- gdcmSwapCode.h

12.307 gdcm::SwapperDoOp Class Reference

```
#include <gdcmSwapper.h>
```

Static Public Member Functions

- template<typename T>
  static T Swap (T val)
- template<typename T>
  static void SwapArray (T *array, size_t n)

12.307.1 Member Function Documentation

12.307.1.1 Swap()

```
template<typename T>
T gdcm::SwapperDoOp::Swap (
            T val)  [static]
```

Referenced by gdcm::Item::Read(), and SwapArray().

12.307.1.2 SwapArray()

```
template<typename T>
void gdcm::SwapperDoOp::SwapArray (
            T * array,
            size_t n)  [inline], [static]
```

References Swap().

The documentation for this class was generated from the following file:

- gdcmSwapper.h

12.308 gdcm::SwapperNoOp Class Reference

```
#include <gdcmSwapper.h>
```

Static Public Member Functions

- template<typename T>
  static T Swap (T val)
- template<typename T>
  static void SwapArray (T *, size_t)

12.308.1 Detailed Description

Examples

DumpSiemensBase64.cxx, DumpToshibaDTI.cxx, DumpToshibaDTI2.cxx, and ReadExplicitLengthSQIVR.cxx.

12.308.2 Member Function Documentation

12.308.2.1 Swap()

```
template<typename T>
T gdcm::SwapperNoOp::Swap (
            T val)  [inline], [static]
```

Referenced by gdcm::EncodingImplementation< VR::VRBINARY >::Write().

12.308.2.2 SwapArray()

```
template<typename T>
void gdcm::SwapperNoOp::SwapArray (
            T * ,
            size_t )  [inline], [static]
```

Referenced by gdcm::EncodingImplementation< VR::VRBINARY >::Read().

The documentation for this class was generated from the following file:

- gdcmSwapper.h

12.309 gdcm::System Class Reference

Class to do system operation.

```
#include <gdcmSystem.h>
```

Static Public Member Functions

- static std::wstring ConvertToUNC (const char ∗utf8path)
- static bool DeleteDirectory (const char ∗source)

  remove a directory named source
- static size_t EncodeBytes (char ∗out, const unsigned char ∗data, int size)
- static bool FileExists (const char ∗filename)

  Check whether the specified file exist on the system.
- static bool FileIsDirectory (const char ∗name)

  Check whether the file specified is a directory:
- static bool FileIsSymlink (const char ∗name)

  Check whether name is a symlink.
- static size_t FileSize (const char ∗filename)
- static time_t FileTime (const char ∗filename)
- static bool FormatDateTime (char date[22], time_t t, long milliseconds=0)
- static bool GetCurrentDateTime (char date[22])
- static const char ∗ GetCurrentModuleFileName ()
- static const char ∗ GetCurrentProcessFileName ()
- static const char ∗ GetCurrentResourcesDirectory ()
- static const char ∗ GetCWD ()
- static bool GetHostName (char hostname[255])
- static const char ∗ GetLastSystemError ()

  Return the last error.
- static const char ∗ GetLocaleCharset ()

  return `locale charmap`
- static const char ∗ GetTimezoneOffsetFromUTC ()
- static bool MakeDirectory (const char ∗path)

  Create a directory name path.
- static bool ParseDateTime (time_t &timep, const char date[22])

  Parse a date stored as ASCII text into a time_t structured (discard millisecond if any).
- static bool ParseDateTime (time_t &timep, long &milliseconds, const char date[22])
- static bool RemoveFile (const char ∗source)

  remove a file named source
- static int StrCaseCmp (const char ∗s1, const char ∗s2)

  consistent func for C99 spec of strcasecmp/strncasecmp
- static int StrNCaseCmp (const char ∗s1, const char ∗s2, size_t n)
- static char ∗ StrSep (char ∗∗stringp, const char ∗delim)
- static char ∗ StrTokR (char ∗ptr, const char ∗sep, char ∗∗end)

  strtok_r

Static Protected Member Functions

- static bool GetPermissions (const char ∗file, unsigned short &mode)

  NOT THREAD SAFE.
- static bool SetPermissions (const char ∗file, unsigned short mode)

12.309.1 Detailed Description

Class to do system operation.

OS independent functionalities

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, Cleaner.cs, ClinicalTrialIdentificationWorkflow.cs, CompressLossyJPEG.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, DumpCSA.cs, ExtractEncapsulatedFile.cs, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, ExtractOneFrame.cs, FileAnonymize.cs, FileChangeTS.cs, FileChangeTSLossy.cs, FileStreaming.cs, GetArray.cs, MetaImageMD5Activiz.cs, MpegVideoInfo.cs, ReformatFile.cs, RescaleImage.cs, ScanDirectory.cs, SimplePrint.cs, and StandardizeFiles.cs.

12.309.2 Member Function Documentation

12.309.2.1 ConvertToUNC()

```
std::wstring gdcm::System::ConvertToUNC (
            const char * utf8path)  [static]
```

When needed convert a PATH into a UNC equivalent. This allow transparent support for path longer that MAX_PATH. Only on _MSC_VER compiler, return empty string otherwise.

12.309.2.2 DeleteDirectory()

```
bool gdcm::System::DeleteDirectory (
            const char * source)  [static]
```

remove a directory named source

12.309.2.3 EncodeBytes()

```
size_t gdcm::System::EncodeBytes (
            char * out,
            const unsigned char * data,
            int size)  [static]
```

Used internally by the UIDGenerator class to convert a uuid tape to a DICOM VR:UI type

12.309.2.4 FileExists()

```
bool gdcm::System::FileExists (
            const char * filename)  [static]
```

Check whether the specified file exist on the system.

Examples

DumpVisusChange.cxx, EncapsulateFileInRawData.cxx, MagnifyFile.cxx, and gdcmorthoplanes.cxx.

12.309.2.5 FileIsDirectory()

```
bool gdcm::System::FileIsDirectory (
            const char * name)  [static]
```

Check whether the file specified is a directory:

Examples

DumpVisusChange.cxx, gdcmorthoplanes.cxx, and threadgdcm.cxx.

12.309.2.6 FileIsSymlink()

```
bool gdcm::System::FileIsSymlink (
            const char * name)  [static]
```

Check whether name is a symlink.

12.309.2.7 FileSize()

```
size_t gdcm::System::FileSize (
            const char * filename)  [static]
```

Return the filesize. 0 if file does not exist.

Warning

you need to use FileExists to differentiate between empty file and missing file.

for very large size file and on system where size_t is not appropriate to store off_t value the function will return 0.

Examples

CheckBigEndianBug.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, EncapsulateFileInRawData.cxx, and SimpleScanner.cxx.

12.309.2.8 FileTime()

```
time_t gdcm::System::FileTime (
            const char * filename)  [static]
```

Return the time of last modification of file 0 if the file does not exist

12.309.2.9 FormatDateTime()

```
bool gdcm::System::FormatDateTime (
            char date[22],
            time_t t,
            long milliseconds = 0)  [static]
```

format as ASCII text a time_t with milliseconds See VR::DT from DICOM PS 3.5 milliseconds is in the range [0, 999999]

12.309.2.10 GetCurrentDateTime()

```
bool gdcm::System::GetCurrentDateTime (
            char date[22])  [static]
```

Return the current data time, and format it as ASCII text. This is simply a call to gettimeofday + FormatDateTime, since WIN32 do not have an implementation for gettimeofday, this is more portable. The call time(0) is not precise for our resolution

Examples

TemplateEmptyImage.cxx.

12.309.2.11 GetCurrentModuleFileName()

```
const char * gdcm::System::GetCurrentModuleFileName ()  [static]
```

Return the directory the current module is located: NOT THREAD SAFE

12.309.2.12 GetCurrentProcessFileName()

```
const char * gdcm::System::GetCurrentProcessFileName ()  [static]
```

Return the directory the current process (executable) is located: NOT THREAD SAFE

12.309.2.13 GetCurrentResourcesDirectory()

```
const char * gdcm::System::GetCurrentResourcesDirectory ()  [static]
```

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

12.309.2.14 GetCWD()

```
const char * gdcm::System::GetCWD ()  [static]
```

Return current working directory Warning: if current working path is too long (>2048 bytes) the call will fail and call will return NULL NOT THREAD SAFE

12.309.2.15 GetHostName()

```
bool gdcm::System::GetHostName (
            char hostname[255])  [static]
```

Retrieve the hostname, only the first 255 byte are copied. This may come handy to specify the Station Name

12.309.2.16 GetLastSystemError()

```
const char * gdcm::System::GetLastSystemError ()  [static]
```

Return the last error.

12.309.2.17 GetLocaleCharset()

```
const char * gdcm::System::GetLocaleCharset ()  [static]
```

return `locale charmap`

12.309.2.18 GetPermissions()

```
bool gdcm::System::GetPermissions (
            const char * file,
            unsigned short & mode)  [static], [protected]
```

NOT THREAD SAFE.

12.309.2.19 GetTimezoneOffsetFromUTC()

```
const char * gdcm::System::GetTimezoneOffsetFromUTC ()  [static]
```

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

12.309.2.20 MakeDirectory()

```
bool gdcm::System::MakeDirectory (
            const char * path)  [static]
```

Create a directory name path.

12.309.2.21 ParseDateTime() [1/2]

```
bool gdcm::System::ParseDateTime (
            time_t & timep,
            const char date[22])  [static]
```

Parse a date stored as ASCII text into a time_t structured (discard millisecond if any).

12.309.2.22 ParseDateTime() [2/2]

```
bool gdcm::System::ParseDateTime (
            time_t & timep,
            long & milliseconds,
            const char date[22])  [static]
```

Parse a date stored as ASCII text into a time_t structured and millisecond

See also

FormatDateTime

12.309.2.23 RemoveFile()

```
bool gdcm::System::RemoveFile (
            const char * source)  [static]
```

remove a file named source

12.309.2.24 SetPermissions()

```
bool gdcm::System::SetPermissions (
            const char * file,
            unsigned short mode)  [static], [protected]
```

12.309.2.25 StrCaseCmp()

```
int gdcm::System::StrCaseCmp (
            const char * s1,
            const char * s2)  [static]
```

consistent func for C99 spec of strcasecmp/strncasecmp

12.309.2.26 StrNCaseCmp()

```
int gdcm::System::StrNCaseCmp (
            const char * s1,
            const char * s2,
            size_t n)  [static]
```

Precondition

n != 0

12.309.2.27 StrSep()

```
char * gdcm::System::StrSep (
            char ** stringp,
            const char * delim)  [static]
```

strsep param stringp is passed by pointer, it may be modified, you'll need to make a copy, in case you want to free the memory pointed at

12.309.2.28 StrTokR()

```
char * gdcm::System::StrTokR (
            char * ptr,
            const char * sep,
            char ** end)  [static]
```

strtok_r

The documentation for this class was generated from the following file:

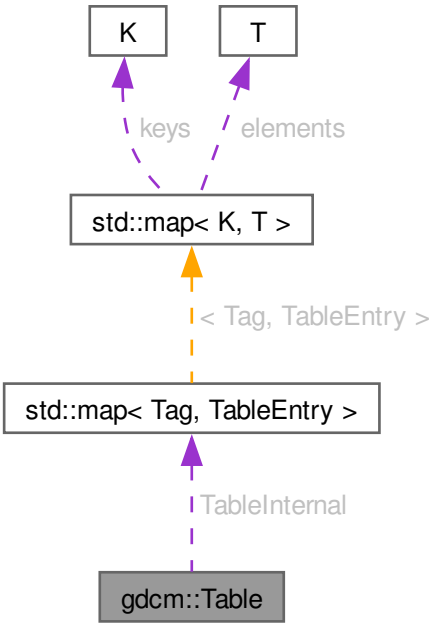
- gdcmSystem.h

12.310 gdcm::Table Class Reference

Table.

```
#include <gdcmTable.h>
```

Collaboration diagram for gdcm::Table:



Public Types

- typedef std::map< Tag, TableEntry > MapTableEntry

Public Member Functions

- Table ()=default
- Table (const Table &_val)=delete
- ~Table ()=default
- const TableEntry & GetTableEntry (const Tag &tag) const
- void InsertEntry (Tag const &tag, TableEntry const &te)
- Table & operator= (const Table &_val)=delete

Public Attributes

- MapTableEntry TableInternal

Friends

- std::ostream & operator<< (std::ostream &_os, const Table &_val)

12.310.1 Detailed Description

Table.

12.310.2 Member Typedef Documentation

12.310.2.1 MapTableEntry

```
typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry
```

12.310.3 Constructor & Destructor Documentation

12.310.3.1 Table() [1/2]

```
gdcm::Table::Table () [default]
```

Referenced by Table(), operator<<, and operator=().

12.310.3.2 ~Table()

```
gdcm::Table::~Table () [default]
```

12.310.3.3 Table() [2/2]

```
gdcm::Table::Table (
            const Table & _val) [delete]
```

References Table().

12.310.4 Member Function Documentation

12.310.4.1 GetTableEntry()

```
const TableEntry & gdcm::Table::GetTableEntry (
            const Tag & tag) const [inline]
```

References GetTableEntry(), and TableInternal.

Referenced by GetTableEntry().

12.310.4.2 InsertEntry()

```
void gdcm::Table::InsertEntry (
            Tag const & tag,
            TableEntry const & te)  [inline]
```

References TableInternal.

12.310.4.3 operator=()

```
Table & gdcm::Table::operator= (
            const Table & _val)  [delete]
```

References Table().

12.310.5 Friends And Related Symbol Documentation

12.310.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Table & _val)  [friend]
```

References Table().

12.310.6 Member Data Documentation

12.310.6.1 TableInternal

```
MapTableEntry gdcm::Table::TableInternal
```

Referenced by GetTableEntry(), and InsertEntry().

The documentation for this class was generated from the following file:

- gdcmTable.h

12.311 gdcm::TableEntry Class Reference

TableEntry.

```
#include <gdcmTableEntry.h>
```

Public Member Functions

- TableEntry (const char *attribute=nullptr, Type const &type=Type(), const char *des=nullptr)
- ~TableEntry ()=default

12.311.1 Detailed Description

TableEntry.

12.311.2 Constructor & Destructor Documentation

12.311.2.1 TableEntry()

```
gdcm::TableEntry::TableEntry (
            const char * attribute = nullptr,
            Type const & type = Type(),
            const char * des = nullptr)  [inline]
```

12.311.2.2 ~TableEntry()

```
gdcm::TableEntry::~TableEntry ()  [default]
```

The documentation for this class was generated from the following file:

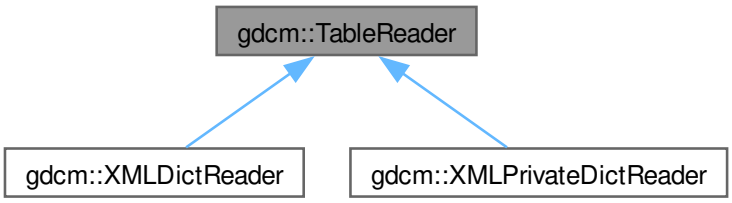
- gdcmTableEntry.h

12.312 gdcm::TableReader Class Reference

Class for representing a TableReader.

```
#include <gdcmTableReader.h>
```

Inheritance diagram for gdcm::TableReader:



Public Member Functions

- TableReader (Defs &defs)
- virtual ~TableReader ()=default
- virtual void CharacterDataHandler (const char *data, int length)
- virtual void EndElement (const char *name)
- const Defs & GetDefs () const
- const char * GetFilename ()
- void HandleIOD (const char **atts)
- void HandleIODEntry (const char **atts)
- void HandleMacro (const char **atts)
- void HandleMacroEntry (const char **atts)
- void HandleMacroEntryDescription (const char **atts)
- void HandleModule (const char **atts)
- void HandleModuleEntry (const char **atts)
- void HandleModuleEntryDescription (const char **atts)
- void HandleModuleInclude (const char **atts)
- int Read ()
- void SetFilename (const char *filename)
- virtual void StartElement (const char *name, const char **atts)

12.312.1 Detailed Description

Class for representing a TableReader.

Note

This class is an empty shell meant to be derived

12.312.2 Constructor & Destructor Documentation

12.312.2.1 TableReader()

```
gdcm::TableReader::TableReader (
            Defs & defs)  [inline]
```

12.312.2.2 ~TableReader()

```
virtual gdcm::TableReader::~TableReader ()  [virtual], [default]
```

12.312.3 Member Function Documentation

12.312.3.1 CharacterDataHandler()

```
virtual void gdcm::TableReader::CharacterDataHandler (
            const char * data,
            int length)  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

12.312.3.2 EndElement()

```
virtual void gdcm::TableReader::EndElement (
            const char * name)  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

12.312.3.3 GetDefs()

```
const Defs & gdcm::TableReader::GetDefs () const  [inline]
```

12.312.3.4 GetFilename()

```
const char * gdcm::TableReader::GetFilename ()  [inline]
```

12.312.3.5 HandleIOD()

```
void gdcm::TableReader::HandleIOD (
            const char ** atts)
```

12.312.3.6 HandleIODEntry()

```
void gdcm::TableReader::HandleIODEntry (
            const char ** atts)
```

12.312.3.7 HandleMacro()

```
void gdcm::TableReader::HandleMacro (
            const char ** atts)
```

12.312.3.8 HandleMacroEntry()

```
void gdcm::TableReader::HandleMacroEntry (
            const char ** atts)
```

12.312.3.9 HandleMacroEntryDescription()

```
void gdcm::TableReader::HandleMacroEntryDescription (
            const char ** atts)
```

12.312.3.10 HandleModule()

```
void gdcm::TableReader::HandleModule (
            const char ** atts)
```

12.312.3.11 HandleModuleEntry()

```
void gdcm::TableReader::HandleModuleEntry (
            const char ** atts)
```

12.312.3.12 HandleModuleEntryDescription()

```
void gdcm::TableReader::HandleModuleEntryDescription (
            const char ** atts)
```

12.312.3.13 HandleModuleInclude()

```
void gdcm::TableReader::HandleModuleInclude (
            const char ** atts)
```

12.312.3.14 Read()

```
int gdcm::TableReader::Read ()
```

12.312.3.15 SetFilename()

```
void gdcm::TableReader::SetFilename (
            const char * filename)  [inline]
```

12.312.3.16 StartElement()

```
virtual void gdcm::TableReader::StartElement (
            const char * name,
            const char ** atts)  [virtual]
```

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

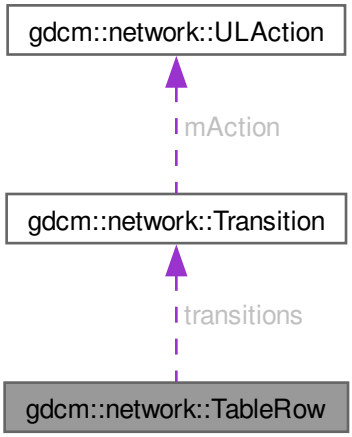
The documentation for this class was generated from the following file:

- gdcmTableReader.h

12.313 gdcm::network::TableRow Class Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::network::TableRow:



Public Member Functions

- TableRow ()
- ~TableRow ()

Public Attributes

- Transition * transitions [cMaxStateID]

12.313.1 Constructor & Destructor Documentation

12.313.1.1 TableRow()

```
gdcm::network::TableRow::TableRow () [inline]
```

References gdcm::network::cMaxStateID, and transitions.

12.313.1.2 ~TableRow()

```
gdcm::network::TableRow::~TableRow () [inline]
```

References gdcm::network::cMaxStateID, and transitions.

12.313.2 Member Data Documentation

12.313.2.1 transitions

```
Transition* gdcm::network::TableRow::transitions[cMaxStateID]
```

Referenced by TableRow(), and ~TableRow().

The documentation for this class was generated from the following file:

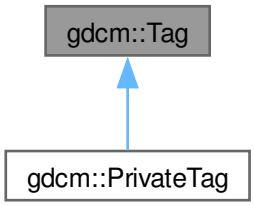
- gdcmULTransitionTable.h

12.314 gdcm::Tag Class Reference

Class to represent a DICOM Data Element (Attribute) Tag (Group, Element).

```
#include <gdcmTag.h>
```

Inheritance diagram for gdcm::Tag:



Public Member Functions

- Tag (const Tag &_val)
- Tag (uint16_t group, uint16_t element)

  Constructor with 2∗uint16_t.
- Tag (uint32_t tag=0)

  Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.
- uint16_t GetElement () const

  Returns the 'Element number' of the given Tag.
- uint32_t GetElementTag () const

  Returns the full tag value of the given Tag.
- uint16_t GetGroup () const

  Returns the 'Group number' of the given Tag.
- uint32_t GetLength () const

  return the length of tag (read: size on disk)
- Tag GetPrivateCreator () const

  Return the Private Creator Data Element tag of a private data element.
- bool IsGroupLength () const

  return whether the tag correspond to a group length tag:
- bool IsGroupXX (const Tag &t) const

  e.g 6002,3000 belong to groupXX: 6000,3000
- bool IsIllegal () const

  return if the tag is considered to be an illegal tag
- bool IsPrivate () const
- bool IsPrivateCreator () const
- bool IsPublic () const
- bool operator!= (const Tag &_val) const
- bool operator< (const Tag &_val) const
- bool operator<= (const Tag &t2) const
- Tag & operator= (const Tag &_val)
- bool operator== (const Tag &_val) const
- uint16_t & operator[ ] (const unsigned int &_id)

  Returns the Group or Element of the given Tag, depending on id (0/1).
- const uint16_t & operator[ ] (const unsigned int &_id) const

  Returns the Group or Element of the given Tag, depending on id (0/1).
- std::string PrintAsContinuousString () const
- std::string PrintAsContinuousUpperCaseString () const

  Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.
- std::string PrintAsPipeSeparatedString () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)

  Read a tag from binary representation.
- bool ReadFromCommaSeparatedString (const char ∗str)
- bool ReadFromContinuousString (const char ∗str)
- bool ReadFromPipeSeparatedString (const char ∗str)
- void SetElement (uint16_t element)

  Sets the 'Element number' of the given Tag.
- void SetElementTag (uint16_t group, uint16_t element)

Sets the 'Group number' & 'Element number' of the given Tag.

- void SetElementTag (uint32_t tag)

  Sets the full tag value of the given Tag.

- void SetGroup (uint16_t group)

  Sets the 'Group number' of the given Tag.

- void SetPrivateCreator (Tag const &t)

  Set private creator:

- template<typename TSwap>
  const std::ostream & Write (std::ostream &os) const

  Write a tag in binary rep.

Friends

- std::ostream & operator<< (std::ostream &_os, const Tag &_val)
- std::istream & operator>> (std::istream &_is, Tag &_val)

12.314.1 Detailed Description

Class to represent a DICOM Data Element (Attribute) Tag (Group, Element).

Basically an uint32_t which can also be expressed as two uint16_t (group and element)

Note

DATA ELEMENT TAG: A unique identifier for a Data Element composed of an ordered pair of numbers (a Group Number followed by an Element Number). GROUP NUMBER: The first number in the ordered pair of numbers that makes up a Data Element Tag. ELEMENT NUMBER: The second number in the ordered pair of numbers that makes up a Data Element Tag.

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangeSequenceUltrasound.cxx, Cleaner.cs, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DecompressImage.cs, DeriveSeries.cxx, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, DumpVisusChange.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, ExtractEncapsulatedFile.cs, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FileAnonymize.cs, FileChangeTS.cs, FileChangeTSLossy.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, MpegVideoInfo.cs, PatchFile.cxx, PublicDict.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndDumpDICOMDIR2.c ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ScanDirectory.cs, SimpleScanner.cxx, SortImage.cxx, StreamImageReaderTest.cxx, TraverseModules.cxx, VolumeSorter.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.314.2 Constructor & Destructor Documentation

12.314.2.1 Tag() [1/3]

```
gdcm::Tag::Tag (
            uint16_t group,
            uint16_t element)  [inline]
```

Constructor with 2∗uint16_t.

Referenced by gdcm::PrivateTag::PrivateTag(), gdcm::PrivateTag::PrivateTag(), Tag(), GetPrivateCreator(), IsGroupXX(), gdcm::PrivateTag::operator!=(), operator!=(), operator<(), operator<<, operator<=(), operator=(), gdcm::PrivateTag::operator==(), operator==(), operator>>, and SetPrivateCreator().

12.314.2.2 Tag() [2/3]

```
gdcm::Tag::Tag (
            uint32_t tag = 0)  [inline]
```

Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.

References SetElementTag(), and tag.

12.314.2.3 Tag() [3/3]

```
gdcm::Tag::Tag (
            const Tag & _val)  [inline]
```

References Tag(), and tag.

12.314.3 Member Function Documentation

12.314.3.1 GetElement()

```
uint16_t gdcm::Tag::GetElement () const  [inline]
```

Returns the 'Element number' of the given Tag.

Examples

DuplicatePCDE.cxx, and PublicDict.cxx.

Referenced by gdcm::PrivateTag::PrivateTag(), gdcm::DataSet::ComputeGroupLength(), GetPrivateCreator(), IsGroupLength(), IsGroupXX(), IsIllegal(), IsPrivateCreator(), gdcm::PrivateDict::PrintXML(), gdcm::SequenceOfFragments::ReadValue(), and SetPrivateCreator().

12.314.3.2 GetElementTag()

```
uint32_t gdcm::Tag::GetElementTag () const [inline]
```

Returns the full tag value of the given Tag.

Referenced by gdcm::PrivateTag::operator!=(), gdcm::PrivateTag::operator!=(), gdcm::PrivateTag::operator=(), gdcm::PrivateTag::operator==(), and gdcm::PrivateTag::operator==().

12.314.3.3 GetGroup()

```
uint16_t gdcm::Tag::GetGroup () const [inline]
```

Returns the 'Group number' of the given Tag.

Examples

DuplicatePCDE.cxx, and GenAllVR.cxx.

Referenced by gdcm::DataSet::ComputeGroupLength(), gdcm::CommandDataSet::Insert(), gdcm::DataSet::Insert(), gdcm::FileMetaInformation::Insert(), IsGroupXX(), IsIllegal(), gdcm::PrivateDict::PrintXML(), gdcm::SequenceOfFragments::ReadValue(), and SetPrivateCreator().

12.314.3.4 GetLength()

```
uint32_t gdcm::Tag::GetLength () const [inline]
```

return the length of tag (read: size on disk)

12.314.3.5 GetPrivateCreator()

```
Tag gdcm::Tag::GetPrivateCreator () const [inline]
```

Return the Private Creator Data Element tag of a private data element.

References Tag(), GetElement(), IsPrivate(), IsPrivateCreator(), and SetElement().

12.314.3.6 IsGroupLength()

```
bool gdcm::Tag::IsGroupLength () const [inline]
```

return whether the tag correspond to a group length tag:

References GetElement().

12.314.3.7 IsGroupXX()

```
bool gdcm::Tag::IsGroupXX (
            const Tag & t) const  [inline]
```

e.g 6002,3000 belong to groupXX: 6000,3000

References Tag(), GetElement(), GetGroup(), and IsPrivate().

12.314.3.8 IsIllegal()

```
bool gdcm::Tag::IsIllegal () const  [inline]
```

return if the tag is considered to be an illegal tag

References GetElement(), GetGroup(), and IsPrivate().

12.314.3.9 IsPrivate()

```
bool gdcm::Tag::IsPrivate () const  [inline]
```

PRIVATE DATA ELEMENT: Additional Data Element, defined by an implementor, to communicate information that is not contained in Standard Data Elements. Private Data elements have odd Group Numbers.

Examples

DuplicatePCDE.cxx.

References IsPublic().

Referenced by GetPrivateCreator(), IsGroupXX(), IsIllegal(), IsPrivateCreator(), and SetPrivateCreator().

12.314.3.10 IsPrivateCreator()

```
bool gdcm::Tag::IsPrivateCreator () const  [inline]
```

Returns if tag is a Private Creator (xxxx,00yy), where xxxx is odd number and yy in [0x10,0xFF]

Examples

DuplicatePCDE.cxx.

References GetElement(), and IsPrivate().

Referenced by GetPrivateCreator().

12.314.3.11 IsPublic()

```
bool gdcm::Tag::IsPublic () const  [inline]
```

STANDARD DATA ELEMENT: A Data Element defined in the DICOM Standard, and therefore listed in the DICOM Data Element Dictionary in PS 3.6. Is the Tag from the Public dict...well the implementation is buggy it does not prove the element is indeed in the dict...

Referenced by IsPrivate().

12.314.3.12 operator"!=()

```
bool gdcm::Tag::operator!= (
            const Tag & _val) const  [inline]
```

References Tag(), and tag.

12.314.3.13 operator<()

```
bool gdcm::Tag::operator< (
            const Tag & _val) const  [inline]
```

DICOM Standard expects the Data Element to be sorted by Tags All other comparison can be constructed from this one and operator ==

References Tag(), tag, and tags.

12.314.3.14 operator<=()

```
bool gdcm::Tag::operator<= (
            const Tag & t2) const  [inline]
```

References Tag().

12.314.3.15 operator=()

```
Tag & gdcm::Tag::operator= (
            const Tag & _val)  [inline]
```

References Tag(), and tag.

12.314.3.16 operator==()

```
bool gdcm::Tag::operator== (
            const Tag & _val) const  [inline]
```

References Tag(), and tag.

12.314.3.17 operator[]() [1/2]

```
uint16_t & gdcm::Tag::operator[] (
            const unsigned int & _id)  [inline]
```

Returns the Group or Element of the given Tag, depending on id (0/1).

12.314.3.18 operator[]() [2/2]

```
const uint16_t & gdcm::Tag::operator[] (
            const unsigned int & _id) const  [inline]
```

Returns the Group or Element of the given Tag, depending on id (0/1).

12.314.3.19 PrintAsContinuousString()

```
std::string gdcm::Tag::PrintAsContinuousString () const
```

Print tag value with no separating comma: eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

12.314.3.20 PrintAsContinuousUpperCaseString()

```
std::string gdcm::Tag::PrintAsContinuousUpperCaseString () const
```

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

12.314.3.21 PrintAsPipeSeparatedString()

```
std::string gdcm::Tag::PrintAsPipeSeparatedString () const
```

Print as a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

ReadFromPipeSeparatedString

12.314.3.22 Read()

```
template<typename TSwap>
std::istream & gdcm::Tag::Read (
            std::istream & is)  [inline]
```

Read a tag from binary representation.

12.314.3.23 ReadFromCommaSeparatedString()

```
bool gdcm::Tag::ReadFromCommaSeparatedString (
            const char * str)
```

Read from a comma separated string. This is a highly user oriented function, the string should be formatted as↩ : 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

12.314.3.24 ReadFromContinuousString()

```
bool gdcm::Tag::ReadFromContinuousString (
            const char * str)
```

Read From XML formatted tag value eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

12.314.3.25 ReadFromPipeSeparatedString()

```
bool gdcm::Tag::ReadFromPipeSeparatedString (
            const char * str)
```

Read from a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

ReadFromCommaSeparatedString

12.314.3.26 SetElement()

```
void gdcm::Tag::SetElement (
            uint16_t element)  [inline]
```

Sets the 'Element number' of the given Tag.

Examples

DuplicatePCDE.cxx, and PublicDict.cxx.

Referenced by gdcm::PrivateTag::PrivateTag(), gdcm::PrivateTag::PrivateTag(), GetPrivateCreator(), operator>>, and SetPrivateCreator().

12.314.3.27 SetElementTag() [1/2]

```
void gdcm::Tag::SetElementTag (
            uint16_t group,
            uint16_t element)  [inline]
```

Sets the 'Group number' & 'Element number' of the given Tag.

Referenced by Tag(), and gdcm::PrivateTag::operator=().

12.314.3.28 SetElementTag() [2/2]

```
void gdcm::Tag::SetElementTag (
            uint32_t tag)  [inline]
```

Sets the full tag value of the given Tag.

References tag.

12.314.3.29 SetGroup()

```
void gdcm::Tag::SetGroup (
            uint16_t group)  [inline]
```

Sets the 'Group number' of the given Tag.

Referenced by operator>>, and SetPrivateCreator().

12.314.3.30 SetPrivateCreator()

```
void gdcm::Tag::SetPrivateCreator (
            Tag const & t)  [inline]
```

Set private creator:

Examples

DuplicatePCDE.cxx.

References Tag(), GetElement(), GetGroup(), IsPrivate(), SetElement(), and SetGroup().

12.314.3.31 Write()

```
template<typename TSwap>
const std::ostream & gdcm::Tag::Write (
            std::ostream & os) const  [inline]
```

Write a tag in binary rep.

Referenced by gdcm::Item::Write(), gdcm::SequenceOfFragments::Write(), and gdcm::SequenceOfItems::Write().

12.314.4 Friends And Related Symbol Documentation

12.314.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Tag & _val)  [friend]
```

References Tag().

12.314.4.2 operator>>

```
std::istream & operator>> (
            std::istream & _is,
            Tag & _val)  [friend]
```

References Tag(), SetElement(), and SetGroup().

12.314.5 Member Data Documentation

12.314.5.1 bytes

```
char gdcm::Tag::bytes[4]
```

12.314.5.2 tag

```
uint32_t gdcm::Tag::tag
```

Referenced by Tag(), Tag(), operator!=(), operator<(), operator=(), operator==(), and SetElementTag().

12.314.5.3 tags

```
uint16_t gdcm::Tag::tags[2]
```

Referenced by operator<().

The documentation for this class was generated from the following file:

- gdcmTag.h

12.315 gdcm::TagPath Class Reference

class to handle a path of tag.

```
#include <gdcmTagPath.h>
```

Public Member Functions

- TagPath ()
- ~TagPath ()
- bool ConstructFromString (const char *path)
- bool ConstructFromTagList (Tag const *l, unsigned int n)

  Construct from a list of tags.
- void Print (std::ostream &) const
- bool Push (Tag const &t)
- bool Push (unsigned int itemnum)

Static Public Member Functions

- static bool IsValid (const char *path)

  Return if path is valid or not.

12.315.1 Detailed Description

class to handle a path of tag.

Any Resemblance to Existing XPath is Purely Coincidental `ftp://medical.nema.org/medical/dicom/supps/sup118↩_pc.pdf`

12.315.2 Constructor & Destructor Documentation

12.315.2.1 TagPath()

```
gdcm::TagPath::TagPath ()
```

12.315.2.2 ~TagPath()

```
gdcm::TagPath::~TagPath ()
```

12.315.3 Member Function Documentation

12.315.3.1 ConstructFromString()

```
bool gdcm::TagPath::ConstructFromString (
            const char * path)
```

"/0018,0018/"... No space allowed, comma is use to separate tag group from tag element and slash is used to separate tag return false if invalid

12.315.3.2 ConstructFromTagList()

```
bool gdcm::TagPath::ConstructFromTagList (
            Tag const * l,
            unsigned int n)
```

Construct from a list of tags.

12.315.3.3 IsValid()

```
bool gdcm::TagPath::IsValid (
            const char * path)  [static]
```

Return if path is valid or not.

12.315.3.4 Print()

```
void gdcm::TagPath::Print (
            std::ostream & ) const
```

12.315.3.5 Push() [1/2]

```
bool gdcm::TagPath::Push (
            Tag const & t)
```

12.315.3.6 Push() [2/2]

```
bool gdcm::TagPath::Push (
            unsigned int itemnum)
```

The documentation for this class was generated from the following file:

- gdcmTagPath.h

12.316 gdcm::Testing Class Reference

class for testing

```
#include <gdcmTesting.h>
```

Public Types

- typedef const char ∗const (∗ MD5DataImagesType)[2]
- typedef const char ∗const (∗ MediaStorageDataFilesType)[2]

  return the table that map the media storage (as string) of a filename (gdcmData)

Public Member Functions

- Testing ()=default
- ∼Testing ()=default
- void Print (std::ostream &os=std::cout)

  Print.

Static Public Member Functions

- static bool ComputeFileMD5 (const char ∗filename, char digest_str[33])
- static bool ComputeMD5 (const char ∗buffer, size_t buf_len, char digest_str[33])
- static const char ∗ GetDataExtraRoot ()

  Return the GDCM DATA EXTRA ROOT.
- static const char ∗ GetDataRoot ()

  Return the GDCM DATA ROOT.
- static const char ∗ GetFileName (unsigned int file)
- static const char ∗const ∗ GetFileNames ()

  return the table of fullpath to gdcmData DICOM files:
- static int GetLossyFlagFromFile (const char ∗filepath)
- static const char ∗const ∗ GetMD5DataImage (unsigned int file)
- static MD5DataImagesType GetMD5DataImages ()
- static const char ∗ GetMD5FromBrokenFile (const char ∗filepath)
- static const char ∗ GetMD5FromFile (const char ∗filepath)
- static const char ∗const ∗ GetMediaStorageDataFile (unsigned int file)
- static MediaStorageDataFilesType GetMediaStorageDataFiles ()
- static const char ∗ GetMediaStorageFromFile (const char ∗filepath)
- static unsigned int GetNumberOfFileNames ()
- static unsigned int GetNumberOfMD5DataImages ()
- static unsigned int GetNumberOfMediaStorageDataFiles ()
- static const char ∗ GetPixelSpacingDataRoot ()

  Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset).
- static std::streamoff GetSelectedPrivateGroupOffsetFromFile (const char ∗filepath)
- static std::streamoff GetSelectedTagsOffsetFromFile (const char ∗filepath)
- static const char ∗ GetSourceDirectory ()
- static std::streamoff GetStreamOffsetFromFile (const char ∗filepath)

- static const char * GetTempDirectory (const char *subdir=nullptr)
- static const wchar_t * GetTempDirectoryW (const wchar_t *subdir=nullptr)

  NOT THREAD SAFE.
- static const char * GetTempFilename (const char *filename, const char *subdir=nullptr)

  NOT THREAD SAFE.
- static const wchar_t * GetTempFilenameW (const wchar_t *filename, const wchar_t *subdir=nullptr)

  NOT THREAD SAFE.

12.316.1 Detailed Description

class for testing

this class is used for the nightly regression system for GDCM It makes heavily use of md5 computation

See also

gdcm::MD5 class for md5 computation

12.316.2 Member Typedef Documentation

12.316.2.1 MD5DataImagesType

```
typedef const char* const(* gdcm::Testing::MD5DataImagesType)[2]
```

return the table that map the md5 (as in md5sum) of the Pixel Data associated to a filename

12.316.2.2 MediaStorageDataFilesType

```
typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType)[2]
```

return the table that map the media storage (as string) of a filename (gdcmData)

12.316.3 Constructor & Destructor Documentation

12.316.3.1 Testing()

```
gdcm::Testing::Testing ()  [default]
```

12.316.3.2 ~Testing()

```
gdcm::Testing::~Testing ()  [default]
```

12.316.4 Member Function Documentation

12.316.4.1 ComputeFileMD5()

```
bool gdcm::Testing::ComputeFileMD5 (
            const char * filename,
            char digest_str[33])  [static]
```

Examples

MetaImageMD5Activiz.cs.

12.316.4.2 ComputeMD5()

```
bool gdcm::Testing::ComputeMD5 (
            const char * buffer,
            size_t buf_len,
            char digest_str[33])  [static]
```

MD5 stuff digest_str needs to be at least : strlen = [2∗16+1]; string will be \0 padded. (md5 are 32 bytes long) Testing is not meant to be shipped with an installed GDCM release, always prefer the gdcm::MD5 API when doing md5 computation.

12.316.4.3 GetDataExtraRoot()

```
const char * gdcm::Testing::GetDataExtraRoot ()  [static]
```

Return the GDCM DATA EXTRA ROOT.

Examples

DiscriminateVolume.cxx, VolumeSorter.cxx, and reslicesphere.cxx.

12.316.4.4 GetDataRoot()

```
const char * gdcm::Testing::GetDataRoot ()  [static]
```

Return the GDCM DATA ROOT.

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, and MagnifyFile.cxx.

12.316.4.5 GetFileName()

```
const char * gdcm::Testing::GetFileName (
            unsigned int file)  [static]
```

Examples

MetaImageMD5Activiz.cs.

12.316.4.6 GetFileNames()

```
const char *const * gdcm::Testing::GetFileNames ()  [static]
```

return the table of fullpath to gdcmData DICOM files:

Examples

TestReader.cxx.

12.316.4.7 GetLossyFlagFromFile()

```
int gdcm::Testing::GetLossyFlagFromFile (
            const char * filepath)  [static]
```

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

12.316.4.8 GetMD5DataImage()

```
const char *const * gdcm::Testing::GetMD5DataImage (
            unsigned int file)  [static]
```

12.316.4.9 GetMD5DataImages()

```
MD5DataImagesType gdcm::Testing::GetMD5DataImages ()  [static]
```

12.316.4.10 GetMD5FromBrokenFile()

```
const char * gdcm::Testing::GetMD5FromBrokenFile (
            const char * filepath)  [static]
```

Return what should have been the md5 of file 'filepath' This is based on current GDCM implementation to decipher a broken DICOM file.

12.316.4.11 GetMD5FromFile()

```
const char * gdcm::Testing::GetMD5FromFile (
            const char * filepath)  [static]
```

12.316.4.12 GetMediaStorageDataFile()

```
const char *const * gdcm::Testing::GetMediaStorageDataFile (
            unsigned int file)  [static]
```

12.316.4.13 GetMediaStorageDataFiles()

```
MediaStorageDataFilesType gdcm::Testing::GetMediaStorageDataFiles ()  [static]
```

12.316.4.14 GetMediaStorageFromFile()

```
const char * gdcm::Testing::GetMediaStorageFromFile (
            const char * filepath)  [static]
```

Examples

MetaImageMD5Activiz.cs, and TestReader.cxx.

12.316.4.15 GetNumberOfFileNames()

```
unsigned int gdcm::Testing::GetNumberOfFileNames ()  [static]
```

Examples

MetaImageMD5Activiz.cs.

12.316.4.16 GetNumberOfMD5DataImages()

```
unsigned int gdcm::Testing::GetNumberOfMD5DataImages ()  [static]
```

12.316.4.17 GetNumberOfMediaStorageDataFiles()

```
unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ()  [static]
```

12.316.4.18 GetPixelSpacingDataRoot()

```
const char * gdcm::Testing::GetPixelSpacingDataRoot ()  [static]
```

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset).

12.316.4.19 GetSelectedPrivateGroupOffsetFromFile()

```
std::streamoff gdcm::Testing::GetSelectedPrivateGroupOffsetFromFile (
            const char * filepath)  [static]
```

Return the offset just after private attribute (0009,0010,"GEMS_IDEN_01") if found. Otherwise the offset of the next attribute -1 if not found

12.316.4.20 GetSelectedTagsOffsetFromFile()

```
std::streamoff gdcm::Testing::GetSelectedTagsOffsetFromFile (
            const char * filepath)  [static]
```

Return the offset just after Pixel Data Length (7fe0,0000) if found. Otherwise the offset of the very first pixel cell in Pixel Data -1 if not found

12.316.4.21 GetSourceDirectory()

```
const char * gdcm::Testing::GetSourceDirectory ()  [static]
```

12.316.4.22 GetStreamOffsetFromFile()

```
std::streamoff gdcm::Testing::GetStreamOffsetFromFile (
            const char * filepath)  [static]
```

Return the offset of the very first pixel cell in the PixelData -1 if not found

12.316.4.23 GetTempDirectory()

```
const char * gdcm::Testing::GetTempDirectory (
            const char * subdir = nullptr)  [static]
```

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

Examples

MetaImageMD5Activiz.cs.

12.316.4.24 GetTempDirectoryW()

```
const wchar_t * gdcm::Testing::GetTempDirectoryW (
            const wchar_t * subdir = nullptr)  [static]
```

NOT THREAD SAFE.

12.316.4.25 GetTempFilename()

```
const char * gdcm::Testing::GetTempFilename (
            const char * filename,
            const char * subdir = nullptr)  [static]
```

NOT THREAD SAFE.

Examples

MetaImageMD5Activiz.cs.

12.316.4.26 GetTempFilenameW()

```
const wchar_t * gdcm::Testing::GetTempFilenameW (
            const wchar_t * filename,
            const wchar_t * subdir = nullptr)  [static]
```

NOT THREAD SAFE.

12.316.4.27 Print()

```
void gdcm::Testing::Print (
            std::ostream & os = std::cout)
```

Print.

The documentation for this class was generated from the following file:

- gdcmTesting.h

12.317 gdcm::Trace Class Reference

Trace.

```
#include <gdcmTrace.h>
```

Public Member Functions

- Trace ()
- ∼Trace ()

Static Public Member Functions

- static void DebugOff ()
- static void DebugOn ()
- static void ErrorOff ()
- static void ErrorOn ()
- static bool GetDebugFlag ()
- static std::ostream & GetDebugStream ()
- static bool GetErrorFlag ()
- static std::ostream & GetErrorStream ()
- static std::ostream & GetStream ()
- static bool GetWarningFlag ()
- static std::ostream & GetWarningStream ()
- static void SetDebug (bool debug)

  Turn debug messages on (default: false).
- static void SetDebugStream (std::ostream &os)

  Explicitly set the stream which receive Debug messages:
- static void SetError (bool debug)

  Turn error messages on (default: true).
- static void SetErrorStream (std::ostream &os)

  Explicitly set the stream which receive Error messages:
- static void SetStream (std::ostream &os)
- static void SetStreamToFile (const char ∗filename)
- static void SetWarning (bool debug)

  Turn warning messages on (default: true).
- static void SetWarningStream (std::ostream &os)

  Explicitly set the stream which receive Warning messages:
- static void WarningOff ()
- static void WarningOn ()

12.317.1 Detailed Description

Trace.

Debug / Warning and Error are encapsulated in this class by default the Trace class will redirect any debug/warning/error to std::cerr. Unless SetStream was specified with another (open) stream or SetStreamToFile was specified to a writable file on the system.

Warning

All string messages are removed during compilation time when compiled with CMAKE_BUILD_TYPE being set to either:

- Release
- MinSizeRel It is recommended to compile with RelWithDebInfo and/or Debug during prototyping of applications.

Examples

DecompressJPEGFile.cs.

12.317.2 Constructor & Destructor Documentation

12.317.2.1 Trace()

```
gdcm::Trace::Trace ()
```

12.317.2.2 ~Trace()

```
gdcm::Trace::~Trace ()
```

12.317.3 Member Function Documentation

12.317.3.1 DebugOff()

```
void gdcm::Trace::DebugOff () [static]
```

Examples

MetaImageMD5Activiz.cs, and TestReader.cxx.

12.317.3.2 DebugOn()

```
void gdcm::Trace::DebugOn () [static]
```

Examples

CreateFakePET.cxx, DecompressJPEGFile.cs, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

12.317.3.3 ErrorOff()

```
void gdcm::Trace::ErrorOff () [static]
```

Examples

MetaImageMD5Activiz.cs.

12.317.3.4 ErrorOn()

```
void gdcm::Trace::ErrorOn () [static]
```

12.317.3.5 GetDebugFlag()

```
bool gdcm::Trace::GetDebugFlag ()  [static]
```

12.317.3.6 GetDebugStream()

```
std::ostream & gdcm::Trace::GetDebugStream ()  [static]
```

12.317.3.7 GetErrorFlag()

```
bool gdcm::Trace::GetErrorFlag ()  [static]
```

12.317.3.8 GetErrorStream()

```
std::ostream & gdcm::Trace::GetErrorStream ()  [static]
```

12.317.3.9 GetStream()

```
std::ostream & gdcm::Trace::GetStream ()  [static]
```

12.317.3.10 GetWarningFlag()

```
bool gdcm::Trace::GetWarningFlag ()  [static]
```

12.317.3.11 GetWarningStream()

```
std::ostream & gdcm::Trace::GetWarningStream ()  [static]
```

12.317.3.12 SetDebug()

```
void gdcm::Trace::SetDebug (
            bool debug)  [static]
```

Turn debug messages on (default: false).

Examples

DumpToSQLITE3.cxx.

12.317.3.13 SetDebugStream()

```
void gdcm::Trace::SetDebugStream (
            std::ostream & os)  [static]
```

Explicitly set the stream which receive Debug messages:

12.317.3.14 SetError()

```
void gdcm::Trace::SetError (
            bool debug)  [static]
```

Turn error messages on (default: true).

12.317.3.15 SetErrorStream()

```
void gdcm::Trace::SetErrorStream (
            std::ostream & os)  [static]
```

Explicitly set the stream which receive Error messages:

Examples

CStoreQtProgress.cxx.

12.317.3.16 SetStream()

```
void gdcm::Trace::SetStream (
            std::ostream & os)  [static]
```

Explicitly set the ostream for gdcm::Trace to report to This will set the DebugStream, WarningStream and ErrorStream at once:

12.317.3.17 SetStreamToFile()

```
void gdcm::Trace::SetStreamToFile (
            const char * filename)  [static]
```

Explicitly set the filename for gdcm::Trace to report to The file will be created (it will not append to existing file)

12.317.3.18 SetWarning()

```
void gdcm::Trace::SetWarning (
            bool debug)  [static]
```

Turn warning messages on (default: true).

Examples

DumpToSQLITE3.cxx.

12.317.3.19 SetWarningStream()

```
void gdcm::Trace::SetWarningStream (
            std::ostream & os)  [static]
```

Explicitly set the stream which receive Warning messages:

12.317.3.20 WarningOff()

```
void gdcm::Trace::WarningOff ()  [static]
```

Examples

MetaImageMD5Activiz.cs, and TestReader.cxx.

12.317.3.21 WarningOn()

```
void gdcm::Trace::WarningOn ()  [static]
```

Examples

Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

The documentation for this class was generated from the following file:

- gdcmTrace.h

12.318 gdcm::TransferSyntax Class Reference

Class to manipulate Transfer Syntax.

```
#include <gdcmTransferSyntax.h>
```

Public Types

- enum NegociatedType {
  Unknown = 0 ,
  Explicit ,
  Implicit }
- enum TSType {
  ImplicitVRLittleEndian = 0 ,
  ImplicitVRBigEndianPrivateGE ,
  ExplicitVRLittleEndian ,
  DeflatedExplicitVRLittleEndian ,
  ExplicitVRBigEndian ,
  JPEGBaselineProcess1 ,
  JPEGExtendedProcess2_4 ,
  JPEGExtendedProcess3_5 ,
  JPEGSpectralSelectionProcess6_8 ,
  JPEGFullProgressionProcess10_12 ,
  JPEGLosslessProcess14 ,
  JPEGLosslessProcess14_1 ,
  JPEGLSLossless ,
  JPEGLSNearLossless ,
  JPEG2000Lossless ,
  JPEG2000 ,
  JPEG2000Part2Lossless ,
  JPEG2000Part2 ,
  RLELossless ,
  MPEG2MainProfile ,
  ImplicitVRBigEndianACRNEMA ,
  WeirdPapryus ,
  CT_private_ELE ,
  JPIPReferenced ,
  MPEG2MainProfileHighLevel ,
  MPEG4AVCH264HighProfileLevel4_1 ,
  MPEG4AVCH264BDcompatibleHighProfileLevel4_1 ,
  TS_END }

Public Member Functions

- TransferSyntax (TSType type=ImplicitVRLittleEndian)
- bool CanStoreLossy () const
- NegociatedType GetNegociatedType () const
- const char * GetString () const
- SwapCode GetSwapCode () const
- bool IsEncapsulated () const
- bool IsEncoded () const
- bool IsExplicit () const
- bool IsImplicit () const
- bool IsLossless () const
- bool IsLossy () const
- bool IsValid () const
- operator TSType () const

Static Public Member Functions

- static const char * GetTSString (TSType ts)
- static TSType GetTSType (const char *str)

Friends

- std::ostream & operator<< (std::ostream &os, const TransferSyntax &ts)

12.318.1 Detailed Description

Class to manipulate Transfer Syntax.

Note

TRANSFER SYNTAX (Standard and Private): A set of encoding rules that allow Application Entities to unambiguously negotiate the encoding techniques (e.g., Data Element structure, byte ordering, compression) they are able to support, thereby allowing these Application Entities to communicate.

Todo : The implementation is completely retarded -> see gdcm::UIDs for a replacement We need: IsSupported We need preprocess of raw/xml file We need GetFullName()

Need a notion of Private Syntax. As defined in PS 3.5. Section 9.2

See also

UIDs

Examples

BasicImageAnonymizer.cs, CompressLossyJPEG.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, GetJPEGSamplePrecision.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, MpegVideoInfo.cs, and StandardizeFiles.cs.

12.318.2 Member Enumeration Documentation

12.318.2.1 NegociatedType

`enum gdcm::TransferSyntax::NegociatedType`

Enumerator

Unknown	

Explicit	
Implicit	

12.318.2.2 TSType

enum gdcm::TransferSyntax::TSType

Enumerator

ImplicitVRLittleEndian	
ImplicitVRBigEndianPrivateGE	
ExplicitVRLittleEndian	
DeflatedExplicitVRLittleEndian	
ExplicitVRBigEndian	
JPEGBaselineProcess1	
JPEGExtendedProcess2_4	
JPEGExtendedProcess3_5	
JPEGSpectralSelectionProcess6_8	
JPEGFullProgressionProcess10_12	
JPEGLosslessProcess14	
JPEGLosslessProcess14_1	
JPEGLSLossless	
JPEGLSNearLossless	
JPEG2000Lossless	
JPEG2000	
JPEG2000Part2Lossless	
JPEG2000Part2	
RLELossless	
MPEG2MainProfile	
ImplicitVRBigEndianACRNEMA	
WeirdPapryus	
CT_private_ELE	
JPIPReferenced	
MPEG2MainProfileHighLevel	
MPEG4AVCH264HighProfileLevel4_1	

MPEG4AVCH264BDcompatibleHighProfileLevel4↩ _1	
TS_END	

Examples

BasicImageAnonymizer.cs, CompressLossyJPEG.cs, DecompressImageMultiframe.cs, DecompressJPEGFile.cs, FileChangeTS.cs, FileChangeTSLossy.cs, MpegVideoInfo.cs, and StandardizeFiles.cs.

12.318.3 Constructor & Destructor Documentation

12.318.3.1 TransferSyntax()

```
gdcm::TransferSyntax::TransferSyntax (
            TSType type = ImplicitVRLittleEndian)  [inline]
```

References ImplicitVRLittleEndian.

Referenced by operator<<.

12.318.4 Member Function Documentation

12.318.4.1 CanStoreLossy()

```
bool gdcm::TransferSyntax::CanStoreLossy () const
```

return true if TransFer Syntax Allow storing of Lossy Pixel Data

12.318.4.2 GetNegociatedType()

```
NegociatedType gdcm::TransferSyntax::GetNegociatedType () const
```

12.318.4.3 GetString()

```
const char * gdcm::TransferSyntax::GetString () const  [inline]
```

References GetTSString().

12.318.4.4 GetSwapCode()

```
SwapCode gdcm::TransferSyntax::GetSwapCode () const
```

Deprecated Return the SwapCode associated with the Transfer Syntax. Be careful with the special GE private syntax the DataSet is written in little endian but the Pixel Data is in Big Endian.

12.318.4.5 GetTSString()

```
const char * gdcm::TransferSyntax::GetTSString (
            TSType ts)  [static]
```

Examples

LargeVRDSExplicit.cxx.

Referenced by GetString(), and operator<<.

12.318.4.6 GetTSType()

```
TSType gdcm::TransferSyntax::GetTSType (
            const char * str)  [static]
```

12.318.4.7 IsEncapsulated()

```
bool gdcm::TransferSyntax::IsEncapsulated () const
```

Examples

ExtractIconFromFile.cxx.

12.318.4.8 IsEncoded()

```
bool gdcm::TransferSyntax::IsEncoded () const
```

12.318.4.9 IsExplicit()

```
bool gdcm::TransferSyntax::IsExplicit () const
```

12.318.4.10 IsImplicit()

```
bool gdcm::TransferSyntax::IsImplicit () const
```

12.318.4.11 IsLossless()

```
bool gdcm::TransferSyntax::IsLossless () const
```

Return true if the transfer syntax algorithm is a lossless algorithm

12.318.4.12 IsLossy()

```
bool gdcm::TransferSyntax::IsLossy () const
```

Return true if the transfer syntax algorithm is a lossy algorithm

12.318.4.13 IsValid()

```
bool gdcm::TransferSyntax::IsValid () const  [inline]
```

References TS_END.

12.318.4.14 operator TSType()

```
gdcm::TransferSyntax::operator TSType () const  [inline]
```

12.318.5 Friends And Related Symbol Documentation

12.318.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const TransferSyntax & ts)  [friend]
```

References TransferSyntax(), and GetTSString().

The documentation for this class was generated from the following file:

- gdcmTransferSyntax.h

12.319 gdcm::network::TransferSyntaxSub Class Reference

TransferSyntaxSub.

```
#include <gdcmTransferSyntaxSub.h>
```

Public Member Functions

- TransferSyntaxSub ()
- const char * GetName () const
- bool operator== (const TransferSyntaxSub &ts) const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void SetName (const char *name)
- void SetNameFromUID (UIDs::TSName tsname)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.319.1 Detailed Description

TransferSyntaxSub.

Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS

TODO what is the goal of :

Table 9-19 TRANSFER SYNTAX SUB-ITEM FIELDS

12.319.2 Constructor & Destructor Documentation

12.319.2.1 TransferSyntaxSub()

```
gdcm::network::TransferSyntaxSub::TransferSyntaxSub ()
```

Referenced by operator==().

12.319.3 Member Function Documentation

12.319.3.1 GetName()

```
const char * gdcm::network::TransferSyntaxSub::GetName () const  [inline]
```

12.319.3.2 operator==()

```
bool gdcm::network::TransferSyntaxSub::operator== (
            const TransferSyntaxSub & ts) const  [inline]
```

References TransferSyntaxSub().

12.319.3.3 Print()

```
void gdcm::network::TransferSyntaxSub::Print (
            std::ostream & os) const
```

12.319.3.4 Read()

```
std::istream & gdcm::network::TransferSyntaxSub::Read (
            std::istream & is)
```

12.319.3.5 SetName()

```
void gdcm::network::TransferSyntaxSub::SetName (
            const char * name)
```

12.319.3.6 SetNameFromUID()

```
void gdcm::network::TransferSyntaxSub::SetNameFromUID (
            UIDs::TSName tsname)
```

12.319.3.7 Size()

```
size_t gdcm::network::TransferSyntaxSub::Size () const
```

12.319.3.8 Write()

```
const std::ostream & gdcm::network::TransferSyntaxSub::Write (
            std::ostream & os) const
```

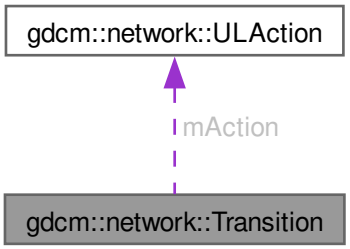
The documentation for this class was generated from the following file:

- gdcmTransferSyntaxSub.h

12.320 gdcm::network::Transition Struct Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::network::Transition:



Public Member Functions

- Transition ()
- Transition (int inEndState, ULAction ∗inAction)
- ∼Transition ()

Static Public Member Functions

- static Transition ∗ MakeNew (int inEndState, ULAction ∗inAction)

Public Attributes

- ULAction ∗ mAction
- int mEnd

12.320.1 Constructor & Destructor Documentation

12.320.1.1 Transition() [1/2]

```
gdcm::network::Transition::Transition () [inline]
```

References gdcm::network::eStaDoesNotExist, mAction, and mEnd.

Referenced by MakeNew().

12.320.1.2 ∼Transition()

```
gdcm::network::Transition::∼Transition () [inline]
```

References mAction.

12.320.1.3 Transition() [2/2]

```
gdcm::network::Transition::Transition (
            int inEndState,
            ULAction * inAction) [inline]
```

References mAction, and mEnd.

12.320.2 Member Function Documentation

12.320.2.1 MakeNew()

```
Transition * gdcm::network::Transition::MakeNew (
            int inEndState,
            ULAction * inAction)  [inline], [static]
```

References Transition().

12.320.3 Member Data Documentation

12.320.3.1 mAction

```
ULAction* gdcm::network::Transition::mAction
```

Referenced by Transition(), Transition(), and ~Transition().

12.320.3.2 mEnd

```
int gdcm::network::Transition::mEnd
```

Referenced by Transition(), and Transition().

The documentation for this struct was generated from the following file:

- gdcmULTransitionTable.h

12.321 gdcm::Type Class Reference

Type.

```
#include <gdcmType.h>
```

Public Types

- enum TypeType {
  T1 = 0 ,
  T1C ,
  T2 ,
  T2C ,
  T3 ,
  UNKNOWN }

Public Member Functions

- Type (TypeType type=UNKNOWN)
- operator TypeType () const

Static Public Member Functions

- static const char * GetTypeString (TypeType type)
- static TypeType GetTypeType (const char *type)

Friends

- std::ostream & operator<< (std::ostream &os, const Type &vr)

12.321.1 Detailed Description

Type.

Note

PS 3.5 7.4 DATA ELEMENT TYPE 7.4.1 TYPE 1 REQUIRED DATA ELEMENTS 7.4.2 TYPE 1C CONDITIONAL DATA ELEMENTS 7.4.3 TYPE 2 REQUIRED DATA ELEMENTS 7.4.4 TYPE 2C CONDITIONAL DATA ELEMENTS 7.4.5 TYPE 3 OPTIONAL DATA ELEMENTS

The intent of Type 2 Data Elements is to allow a zero length to be conveyed when the operator or application does not know its value or has a specific reason for not specifying its value. It is the intent that the device should support these Data Elements.

Examples

TraverseModules.cxx.

12.321.2 Member Enumeration Documentation

12.321.2.1 TypeType

```
enum gdcm::Type::TypeType
```

Enumerator

T1	
T1C	
T2	

T2C	
T3	
UNKNOWN	

12.321.3 Constructor & Destructor Documentation

12.321.3.1 Type()

```
gdcm::Type::Type (
            TypeType type = UNKNOWN)  [inline]
```

References UNKNOWN.

Referenced by operator<<.

12.321.4 Member Function Documentation

12.321.4.1 GetTypeString()

```
const char * gdcm::Type::GetTypeString (
            TypeType type)  [static]
```

Referenced by operator<<.

12.321.4.2 GetTypeType()

```
TypeType gdcm::Type::GetTypeType (
            const char * type)  [static]
```

Referenced by gdcm::ModuleEntry::ModuleEntry().

12.321.4.3 operator TypeType()

```
gdcm::Type::operator TypeType () const  [inline]
```

12.321.5 Friends And Related Symbol Documentation

12.321.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Type & vr)  [friend]
```

References Type(), and GetTypeString().

The documentation for this class was generated from the following file:

- gdcmType.h

12.322 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

Public Attributes

- char Internal [64+1]

Friends

- std::ostream & operator<< (std::ostream &_os, const UI &_val)

12.322.1 Friends And Related Symbol Documentation

12.322.1.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const UI & _val)  [friend]
```

References Internal.

12.322.2 Member Data Documentation

12.322.2.1 Internal

```
char gdcm::UI::Internal[64+1]
```

Referenced by operator<<.

The documentation for this struct was generated from the following file:

- gdcmVR.h

12.323 gdcm::UIDGenerator Class Reference

Class for generating unique UID.

```
#include <gdcmUIDGenerator.h>
```

Public Member Functions

- UIDGenerator ()

  By default the root of a UID is a GDCM Root...
- const char ∗ Generate ()

Static Public Member Functions

- static const char ∗ GetGDCMUID ()

  Return the default (GDCM) root UID:
- static const char ∗ GetRoot ()
- static bool IsValid (const char ∗uid)
- static void SetRoot (const char ∗root)

Static Protected Member Functions

- static bool GenerateUUID (unsigned char ∗uuid_data)

12.323.1 Detailed Description

Class for generating unique UID.

When constructing a Series or Study UID, user *has* to keep around the UID, otherwise the UID Generator will simply forget the value and create a new UID.

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, ManipulateFile.cs, MpegVideoInfo.cs, ReformatFile.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, and uid_unique.cxx.

12.323.2 Constructor & Destructor Documentation

12.323.2.1 UIDGenerator()

```
gdcm::UIDGenerator::UIDGenerator () [inline]
```

By default the root of a UID is a GDCM Root...

12.323.3 Member Function Documentation

12.323.3.1 Generate()

```
const char * gdcm::UIDGenerator::Generate ()
```

Internally uses a std::string, so two calls have the same pointer ! save into a std::string In summary do not write code like that: const char ∗uid1 = uid.Generate(); const char ∗uid2 = uid.Generate(); since uid1 == uid2

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, ManipulateFile.cs, ReformatFile.cs, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, and uid_unique.cxx.

12.323.3.2 GenerateUUID()

```
bool gdcm::UIDGenerator::GenerateUUID (
            unsigned char * uuid_data)  [static], [protected]
```

12.323.3.3 GetGDCMUID()

```
const char * gdcm::UIDGenerator::GetGDCMUID ()  [static]
```

Return the default (GDCM) root UID:

12.323.3.4 GetRoot()

```
const char * gdcm::UIDGenerator::GetRoot ()  [static]
```

Examples

ReformatFile.cs, and StandardizeFiles.cs.

12.323.3.5 IsValid()

```
bool gdcm::UIDGenerator::IsValid (
            const char * uid)  [static]
```

Find out if the string is a valid UID or not

Todo : Move that in DataStructureAndEncoding (see FileMetaInformation::CheckFileMetaInformation)

12.323.3.6 SetRoot()

```
void gdcm::UIDGenerator::SetRoot (
            const char * root)  [static]
```

The current implementation in GDCM make use of the UUID implementation (RFC 4122) and has been successfully been tested for a root of size 26 bytes. Any longer root should work (the Generate() function will return a string), but will truncate the high bits of the 128bits UUID until the generated string fits on 64 bits. The authors disclaims any responsabitlity for guaranteeing uniqueness of UIDs when the root is longer than 26 bytes.

Examples

ReformatFile.cs, StandardizeFiles.cs, and uid_unique.cxx.

The documentation for this class was generated from the following file:

- gdcmUIDGenerator.h

12.324 gdcm::UIDs Class Reference

all known uids

```
#include <gdcmUIDs.h>
```

Public Types

- typedef const char *const (* TransferSyntaxStringsType)[2]
- enum TSName {
  VerificationSOPClass = 1 ,
  ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM = 2 ,
  ExplicitVRLittleEndian = 3 ,
  DeflatedExplicitVRLittleEndian = 4 ,
  ExplicitVRBigEndian = 5 ,
  JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression = 6 ,
  JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only = 7 ,
  JPEGExtendedProcess35Retired = 8 ,
  JPEGSpectralSelectionNonHierarchicalProcess68Retired = 9 ,
  JPEGSpectralSelectionNonHierarchicalProcess79Retired = 10 ,
  JPEGFullProgressionNonHierarchicalProcess1012Retired = 11 ,
  JPEGFullProgressionNonHierarchicalProcess1113Retired = 12 ,
  JPEGLosslessNonHierarchicalProcess14 = 13 ,
  JPEGLosslessNonHierarchicalProcess15Retired = 14 ,
  JPEGExtendedHierarchicalProcess1618Retired = 15 ,
  JPEGExtendedHierarchicalProcess1719Retired = 16 ,
  JPEGSpectralSelectionHierarchicalProcess2022Retired = 17 ,
  JPEGSpectralSelectionHierarchicalProcess2123Retired = 18 ,
  JPEGFullProgressionHierarchicalProcess2426Retired = 19 ,
  JPEGFullProgressionHierarchicalProcess2527Retired = 20 ,
  JPEGLosslessHierarchicalProcess28Retired = 21 ,

JPEGLosslessHierarchicalProcess29Retired = 22 ,
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompr
= 23 ,
JPEGLSLosslessImageCompression = 24 ,
JPEGLSLossyNearLosslessImageCompression = 25 ,
JPEG2000ImageCompressionLosslessOnly = 26 ,
JPEG2000ImageCompression = 27 ,
JPEG2000Part2MulticomponentImageCompressionLosslessOnly = 28 ,
JPEG2000Part2MulticomponentImageCompression = 29 ,
JPIPReferenced = 30 ,
JPIPReferencedDeflate = 31 ,
MPEG2MainProfileMainLevel = 32 ,
RLELossless = 33 ,
RFC2557MIMEencapsulation = 34 ,
XMLEncoding = 35 ,
MediaStorageDirectoryStorage = 36 ,
TalairachBrainAtlasFrameofReference = 37 ,
SPM2T1FrameofReference = 38 ,
SPM2T2FrameofReference = 39 ,
SPM2PDFrameofReference = 40 ,
SPM2EPIFrameofReference = 41 ,
SPM2FILT1FrameofReference = 42 ,
SPM2PETFrameofReference = 43 ,
SPM2TRANSMFrameofReference = 44 ,
SPM2SPECTFrameofReference = 45 ,
SPM2GRAYFrameofReference = 46 ,
SPM2WHITEFrameofReference = 47 ,
SPM2CSFFrameofReference = 48 ,
SPM2BRAINMASKFrameofReference = 49 ,
SPM2AVG305T1FrameofReference = 50 ,
SPM2AVG152T1FrameofReference = 51 ,
SPM2AVG152T2FrameofReference = 52 ,
SPM2AVG152PDFrameofReference = 53 ,
SPM2SINGLESUBJT1FrameofReference = 54 ,
ICBM452T1FrameofReference = 55 ,
ICBMSingleSubjectMRIFrameofReference = 56 ,
BasicStudyContentNotificationSOPClassRetired = 57 ,
StorageCommitmentPushModelSOPClass = 58 ,
StorageCommitmentPushModelSOPInstance = 59 ,
StorageCommitmentPullModelSOPClassRetired = 60 ,
StorageCommitmentPullModelSOPInstanceRetired = 61 ,
ProceduralEventLoggingSOPClass = 62 ,
ProceduralEventLoggingSOPInstance = 63 ,
SubstanceAdministrationLoggingSOPClass = 64 ,
SubstanceAdministrationLoggingSOPInstance = 65 ,
DICOMUIDRegistry = 66 ,
DICOMControlledTerminology = 67 ,
DICOMApplicationContextName = 68 ,
DetachedPatientManagementSOPClassRetired = 69 ,
DetachedPatientManagementMetaSOPClassRetired = 70 ,
DetachedVisitManagementSOPClassRetired = 71 ,
DetachedStudyManagementSOPClassRetired = 72 ,
StudyComponentManagementSOPClassRetired = 73 ,
ModalityPerformedProcedureStepSOPClass = 74 ,

ModalityPerformedProcedureStepRetrieveSOPClass = 75 ,
ModalityPerformedProcedureStepNotificationSOPClass = 76 ,
DetachedResultsManagementSOPClassRetired = 77 ,
DetachedResultsManagementMetaSOPClassRetired = 78 ,
DetachedStudyManagementMetaSOPClassRetired = 79 ,
DetachedInterpretationManagementSOPClassRetired = 80 ,
StorageServiceClass = 81 ,
BasicFilmSessionSOPClass = 82 ,
BasicFilmBoxSOPClass = 83 ,
BasicGrayscaleImageBoxSOPClass = 84 ,
BasicColorImageBoxSOPClass = 85 ,
ReferencedImageBoxSOPClassRetired = 86 ,
BasicGrayscalePrintManagementMetaSOPClass = 87 ,
ReferencedGrayscalePrintManagementMetaSOPClassRetired = 88 ,
PrintJobSOPClass = 89 ,
BasicAnnotationBoxSOPClass = 90 ,
PrinterSOPClass = 91 ,
PrinterConfigurationRetrievalSOPClass = 92 ,
PrinterSOPInstance = 93 ,
PrinterConfigurationRetrievalSOPInstance = 94 ,
BasicColorPrintManagementMetaSOPClass = 95 ,
ReferencedColorPrintManagementMetaSOPClassRetired = 96 ,
VOILUTBoxSOPClass = 97 ,
PresentationLUTSOPClass = 98 ,
ImageOverlayBoxSOPClassRetired = 99 ,
BasicPrintImageOverlayBoxSOPClassRetired = 100 ,
PrintQueueSOPInstanceRetired = 101 ,
PrintQueueManagementSOPClassRetired = 102 ,
StoredPrintStorageSOPClassRetired = 103 ,
HardcopyGrayscaleImageStorageSOPClassRetired = 104 ,
HardcopyColorImageStorageSOPClassRetired = 105 ,
PullPrintRequestSOPClassRetired = 106 ,
PullStoredPrintManagementMetaSOPClassRetired = 107 ,
MediaCreationManagementSOPClassUID = 108 ,
ComputedRadiographyImageStorage = 109 ,
DigitalXRayImageStorageForPresentation = 110 ,
DigitalXRayImageStorageForProcessing = 111 ,
DigitalMammographyXRayImageStorageForPresentation = 112 ,
DigitalMammographyXRayImageStorageForProcessing = 113 ,
DigitalIntraoralXRayImageStorageForPresentation = 114 ,
DigitalIntraoralXRayImageStorageForProcessing = 115 ,
CTImageStorage = 116 ,
EnhancedCTImageStorage = 117 ,
UltrasoundMultiframeImageStorageRetired = 118 ,
UltrasoundMultiframeImageStorage = 119 ,
MRImageStorage = 120 ,
EnhancedMRImageStorage = 121 ,
MRSpectroscopyStorage = 122 ,
NuclearMedicineImageStorageRetired = 123 ,
UltrasoundImageStorageRetired = 124 ,
UltrasoundImageStorage = 125 ,
SecondaryCaptureImageStorage = 126 ,
MultiframeSingleBitSecondaryCaptureImageStorage = 127 ,
MultiframeGrayscaleByteSecondaryCaptureImageStorage = 128 ,

MultiframeGrayscaleWordSecondaryCaptureImageStorage = 129 ,
MultiframeTrueColorSecondaryCaptureImageStorage = 130 ,
StandaloneOverlayStorageRetired = 131 ,
StandaloneCurveStorageRetired = 132 ,
WaveformStorageTrialRetired = 133 ,
ECG12leadWaveformStorage = 134 ,
GeneralECGWaveformStorage = 135 ,
AmbulatoryECGWaveformStorage = 136 ,
HemodynamicWaveformStorage = 137 ,
CardiacElectrophysiologyWaveformStorage = 138 ,
BasicVoiceAudioWaveformStorage = 139 ,
StandaloneModalityLUTStorageRetired = 140 ,
StandaloneVOILUTStorageRetired = 141 ,
GrayscaleSoftcopyPresentationStateStorageSOPClass = 142 ,
ColorSoftcopyPresentationStateStorageSOPClass = 143 ,
PseudoColorSoftcopyPresentationStateStorageSOPClass = 144 ,
BlendingSoftcopyPresentationStateStorageSOPClass = 145 ,
XRayAngiographicImageStorage = 146 ,
EnhancedXAImageStorage = 147 ,
XRayRadiofluoroscopicImageStorage = 148 ,
EnhancedXRFImageStorage = 149 ,
XRay3DAngiographicImageStorage = 150 ,
XRay3DCraniofacialImageStorage = 151 ,
XRayAngiographicBiPlaneImageStorageRetired = 152 ,
NuclearMedicineImageStorage = 153 ,
RawDataStorage = 154 ,
SpatialRegistrationStorage = 155 ,
SpatialFiducialsStorage = 156 ,
DeformableSpatialRegistrationStorage = 157 ,
SegmentationStorage = 158 ,
RealWorldValueMappingStorage = 159 ,
VLImageStorageTrialRetired = 160 ,
VLMultiframeImageStorageTrialRetired = 161 ,
VLEndoscopicImageStorage = 162 ,
VideoEndoscopicImageStorage = 163 ,
VLMicroscopicImageStorage = 164 ,
VideoMicroscopicImageStorage = 165 ,
VLSlideCoordinatesMicroscopicImageStorage = 166 ,
VLPhotographicImageStorage = 167 ,
VideoPhotographicImageStorage = 168 ,
OphthalmicPhotography8BitImageStorage = 169 ,
OphthalmicPhotography16BitImageStorage = 170 ,
StereometricRelationshipStorage = 171 ,
OphthalmicTomographyImageStorage = 172 ,
TextSRStorageTrialRetired = 173 ,
AudioSRStorageTrialRetired = 174 ,
DetailSRStorageTrialRetired = 175 ,
ComprehensiveSRStorageTrialRetired = 176 ,
BasicTextSRStorage = 177 ,
EnhancedSRStorage = 178 ,
ComprehensiveSRStorage = 179 ,
ProcedureLogStorage = 180 ,
MammographyCADSRStorage = 181 ,
KeyObjectSelectionDocumentStorage = 182 ,

ChestCADSRStorage = 183 ,
XRayRadiationDoseSRStorage = 184 ,
EncapsulatedPDFStorage = 185 ,
EncapsulatedCDAStorage = 186 ,
PositronEmissionTomographyImageStorage = 187 ,
StandalonePETCurveStorageRetired = 188 ,
RTImageStorage = 189 ,
RTDoseStorage = 190 ,
RTStructureSetStorage = 191 ,
RTBeamsTreatmentRecordStorage = 192 ,
RTPlanStorage = 193 ,
RTBrachyTreatmentRecordStorage = 194 ,
RTTreatmentSummaryRecordStorage = 195 ,
RTIonPlanStorage = 196 ,
RTIonBeamsTreatmentRecordStorage = 197 ,
PatientRootQueryRetrieveInformationModelFIND = 198 ,
PatientRootQueryRetrieveInformationModelMOVE = 199 ,
PatientRootQueryRetrieveInformationModelGET = 200 ,
StudyRootQueryRetrieveInformationModelFIND = 201 ,
StudyRootQueryRetrieveInformationModelMOVE = 202 ,
StudyRootQueryRetrieveInformationModelGET = 203 ,
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired = 204 ,
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired = 205 ,
PatientStudyOnlyQueryRetrieveInformationModelGETRetired = 206 ,
ModalityWorklistInformationModelFIND = 207 ,
GeneralPurposeWorklistInformationModelFIND = 208 ,
GeneralPurposeScheduledProcedureStepSOPClass = 209 ,
GeneralPurposePerformedProcedureStepSOPClass = 210 ,
GeneralPurposeWorklistManagementMetaSOPClass = 211 ,
InstanceAvailabilityNotificationSOPClass = 212 ,
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft = 213 ,
RTConventionalMachineVerificationSupplement74FrozenDraft = 214 ,
RTIonMachineVerificationSupplement74FrozenDraft = 215 ,
UnifiedWorklistandProcedureStepServiceClass = 216 ,
UnifiedProcedureStepPushSOPClass = 217 ,
UnifiedProcedureStepWatchSOPClass = 218 ,
UnifiedProcedureStepPullSOPClass = 219 ,
UnifiedProcedureStepEventSOPClass = 220 ,
UnifiedWorklistandProcedureStepSOPInstance = 221 ,
GeneralRelevantPatientInformationQuery = 222 ,
BreastImagingRelevantPatientInformationQuery = 223 ,
CardiacRelevantPatientInformationQuery = 224 ,
HangingProtocolStorage = 225 ,
HangingProtocolInformationModelFIND = 226 ,
HangingProtocolInformationModelMOVE = 227 ,
ProductCharacteristicsQuerySOPClass = 228 ,
SubstanceApprovalQuerySOPClass = 229 ,
dicomDeviceName = 230 ,
dicomDescription = 231 ,
dicomManufacturer = 232 ,
dicomManufacturerModelName = 233 ,
dicomSoftwareVersion = 234 ,
dicomVendorData = 235 ,
dicomAETitle = 236 ,

dicomNetworkConnectionReference = 237 ,
dicomApplicationCluster = 238 ,
dicomAssociationInitiator = 239 ,
dicomAssociationAcceptor = 240 ,
dicomHostname = 241 ,
dicomPort = 242 ,
dicomSOPClass = 243 ,
dicomTransferRole = 244 ,
dicomTransferSyntax = 245 ,
dicomPrimaryDeviceType = 246 ,
dicomRelatedDeviceReference = 247 ,
dicomPreferredCalledAETitle = 248 ,
dicomTLSCyphersuite = 249 ,
dicomAuthorizedNodeCertificateReference = 250 ,
dicomThisNodeCertificateReference = 251 ,
dicomInstalled = 252 ,
dicomStationName = 253 ,
dicomDeviceSerialNumber = 254 ,
dicomInstitutionName = 255 ,
dicomInstitutionAddress = 256 ,
dicomInstitutionDepartmentName = 257 ,
dicomIssuerOfPatientID = 258 ,
dicomPreferredCallingAETitle = 259 ,
dicomSupportedCharacterSet = 260 ,
dicomConfigurationRoot = 261 ,
dicomDevicesRoot = 262 ,
dicomUniqueAETitlesRegistryRoot = 263 ,
dicomDevice = 264 ,
dicomNetworkAE = 265 ,
dicomNetworkConnection = 266 ,
dicomUniqueAETitle = 267 ,
dicomTransferCapability = 268 ,
VLWholeSlideMicroscopyImageStorage = 269 ,
EnhancedUSVolumeStorage = 270 ,
SurfaceSegmentationStorage = 271 ,
BreastTomosynthesisImageStorage = 272 ,
LegacyConvertedEnhancedCTImageStorage = 273 ,
LegacyConvertedEnhancedMRImageStorage = 274 ,
LegacyConvertedEnhancedPETImageStorage = 275 ,
MPEG2MainProfileHighLevel = 276 ,
MPEG4AVCH_264HighProfileLevel4_1 = 277 ,
MPEG4AVCH_264BDcompatibleHighProfileLevel4_1 = 278 ,
PETColorPaletteSOPInstance = 279 ,
HotMetalBlueColorPaletteSOPInstance = 280 ,
PET20StepColorPaletteSOPInstance = 281 ,
SpringColorPaletteSOPInstance = 282 ,
SummerColorPaletteSOPInstance = 283 ,
FallColorPaletteSOPInstance = 284 ,
WinterColorPaletteSOPInstance = 285 ,
Papyrus3ImplicitVRLittleEndian = 286 ,
AdultMouseAnatomyOntology = 287 ,
UberonOntology = 288 ,
IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN = 289 ,
MouseGenomeInitiativeMGI = 290 ,

PubChemCompoundCID = 291 ,
ICD11 = 292 ,
NewYorkUniversityMelanomaClinicalCooperativeGroup = 293 ,
MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide = 294 ,
ImageBiomarkerStandardisationInitiative = 295 ,
RadiomicsOntology = 296 ,
DisplaySystemSOPClass = 297 ,
DisplaySystemSOPInstance = 298 ,
GeneralAudioWaveformStorage = 299 ,
ArterialPulseWaveformStorage = 300 ,
RespiratoryWaveformStorage = 301 ,
XAXRFGrayscaleSoftcopyPresentationStateStorage = 302 ,
GrayscalePlanarMPRVolumetricPresentationStateStorage = 303 ,
MPEG4AVCH_264HighProfileLevel4_2For2DVideo = 304 ,
MPEG4AVCH_264HighProfileLevel4_2For3DVideo = 305 ,
MPEG4AVCH_264StereoHighProfileLevel4_2 = 306 ,
HEVCH_265MainProfileLevel5_1 = 307 ,
HEVCH_265Main10ProfileLevel5_1 = 308 ,
HotIronColorPaletteSOPInstance = 309 ,
CompositingPlanarMPRVolumetricPresentationStateStorage = 310 ,
AdvancedBlendingPresentationStateStorage = 311 ,
VolumeRenderingVolumetricPresentationStateStorage = 312 ,
SegmentedVolumeRenderingVolumetricPresentationStateStorage = 313 ,
MultipleVolumeRenderingVolumetricPresentationStateStorage = 314 ,
Null0 = 315 ,
BreastProjectionXRayImageStorageForPresentation = 316 ,
BreastProjectionXRayImageStorageForProcessing = 317 ,
IntravascularOpticalCoherenceTomographyImageStorageForPresentation = 318 ,
IntravascularOpticalCoherenceTomographyImageStorageForProcessing = 319 ,
ParametricMapStorage = 320 ,
Null1 = 321 ,
TractographyResultsStorage = 322 ,
SurfaceScanMeshStorage = 323 ,
SurfaceScanPointCloudStorage = 324 ,
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage = 325 ,
WideFieldOphthalmicPhotography3DCoordinatesImageStorage = 326 ,
OphthalmicOpticalCoherenceTomographyEnFaceImageStorage = 327 ,
OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage = 328 ,
LensometryMeasurementsStorage = 329 ,
AutorefractionMeasurementsStorage = 330 ,
KeratometryMeasurementsStorage = 331 ,
SubjectiveRefractionMeasurementsStorage = 332 ,
VisualAcuityMeasurementsStorage = 333 ,
SpectaclePrescriptionReportStorage = 334 ,
OphthalmicAxialMeasurementsStorage = 335 ,
IntraocularLensCalculationsStorage = 336 ,
MacularGridThicknessandVolumeReportStorage = 337 ,
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage = 338 ,
OphthalmicThicknessMapStorage = 339 ,
CornealTopographyMapStorage = 340 ,
Comprehensive3DSRStorage = 341 ,
ExtensibleSRStorage = 342 ,
RadiopharmaceuticalRadiationDoseSRStorage = 343 ,
ColonCADSRStorage = 344 ,

ImplantationPlanSRStorage = 345 ,
AcquisitionContextSRStorage = 346 ,
SimplifiedAdultEchoSRStorage = 347 ,
PatientRadiationDoseSRStorage = 348 ,
PlannedImagingAgentAdministrationSRStorage = 349 ,
PerformedImagingAgentAdministrationSRStorage = 350 ,
ContentAssessmentResultsStorage = 351 ,
EncapsulatedSTLStorage = 352 ,
EnhancedPETImageStorage = 353 ,
BasicStructuredDisplayStorage = 354 ,
CTDefinedProcedureProtocolStorage = 355 ,
CTPerformedProcedureProtocolStorage = 356 ,
ProtocolApprovalStorage = 357 ,
ProtocolApprovalInformationModelFIND = 358 ,
ProtocolApprovalInformationModelMOVE = 359 ,
ProtocolApprovalInformationModelGET = 360 ,
RTPhysicianIntentStorage = 361 ,
RTSegmentAnnotationStorage = 362 ,
DICOSCTImageStorage = 363 ,
DICOSDigitalXRayImageStorageForPresentation = 364 ,
DICOSDigitalXRayImageStorageForProcessing = 365 ,
DICOSThreatDetectionReportStorage = 366 ,
DICOS2DAITStorage = 367 ,
DICOS3DAITStorage = 368 ,
DICOSQuadrupoleResonanceQRStorage = 369 ,
EddyCurrentImageStorage = 370 ,
EddyCurrentMultiframeImageStorage = 371 ,
CompositeInstanceRootRetrieveMOVE = 372 ,
CompositeInstanceRootRetrieveGET = 373 ,
CompositeInstanceRetrieveWithoutBulkDataGET = 374 ,
DefinedProcedureProtocolInformationModelFIND = 375 ,
DefinedProcedureProtocolInformationModelMOVE = 376 ,
DefinedProcedureProtocolInformationModelGET = 377 ,
UPSFilteredGlobalSubscriptionSOPInstance = 378 ,
UnifiedWorklistandProcedureStepServiceClass1 = 379 ,
UnifiedProcedureStepPushSOPClass1 = 380 ,
UnifiedProcedureStepWatchSOPClass1 = 381 ,
UnifiedProcedureStepPullSOPClass1 = 382 ,
UnifiedProcedureStepEventSOPClass1 = 383 ,
RTBeamsDeliveryInstructionStorage = 384 ,
RTConventionalMachineVerification = 385 ,
RTIonMachineVerification = 386 ,
RTBrachyApplicationSetupDeliveryInstructionStorage = 387 ,
HangingProtocolInformationModelGET = 388 ,
ColorPaletteStorage = 389 ,
ColorPaletteQueryRetrieveInformationModelFIND = 390 ,
ColorPaletteQueryRetrieveInformationModelMOVE = 391 ,
ColorPaletteQueryRetrieveInformationModelGET = 392 ,
GenericImplantTemplateStorage = 393 ,
GenericImplantTemplateInformationModelFIND = 394 ,
GenericImplantTemplateInformationModelMOVE = 395 ,
GenericImplantTemplateInformationModelGET = 396 ,
ImplantAssemblyTemplateStorage = 397 ,
ImplantAssemblyTemplateInformationModelFIND = 398 ,

ImplantAssemblyTemplateInformationModelMOVE = 399 ,
ImplantAssemblyTemplateInformationModelGET = 400 ,
ImplantTemplateGroupStorage = 401 ,
ImplantTemplateGroupInformationModelFIND = 402 ,
ImplantTemplateGroupInformationModelMOVE = 403 ,
ImplantTemplateGroupInformationModelGET = 404 ,
NativeDICOMModel = 405 ,
AbstractMultiDimensionalImageModel = 406 ,
DICOMContentMappingResource = 407 ,
EnhancedMRColorImageStorage = 408 ,
UniversalCoordinatedTime = 409 }

- enum TSType {
uid_1_2_840_10008_1_1 = 1 ,
uid_1_2_840_10008_1_2 = 2 ,
uid_1_2_840_10008_1_2_1 = 3 ,
uid_1_2_840_10008_1_2_1_99 = 4 ,
uid_1_2_840_10008_1_2_2 = 5 ,
uid_1_2_840_10008_1_2_4_50 = 6 ,
uid_1_2_840_10008_1_2_4_51 = 7 ,
uid_1_2_840_10008_1_2_4_52 = 8 ,
uid_1_2_840_10008_1_2_4_53 = 9 ,
uid_1_2_840_10008_1_2_4_54 = 10 ,
uid_1_2_840_10008_1_2_4_55 = 11 ,
uid_1_2_840_10008_1_2_4_56 = 12 ,
uid_1_2_840_10008_1_2_4_57 = 13 ,
uid_1_2_840_10008_1_2_4_58 = 14 ,
uid_1_2_840_10008_1_2_4_59 = 15 ,
uid_1_2_840_10008_1_2_4_60 = 16 ,
uid_1_2_840_10008_1_2_4_61 = 17 ,
uid_1_2_840_10008_1_2_4_62 = 18 ,
uid_1_2_840_10008_1_2_4_63 = 19 ,
uid_1_2_840_10008_1_2_4_64 = 20 ,
uid_1_2_840_10008_1_2_4_65 = 21 ,
uid_1_2_840_10008_1_2_4_66 = 22 ,
uid_1_2_840_10008_1_2_4_70 = 23 ,
uid_1_2_840_10008_1_2_4_80 = 24 ,
uid_1_2_840_10008_1_2_4_81 = 25 ,
uid_1_2_840_10008_1_2_4_90 = 26 ,
uid_1_2_840_10008_1_2_4_91 = 27 ,
uid_1_2_840_10008_1_2_4_92 = 28 ,
uid_1_2_840_10008_1_2_4_93 = 29 ,
uid_1_2_840_10008_1_2_4_94 = 30 ,
uid_1_2_840_10008_1_2_4_95 = 31 ,
uid_1_2_840_10008_1_2_4_100 = 32 ,
uid_1_2_840_10008_1_2_5 = 33 ,
uid_1_2_840_10008_1_2_6_1 = 34 ,
uid_1_2_840_10008_1_2_6_2 = 35 ,
uid_1_2_840_10008_1_3_10 = 36 ,
uid_1_2_840_10008_1_4_1_1 = 37 ,
uid_1_2_840_10008_1_4_1_2 = 38 ,
uid_1_2_840_10008_1_4_1_3 = 39 ,
uid_1_2_840_10008_1_4_1_4 = 40 ,
uid_1_2_840_10008_1_4_1_5 = 41 ,
uid_1_2_840_10008_1_4_1_6 = 42 ,

uid_1_2_840_10008_1_4_1_7 = 43 ,
uid_1_2_840_10008_1_4_1_8 = 44 ,
uid_1_2_840_10008_1_4_1_9 = 45 ,
uid_1_2_840_10008_1_4_1_10 = 46 ,
uid_1_2_840_10008_1_4_1_11 = 47 ,
uid_1_2_840_10008_1_4_1_12 = 48 ,
uid_1_2_840_10008_1_4_1_13 = 49 ,
uid_1_2_840_10008_1_4_1_14 = 50 ,
uid_1_2_840_10008_1_4_1_15 = 51 ,
uid_1_2_840_10008_1_4_1_16 = 52 ,
uid_1_2_840_10008_1_4_1_17 = 53 ,
uid_1_2_840_10008_1_4_1_18 = 54 ,
uid_1_2_840_10008_1_4_2_1 = 55 ,
uid_1_2_840_10008_1_4_2_2 = 56 ,
uid_1_2_840_10008_1_9 = 57 ,
uid_1_2_840_10008_1_20_1 = 58 ,
uid_1_2_840_10008_1_20_1_1 = 59 ,
uid_1_2_840_10008_1_20_2 = 60 ,
uid_1_2_840_10008_1_20_2_1 = 61 ,
uid_1_2_840_10008_1_40 = 62 ,
uid_1_2_840_10008_1_40_1 = 63 ,
uid_1_2_840_10008_1_42 = 64 ,
uid_1_2_840_10008_1_42_1 = 65 ,
uid_1_2_840_10008_2_6_1 = 66 ,
uid_1_2_840_10008_2_16_4 = 67 ,
uid_1_2_840_10008_3_1_1_1 = 68 ,
uid_1_2_840_10008_3_1_2_1_1 = 69 ,
uid_1_2_840_10008_3_1_2_1_4 = 70 ,
uid_1_2_840_10008_3_1_2_2_1 = 71 ,
uid_1_2_840_10008_3_1_2_3_1 = 72 ,
uid_1_2_840_10008_3_1_2_3_2 = 73 ,
uid_1_2_840_10008_3_1_2_3_3 = 74 ,
uid_1_2_840_10008_3_1_2_3_4 = 75 ,
uid_1_2_840_10008_3_1_2_3_5 = 76 ,
uid_1_2_840_10008_3_1_2_5_1 = 77 ,
uid_1_2_840_10008_3_1_2_5_4 = 78 ,
uid_1_2_840_10008_3_1_2_5_5 = 79 ,
uid_1_2_840_10008_3_1_2_6_1 = 80 ,
uid_1_2_840_10008_4_2 = 81 ,
uid_1_2_840_10008_5_1_1_1 = 82 ,
uid_1_2_840_10008_5_1_1_2 = 83 ,
uid_1_2_840_10008_5_1_1_4 = 84 ,
uid_1_2_840_10008_5_1_1_4_1 = 85 ,
uid_1_2_840_10008_5_1_1_4_2 = 86 ,
uid_1_2_840_10008_5_1_1_9 = 87 ,
uid_1_2_840_10008_5_1_1_9_1 = 88 ,
uid_1_2_840_10008_5_1_1_14 = 89 ,
uid_1_2_840_10008_5_1_1_15 = 90 ,
uid_1_2_840_10008_5_1_1_16 = 91 ,
uid_1_2_840_10008_5_1_1_16_376 = 92 ,
uid_1_2_840_10008_5_1_1_17 = 93 ,
uid_1_2_840_10008_5_1_1_17_376 = 94 ,
uid_1_2_840_10008_5_1_1_18 = 95 ,
uid_1_2_840_10008_5_1_1_18_1 = 96 ,

uid_1_2_840_10008_5_1_1_22 = 97 ,
uid_1_2_840_10008_5_1_1_23 = 98 ,
uid_1_2_840_10008_5_1_1_24 = 99 ,
uid_1_2_840_10008_5_1_1_24_1 = 100 ,
uid_1_2_840_10008_5_1_1_25 = 101 ,
uid_1_2_840_10008_5_1_1_26 = 102 ,
uid_1_2_840_10008_5_1_1_27 = 103 ,
uid_1_2_840_10008_5_1_1_29 = 104 ,
uid_1_2_840_10008_5_1_1_30 = 105 ,
uid_1_2_840_10008_5_1_1_31 = 106 ,
uid_1_2_840_10008_5_1_1_32 = 107 ,
uid_1_2_840_10008_5_1_1_33 = 108 ,
uid_1_2_840_10008_5_1_4_1_1_1 = 109 ,
uid_1_2_840_10008_5_1_4_1_1_1_1 = 110 ,
uid_1_2_840_10008_5_1_4_1_1_1_1_1 = 111 ,
uid_1_2_840_10008_5_1_4_1_1_1_2 = 112 ,
uid_1_2_840_10008_5_1_4_1_1_1_2_1 = 113 ,
uid_1_2_840_10008_5_1_4_1_1_1_3 = 114 ,
uid_1_2_840_10008_5_1_4_1_1_1_3_1 = 115 ,
uid_1_2_840_10008_5_1_4_1_1_2 = 116 ,
uid_1_2_840_10008_5_1_4_1_1_2_1 = 117 ,
uid_1_2_840_10008_5_1_4_1_1_3 = 118 ,
uid_1_2_840_10008_5_1_4_1_1_3_1 = 119 ,
uid_1_2_840_10008_5_1_4_1_1_4 = 120 ,
uid_1_2_840_10008_5_1_4_1_1_4_1 = 121 ,
uid_1_2_840_10008_5_1_4_1_1_4_2 = 122 ,
uid_1_2_840_10008_5_1_4_1_1_5 = 123 ,
uid_1_2_840_10008_5_1_4_1_1_6 = 124 ,
uid_1_2_840_10008_5_1_4_1_1_6_1 = 125 ,
uid_1_2_840_10008_5_1_4_1_1_7 = 126 ,
uid_1_2_840_10008_5_1_4_1_1_7_1 = 127 ,
uid_1_2_840_10008_5_1_4_1_1_7_2 = 128 ,
uid_1_2_840_10008_5_1_4_1_1_7_3 = 129 ,
uid_1_2_840_10008_5_1_4_1_1_7_4 = 130 ,
uid_1_2_840_10008_5_1_4_1_1_8 = 131 ,
uid_1_2_840_10008_5_1_4_1_1_9 = 132 ,
uid_1_2_840_10008_5_1_4_1_1_9_1 = 133 ,
uid_1_2_840_10008_5_1_4_1_1_9_1_1 = 134 ,
uid_1_2_840_10008_5_1_4_1_1_9_1_2 = 135 ,
uid_1_2_840_10008_5_1_4_1_1_9_1_3 = 136 ,
uid_1_2_840_10008_5_1_4_1_1_9_2_1 = 137 ,
uid_1_2_840_10008_5_1_4_1_1_9_3_1 = 138 ,
uid_1_2_840_10008_5_1_4_1_1_9_4_1 = 139 ,
uid_1_2_840_10008_5_1_4_1_1_10 = 140 ,
uid_1_2_840_10008_5_1_4_1_1_11 = 141 ,
uid_1_2_840_10008_5_1_4_1_1_11_1 = 142 ,
uid_1_2_840_10008_5_1_4_1_1_11_2 = 143 ,
uid_1_2_840_10008_5_1_4_1_1_11_3 = 144 ,
uid_1_2_840_10008_5_1_4_1_1_11_4 = 145 ,
uid_1_2_840_10008_5_1_4_1_1_12_1 = 146 ,
uid_1_2_840_10008_5_1_4_1_1_12_1_1 = 147 ,
uid_1_2_840_10008_5_1_4_1_1_12_2 = 148 ,
uid_1_2_840_10008_5_1_4_1_1_12_2_1 = 149 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_1 = 150 ,

uid_1_2_840_10008_5_1_4_1_1_13_1_2 = 151 ,
uid_1_2_840_10008_5_1_4_1_1_12_3 = 152 ,
uid_1_2_840_10008_5_1_4_1_1_20 = 153 ,
uid_1_2_840_10008_5_1_4_1_1_66 = 154 ,
uid_1_2_840_10008_5_1_4_1_1_66_1 = 155 ,
uid_1_2_840_10008_5_1_4_1_1_66_2 = 156 ,
uid_1_2_840_10008_5_1_4_1_1_66_3 = 157 ,
uid_1_2_840_10008_5_1_4_1_1_66_4 = 158 ,
uid_1_2_840_10008_5_1_4_1_1_67 = 159 ,
uid_1_2_840_10008_5_1_4_1_1_77_1 = 160 ,
uid_1_2_840_10008_5_1_4_1_1_77_2 = 161 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_1 = 162 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1 = 163 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_2 = 164 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1 = 165 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_3 = 166 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_4 = 167 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1 = 168 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1 = 169 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2 = 170 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3 = 171 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4 = 172 ,
uid_1_2_840_10008_5_1_4_1_1_88_1 = 173 ,
uid_1_2_840_10008_5_1_4_1_1_88_2 = 174 ,
uid_1_2_840_10008_5_1_4_1_1_88_3 = 175 ,
uid_1_2_840_10008_5_1_4_1_1_88_4 = 176 ,
uid_1_2_840_10008_5_1_4_1_1_88_11 = 177 ,
uid_1_2_840_10008_5_1_4_1_1_88_22 = 178 ,
uid_1_2_840_10008_5_1_4_1_1_88_33 = 179 ,
uid_1_2_840_10008_5_1_4_1_1_88_40 = 180 ,
uid_1_2_840_10008_5_1_4_1_1_88_50 = 181 ,
uid_1_2_840_10008_5_1_4_1_1_88_59 = 182 ,
uid_1_2_840_10008_5_1_4_1_1_88_65 = 183 ,
uid_1_2_840_10008_5_1_4_1_1_88_67 = 184 ,
uid_1_2_840_10008_5_1_4_1_1_104_1 = 185 ,
uid_1_2_840_10008_5_1_4_1_1_104_2 = 186 ,
uid_1_2_840_10008_5_1_4_1_1_128 = 187 ,
uid_1_2_840_10008_5_1_4_1_1_129 = 188 ,
uid_1_2_840_10008_5_1_4_1_1_481_1 = 189 ,
uid_1_2_840_10008_5_1_4_1_1_481_2 = 190 ,
uid_1_2_840_10008_5_1_4_1_1_481_3 = 191 ,
uid_1_2_840_10008_5_1_4_1_1_481_4 = 192 ,
uid_1_2_840_10008_5_1_4_1_1_481_5 = 193 ,
uid_1_2_840_10008_5_1_4_1_1_481_6 = 194 ,
uid_1_2_840_10008_5_1_4_1_1_481_7 = 195 ,
uid_1_2_840_10008_5_1_4_1_1_481_8 = 196 ,
uid_1_2_840_10008_5_1_4_1_1_481_9 = 197 ,
uid_1_2_840_10008_5_1_4_1_2_1_1 = 198 ,
uid_1_2_840_10008_5_1_4_1_2_1_2 = 199 ,
uid_1_2_840_10008_5_1_4_1_2_1_3 = 200 ,
uid_1_2_840_10008_5_1_4_1_2_2_1 = 201 ,
uid_1_2_840_10008_5_1_4_1_2_2_2 = 202 ,
uid_1_2_840_10008_5_1_4_1_2_2_3 = 203 ,
uid_1_2_840_10008_5_1_4_1_2_3_1 = 204 ,

uid_1_2_840_10008_5_1_4_1_2_3_2 = 205 ,
uid_1_2_840_10008_5_1_4_1_2_3_3 = 206 ,
uid_1_2_840_10008_5_1_4_31 = 207 ,
uid_1_2_840_10008_5_1_4_32_1 = 208 ,
uid_1_2_840_10008_5_1_4_32_2 = 209 ,
uid_1_2_840_10008_5_1_4_32_3 = 210 ,
uid_1_2_840_10008_5_1_4_32 = 211 ,
uid_1_2_840_10008_5_1_4_33 = 212 ,
uid_1_2_840_10008_5_1_4_34_1 = 213 ,
uid_1_2_840_10008_5_1_4_34_2 = 214 ,
uid_1_2_840_10008_5_1_4_34_3 = 215 ,
uid_1_2_840_10008_5_1_4_34_4 = 216 ,
uid_1_2_840_10008_5_1_4_34_4_1 = 217 ,
uid_1_2_840_10008_5_1_4_34_4_2 = 218 ,
uid_1_2_840_10008_5_1_4_34_4_3 = 219 ,
uid_1_2_840_10008_5_1_4_34_4_4 = 220 ,
uid_1_2_840_10008_5_1_4_34_5 = 221 ,
uid_1_2_840_10008_5_1_4_37_1 = 222 ,
uid_1_2_840_10008_5_1_4_37_2 = 223 ,
uid_1_2_840_10008_5_1_4_37_3 = 224 ,
uid_1_2_840_10008_5_1_4_38_1 = 225 ,
uid_1_2_840_10008_5_1_4_38_2 = 226 ,
uid_1_2_840_10008_5_1_4_38_3 = 227 ,
uid_1_2_840_10008_5_1_4_41 = 228 ,
uid_1_2_840_10008_5_1_4_42 = 229 ,
uid_1_2_840_10008_15_0_3_1 = 230 ,
uid_1_2_840_10008_15_0_3_2 = 231 ,
uid_1_2_840_10008_15_0_3_3 = 232 ,
uid_1_2_840_10008_15_0_3_4 = 233 ,
uid_1_2_840_10008_15_0_3_5 = 234 ,
uid_1_2_840_10008_15_0_3_6 = 235 ,
uid_1_2_840_10008_15_0_3_7 = 236 ,
uid_1_2_840_10008_15_0_3_8 = 237 ,
uid_1_2_840_10008_15_0_3_9 = 238 ,
uid_1_2_840_10008_15_0_3_10 = 239 ,
uid_1_2_840_10008_15_0_3_11 = 240 ,
uid_1_2_840_10008_15_0_3_12 = 241 ,
uid_1_2_840_10008_15_0_3_13 = 242 ,
uid_1_2_840_10008_15_0_3_14 = 243 ,
uid_1_2_840_10008_15_0_3_15 = 244 ,
uid_1_2_840_10008_15_0_3_16 = 245 ,
uid_1_2_840_10008_15_0_3_17 = 246 ,
uid_1_2_840_10008_15_0_3_18 = 247 ,
uid_1_2_840_10008_15_0_3_19 = 248 ,
uid_1_2_840_10008_15_0_3_20 = 249 ,
uid_1_2_840_10008_15_0_3_21 = 250 ,
uid_1_2_840_10008_15_0_3_22 = 251 ,
uid_1_2_840_10008_15_0_3_23 = 252 ,
uid_1_2_840_10008_15_0_3_24 = 253 ,
uid_1_2_840_10008_15_0_3_25 = 254 ,
uid_1_2_840_10008_15_0_3_26 = 255 ,
uid_1_2_840_10008_15_0_3_27 = 256 ,
uid_1_2_840_10008_15_0_3_28 = 257 ,
uid_1_2_840_10008_15_0_3_29 = 258 ,

uid_1_2_840_10008_15_0_3_30 = 259 ,
uid_1_2_840_10008_15_0_3_31 = 260 ,
uid_1_2_840_10008_15_0_4_1 = 261 ,
uid_1_2_840_10008_15_0_4_2 = 262 ,
uid_1_2_840_10008_15_0_4_3 = 263 ,
uid_1_2_840_10008_15_0_4_4 = 264 ,
uid_1_2_840_10008_15_0_4_5 = 265 ,
uid_1_2_840_10008_15_0_4_6 = 266 ,
uid_1_2_840_10008_15_0_4_7 = 267 ,
uid_1_2_840_10008_15_0_4_8 = 268 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_6 = 269 ,
uid_1_2_840_10008_5_1_4_1_1_6_2 = 270 ,
uid_1_2_840_10008_5_1_4_1_1_66_5 = 271 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_3 = 272 ,
uid_1_2_840_10008_5_1_4_1_1_2_2 = 273 ,
uid_1_2_840_10008_5_1_4_1_1_4_4 = 274 ,
uid_1_2_840_10008_5_1_4_1_1_128_1 = 275 ,
uid_1_2_840_10008_1_2_4_101 = 276 ,
uid_1_2_840_10008_1_2_4_102 = 277 ,
uid_1_2_840_10008_1_2_4_103 = 278 ,
uid_1_2_840_10008_1_5_2 = 279 ,
uid_1_2_840_10008_1_5_3 = 280 ,
uid_1_2_840_10008_1_5_4 = 281 ,
uid_1_2_840_10008_1_5_5 = 282 ,
uid_1_2_840_10008_1_5_6 = 283 ,
uid_1_2_840_10008_1_5_7 = 284 ,
uid_1_2_840_10008_1_5_8 = 285 ,
uid_1_2_840_10008_1_20 = 286 ,
uid_1_2_840_10008_2_16_5 = 287 ,
uid_1_2_840_10008_2_16_6 = 288 ,
uid_1_2_840_10008_2_16_7 = 289 ,
uid_1_2_840_10008_2_16_8 = 290 ,
uid_1_2_840_10008_2_16_9 = 291 ,
uid_1_2_840_10008_2_16_10 = 292 ,
uid_1_2_840_10008_2_16_11 = 293 ,
uid_1_2_840_10008_2_16_12 = 294 ,
uid_1_2_840_10008_2_16_13 = 295 ,
uid_1_2_840_10008_2_16_14 = 296 ,
uid_1_2_840_10008_5_1_1_40 = 297 ,
uid_1_2_840_10008_5_1_1_40_1 = 298 ,
uid_1_2_840_10008_5_1_4_1_1_9_4_2 = 299 ,
uid_1_2_840_10008_5_1_4_1_1_9_5_1 = 300 ,
uid_1_2_840_10008_5_1_4_1_1_9_6_1 = 301 ,
uid_1_2_840_10008_5_1_4_1_1_11_5 = 302 ,
uid_1_2_840_10008_5_1_4_1_1_11_6 = 303 ,
uid_1_2_840_10008_1_2_4_104 = 304 ,
uid_1_2_840_10008_1_2_4_105 = 305 ,
uid_1_2_840_10008_1_2_4_106 = 306 ,
uid_1_2_840_10008_1_2_4_107 = 307 ,
uid_1_2_840_10008_1_2_4_108 = 308 ,
uid_1_2_840_10008_1_5_1 = 309 ,
uid_1_2_840_10008_5_1_4_1_1_11_7 = 310 ,
uid_1_2_840_10008_5_1_4_1_1_11_8 = 311 ,
uid_1_2_840_10008_5_1_4_1_1_11_9 = 312 ,

uid_1_2_840_10008_5_1_4_1_1_11_10 = 313 ,
uid_1_2_840_10008_5_1_4_1_1_11_11 = 314 ,
uid_1_2_840_10008_5_1_4_1_1_12_77 = 315 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_4 = 316 ,
uid_1_2_840_10008_5_1_4_1_1_13_1_5 = 317 ,
uid_1_2_840_10008_5_1_4_1_1_14_1 = 318 ,
uid_1_2_840_10008_5_1_4_1_1_14_2 = 319 ,
uid_1_2_840_10008_5_1_4_1_1_30 = 320 ,
uid_1_2_840_10008_5_1_4_1_1_40 = 321 ,
uid_1_2_840_10008_5_1_4_1_1_66_6 = 322 ,
uid_1_2_840_10008_5_1_4_1_1_68_1 = 323 ,
uid_1_2_840_10008_5_1_4_1_1_68_2 = 324 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5 = 325 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6 = 326 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7 = 327 ,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8 = 328 ,
uid_1_2_840_10008_5_1_4_1_1_78_1 = 329 ,
uid_1_2_840_10008_5_1_4_1_1_78_2 = 330 ,
uid_1_2_840_10008_5_1_4_1_1_78_3 = 331 ,
uid_1_2_840_10008_5_1_4_1_1_78_4 = 332 ,
uid_1_2_840_10008_5_1_4_1_1_78_5 = 333 ,
uid_1_2_840_10008_5_1_4_1_1_78_6 = 334 ,
uid_1_2_840_10008_5_1_4_1_1_78_7 = 335 ,
uid_1_2_840_10008_5_1_4_1_1_78_8 = 336 ,
uid_1_2_840_10008_5_1_4_1_1_79_1 = 337 ,
uid_1_2_840_10008_5_1_4_1_1_80_1 = 338 ,
uid_1_2_840_10008_5_1_4_1_1_81_1 = 339 ,
uid_1_2_840_10008_5_1_4_1_1_82_1 = 340 ,
uid_1_2_840_10008_5_1_4_1_1_88_34 = 341 ,
uid_1_2_840_10008_5_1_4_1_1_88_35 = 342 ,
uid_1_2_840_10008_5_1_4_1_1_88_68 = 343 ,
uid_1_2_840_10008_5_1_4_1_1_88_69 = 344 ,
uid_1_2_840_10008_5_1_4_1_1_88_70 = 345 ,
uid_1_2_840_10008_5_1_4_1_1_88_71 = 346 ,
uid_1_2_840_10008_5_1_4_1_1_88_72 = 347 ,
uid_1_2_840_10008_5_1_4_1_1_88_73 = 348 ,
uid_1_2_840_10008_5_1_4_1_1_88_74 = 349 ,
uid_1_2_840_10008_5_1_4_1_1_88_75 = 350 ,
uid_1_2_840_10008_5_1_4_1_1_90_1 = 351 ,
uid_1_2_840_10008_5_1_4_1_1_104_3 = 352 ,
uid_1_2_840_10008_5_1_4_1_1_130 = 353 ,
uid_1_2_840_10008_5_1_4_1_1_131 = 354 ,
uid_1_2_840_10008_5_1_4_1_1_200_1 = 355 ,
uid_1_2_840_10008_5_1_4_1_1_200_2 = 356 ,
uid_1_2_840_10008_5_1_4_1_1_200_3 = 357 ,
uid_1_2_840_10008_5_1_4_1_1_200_4 = 358 ,
uid_1_2_840_10008_5_1_4_1_1_200_5 = 359 ,
uid_1_2_840_10008_5_1_4_1_1_200_6 = 360 ,
uid_1_2_840_10008_5_1_4_1_1_481_10 = 361 ,
uid_1_2_840_10008_5_1_4_1_1_481_11 = 362 ,
uid_1_2_840_10008_5_1_4_1_1_501_1 = 363 ,
uid_1_2_840_10008_5_1_4_1_1_501_2_1 = 364 ,
uid_1_2_840_10008_5_1_4_1_1_501_2_2 = 365 ,
uid_1_2_840_10008_5_1_4_1_1_501_3 = 366 ,

uid_1_2_840_10008_5_1_4_1_1_501_4 = 367 ,
uid_1_2_840_10008_5_1_4_1_1_501_5 = 368 ,
uid_1_2_840_10008_5_1_4_1_1_501_6 = 369 ,
uid_1_2_840_10008_5_1_4_1_1_601_1 = 370 ,
uid_1_2_840_10008_5_1_4_1_1_601_2 = 371 ,
uid_1_2_840_10008_5_1_4_1_2_4_2 = 372 ,
uid_1_2_840_10008_5_1_4_1_2_4_3 = 373 ,
uid_1_2_840_10008_5_1_4_1_2_5_3 = 374 ,
uid_1_2_840_10008_5_1_4_20_1 = 375 ,
uid_1_2_840_10008_5_1_4_20_2 = 376 ,
uid_1_2_840_10008_5_1_4_20_3 = 377 ,
uid_1_2_840_10008_5_1_4_34_5_1 = 378 ,
uid_1_2_840_10008_5_1_4_34_6 = 379 ,
uid_1_2_840_10008_5_1_4_34_6_1 = 380 ,
uid_1_2_840_10008_5_1_4_34_6_2 = 381 ,
uid_1_2_840_10008_5_1_4_34_6_3 = 382 ,
uid_1_2_840_10008_5_1_4_34_6_4 = 383 ,
uid_1_2_840_10008_5_1_4_34_7 = 384 ,
uid_1_2_840_10008_5_1_4_34_8 = 385 ,
uid_1_2_840_10008_5_1_4_34_9 = 386 ,
uid_1_2_840_10008_5_1_4_34_10 = 387 ,
uid_1_2_840_10008_5_1_4_38_4 = 388 ,
uid_1_2_840_10008_5_1_4_39_1 = 389 ,
uid_1_2_840_10008_5_1_4_39_2 = 390 ,
uid_1_2_840_10008_5_1_4_39_3 = 391 ,
uid_1_2_840_10008_5_1_4_39_4 = 392 ,
uid_1_2_840_10008_5_1_4_43_1 = 393 ,
uid_1_2_840_10008_5_1_4_43_2 = 394 ,
uid_1_2_840_10008_5_1_4_43_3 = 395 ,
uid_1_2_840_10008_5_1_4_43_4 = 396 ,
uid_1_2_840_10008_5_1_4_44_1 = 397 ,
uid_1_2_840_10008_5_1_4_44_2 = 398 ,
uid_1_2_840_10008_5_1_4_44_3 = 399 ,
uid_1_2_840_10008_5_1_4_44_4 = 400 ,
uid_1_2_840_10008_5_1_4_45_1 = 401 ,
uid_1_2_840_10008_5_1_4_45_2 = 402 ,
uid_1_2_840_10008_5_1_4_45_3 = 403 ,
uid_1_2_840_10008_5_1_4_45_4 = 404 ,
uid_1_2_840_10008_7_1_1 = 405 ,
uid_1_2_840_10008_7_1_2 = 406 ,
uid_1_2_840_10008_8_1_1 = 407 ,
uid_1_2_840_10008_5_1_4_1_1_4_3 = 408 ,
uid_1_2_840_10008_15_1_1 = 409 }

Public Member Functions

- const char ∗ GetName () const
- const char ∗ GetString () const
- operator TSType () const
- bool SetFromUID (const char ∗str)

Static Public Member Functions

- static unsigned int GetNumberOfTransferSyntaxStrings ()
- static const char *const * GetTransferSyntaxString (unsigned int ts)
- static TransferSyntaxStringsType GetTransferSyntaxStrings ()
- static const char * GetUIDName (unsigned int ts)
- static const char * GetUIDString (unsigned int ts)

12.324.1 Detailed Description

all known uids

Examples

GenerateStandardSOPClasses.cxx.

12.324.2 Member Typedef Documentation

12.324.2.1 TransferSyntaxStringsType

```
typedef const char* const(* gdcm::UIDs::TransferSyntaxStringsType)[2]
```

12.324.3 Member Enumeration Documentation

12.324.3.1 TSName

```
enum gdcm::UIDs::TSName
```

12.324.3.2 TSType

```
enum gdcm::UIDs::TSType
```

12.324.4 Member Function Documentation

12.324.4.1 GetName()

```
const char * gdcm::UIDs::GetName () const
```

When object is Initialize function return the well known name associated with uid return NULL when not initialized

Examples

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

12.324.4.2 GetNumberOfTransferSyntaxStrings()

```
unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ()  [static]
```

12.324.4.3 GetString()

```
const char * gdcm::UIDs::GetString () const
```

When object is Initialize function return the uid return NULL when not initialized

Examples

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

12.324.4.4 GetTransferSyntaxString()

```
const char *const * gdcm::UIDs::GetTransferSyntaxString (
            unsigned int ts)  [static]
```

12.324.4.5 GetTransferSyntaxStrings()

```
TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ()  [static]
```

12.324.4.6 GetUIDName()

```
const char * gdcm::UIDs::GetUIDName (
            unsigned int ts)  [static]
```

12.324.4.7 GetUIDString()

```
const char * gdcm::UIDs::GetUIDString (
            unsigned int ts)  [static]
```

12.324.4.8 operator TSType()

```
gdcm::UIDs::operator TSType () const  [inline]
```

12.324.4.9 SetFromUID()

```
bool gdcm::UIDs::SetFromUID (
            const char * str)
```

Initialize object from a string (a uid number) return false on error, and internal state is set to 0

Examples

GenerateStandardSOPClasses.cxx.

The documentation for this class was generated from the following file:

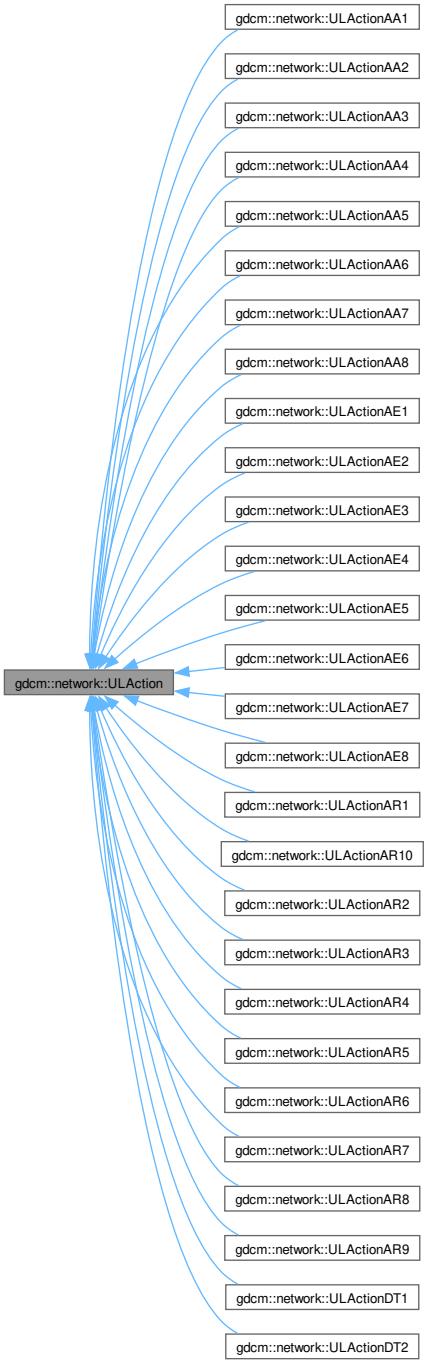
- gdcmUIDs.h

12.325 gdcm::network::ULAction Class Reference

ULAction.

```
#include <gdcmULAction.h>
```

Inheritance diagram for gdcm::network::ULAction:



Public Member Functions

- ULAction ()=default

- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete
- virtual EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaiting↩ ForEvent, EEventID &outRaisedEvent)=0

12.325.1 Detailed Description

ULAction.

A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

Essentially, the ULConnectionManager will take this object, determined from the current ULState of the ULConnection, and pass the ULConnection object to the ULAction. The ULAction will then invoke whatever necessary commands are required by a given action.

The result of a ULAction is a ULEvent (ie, what happened as a result of the action).

This ULEvent is passed to the ULState, so that the transition to the next state can occur.

Actions are associated with Payloads – be those filestreams, AETitles to establish connections, whatever. The actual parameters that the user will pass via an action will come through a Payload object, which should, in itself, be some gdcm-based object (but not all objects can be payloads; sending a single dataelement as a payload isn't meaningful). As such, each action has its own particular payload.

For the sake of keeping files together, both the particular payload class and the action class will be defined in the same header file. Payloads should JUST be data (or streams), NO METHODS.

Some actions perform changes that should raise events on the local system, and some actions perform changes that will require waiting for events from the remote system.

Therefore, this base action has been modified so that those events are set by each action. When the event loop runs an action, it will then test to see if a local event was raised by the action, and if so, perform the appropriate subsequent action. If the action requires waiting for a response from the remote system, then the event loop will sit there (presumably with the ARTIM timer running) and wait for a response from the remote system. Once a response is obtained, then the the rest of the state transitions can happen.

12.325.2 Constructor & Destructor Documentation

12.325.2.1 ULAction() [1/2]

```
gdcm::network::ULAction::ULAction ()  [default]
```

Referenced by ULAction(), and operator=().

12.325.2.2 ~ULAction()

```
virtual gdcm::network::ULAction::~ULAction ()  [virtual], [default]
```

12.325.2.3 ULAction() [2/2]

```
gdcm::network::ULAction::ULAction (
            const ULAction & inAction)  [delete]
```

References ULAction().

12.325.3 Member Function Documentation

12.325.3.1 operator=()

```
void gdcm::network::ULAction::operator= (
            const ULAction & )  [delete]
```

References ULAction().

12.325.3.2 PerformAction()

```
virtual EStateID gdcm::network::ULAction::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [pure virtual]
```

Implemented in gdcm::network::ULActionAA1, gdcm::network::ULActionAA2, gdcm::network::ULActionAA3, gdcm::network::ULActionAA4 gdcm::network::ULActionAA5, gdcm::network::ULActionAA6, gdcm::network::ULActionAA7, gdcm::network::ULActionAA8, gdcm::network::ULActionAE1, gdcm::network::ULActionAE2, gdcm::network::ULActionAE3, gdcm::network::ULActionAE4, gdcm::network::ULActionAE5, gdcm::network::ULActionAE6, gdcm::network::ULActionAE7, gdcm::network::ULActionAE8, gdcm::network::ULActionAR10, gdcm::network::ULActionAR1, gdcm::network::ULActionAR2, gdcm::network::ULActionAR3, gdcm::network::ULActionAR4, gdcm::network::ULActionAR5, gdcm::network::ULActionAR6, gdcm::network::ULActionAR7, gdcm::network::ULActionAR8, gdcm::network::ULActionAR9, gdcm::network::ULActionDT1, and gdcm::network::ULActionDT2.

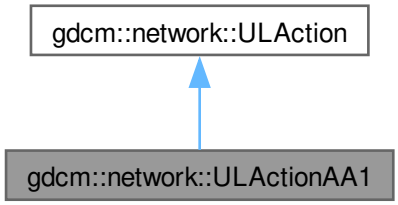
The documentation for this class was generated from the following file:

- gdcmULAction.h

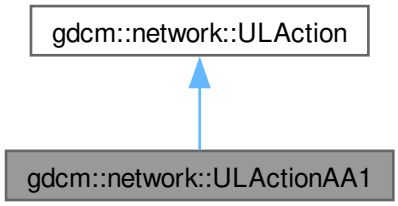
12.326 gdcm::network::ULActionAA1 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA1:



Collaboration diagram for gdcm::network::ULActionAA1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.326.1 Member Function Documentation

12.326.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

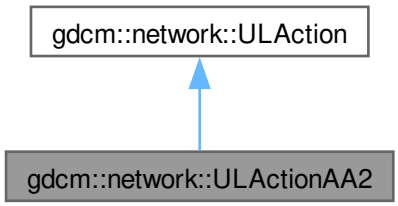
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

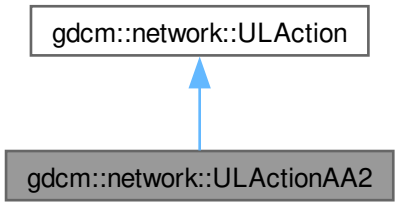
12.327 gdcm::network::ULActionAA2 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA2:



Collaboration diagram for gdcm::network::ULActionAA2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.327.1 Member Function Documentation

12.327.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

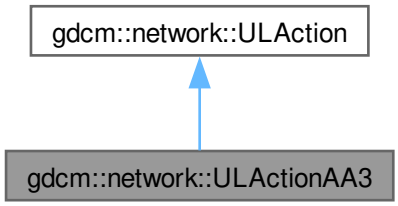
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

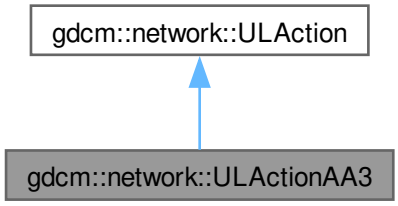
12.328 gdcm::network::ULActionAA3 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA3:



Collaboration diagram for gdcm::network::ULActionAA3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.328.1 Member Function Documentation

12.328.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA3::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

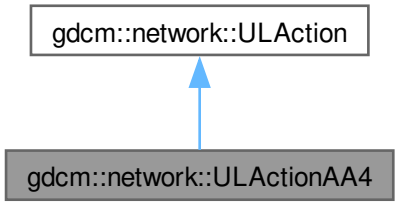
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

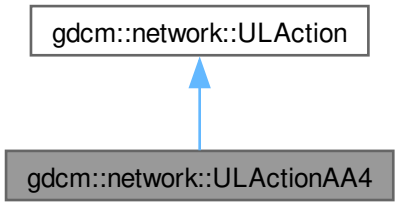
12.329 gdcm::network::ULActionAA4 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA4:



Collaboration diagram for gdcm::network::ULActionAA4:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.329.1 Member Function Documentation

12.329.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

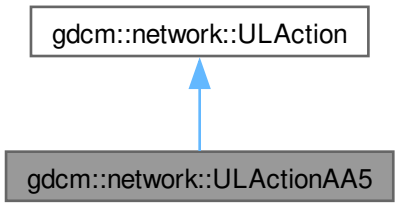
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

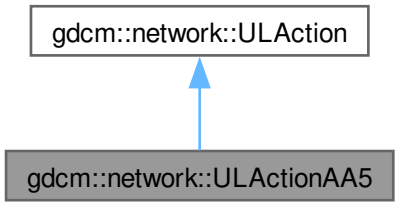
12.330 gdcm::network::ULActionAA5 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA5:



Collaboration diagram for gdcm::network::ULActionAA5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.330.1 Member Function Documentation

12.330.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

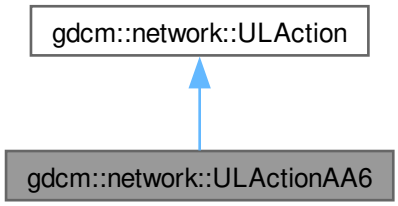
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

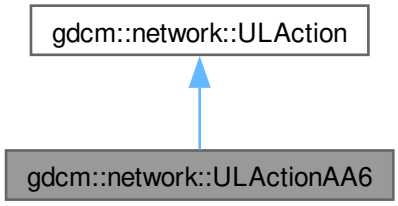
12.331 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA6:



Collaboration diagram for gdcm::network::ULActionAA6:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.331.1 Member Function Documentation

12.331.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA6::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

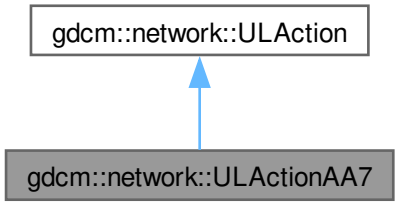
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

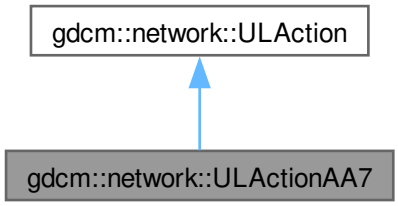
12.332 gdcm::network::ULActionAA7 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA7:



Collaboration diagram for gdcm::network::ULActionAA7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.332.1 Member Function Documentation

12.332.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA7::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

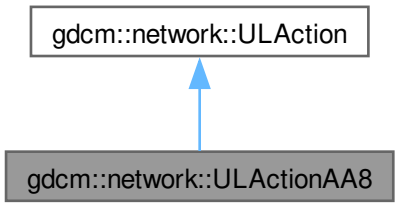
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

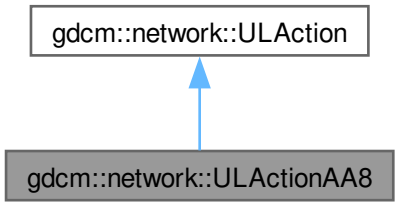
12.333 gdcm::network::ULActionAA8 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA8:



Collaboration diagram for gdcm::network::ULActionAA8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.333.1 Member Function Documentation

12.333.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAA8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

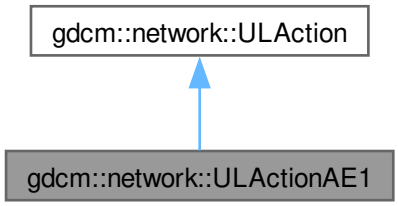
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

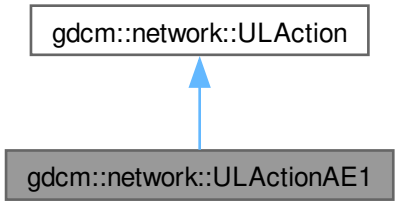
12.334 gdcm::network::ULActionAE1 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE1:



Collaboration diagram for gdcm::network::ULActionAE1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.334.1 Member Function Documentation

12.334.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

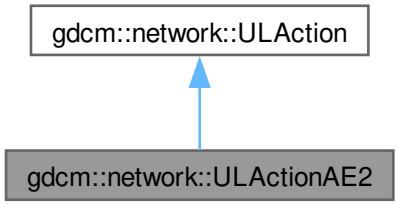
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

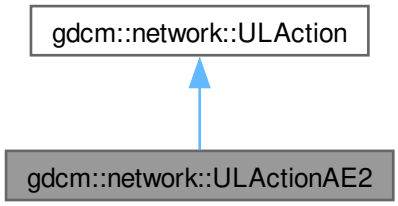
12.335 gdcm::network::ULActionAE2 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE2:



Collaboration diagram for gdcm::network::ULActionAE2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.335.1 Member Function Documentation

12.335.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

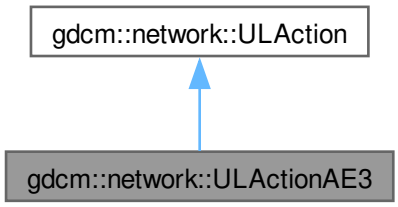
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

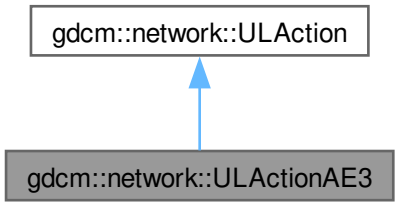
12.336 gdcm::network::ULActionAE3 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE3:



Collaboration diagram for gdcm::network::ULActionAE3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.336.1 Member Function Documentation

12.336.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE3::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

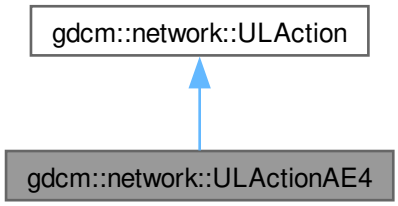
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

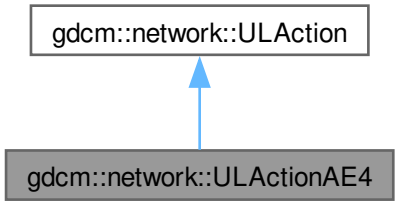
12.337 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE4:



Collaboration diagram for gdcm::network::ULActionAE4:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.337.1 Member Function Documentation

12.337.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

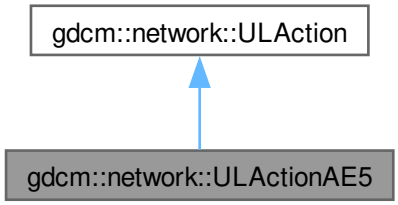
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

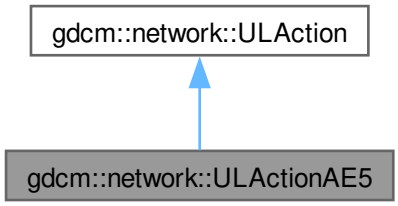
12.338 gdcm::network::ULActionAE5 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE5:



Collaboration diagram for gdcm::network::ULActionAE5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.338.1 Member Function Documentation

12.338.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

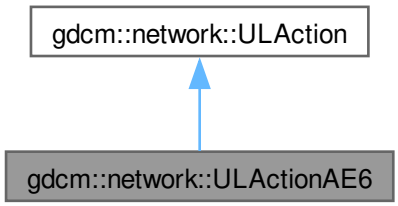
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

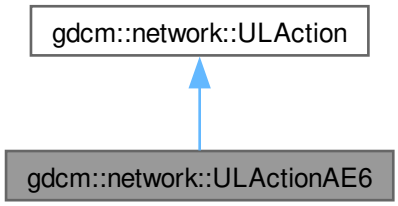
12.339 gdcm::network::ULActionAE6 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE6:



Collaboration diagram for gdcm::network::ULActionAE6:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.339.1 Member Function Documentation

12.339.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE6::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

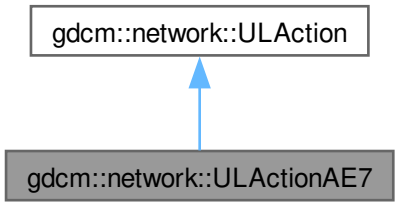
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

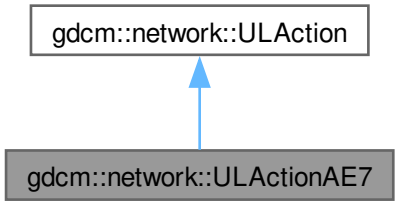
12.340 gdcm::network::ULActionAE7 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE7:



Collaboration diagram for gdcm::network::ULActionAE7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.340.1 Member Function Documentation

12.340.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE7::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

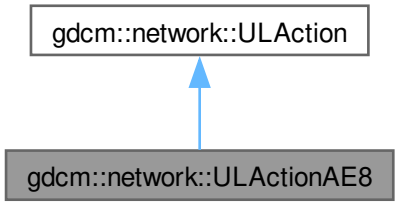
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

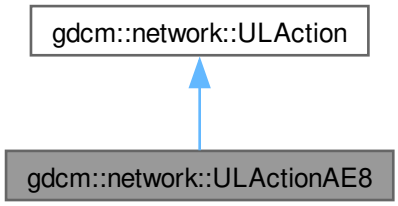
12.341 gdcm::network::ULActionAE8 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE8:



Collaboration diagram for gdcm::network::ULActionAE8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.341.1 Member Function Documentation

12.341.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAE8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

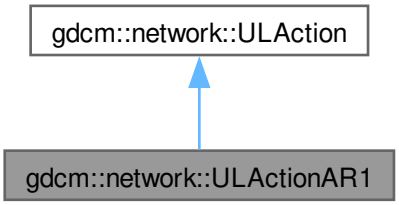
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

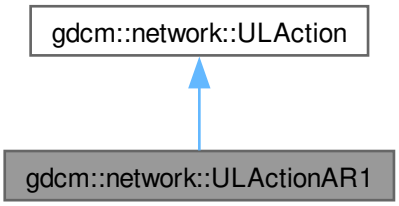
12.342 gdcm::network::ULActionAR1 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR1:



Collaboration diagram for gdcm::network::ULActionAR1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.342.1 Member Function Documentation

12.342.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

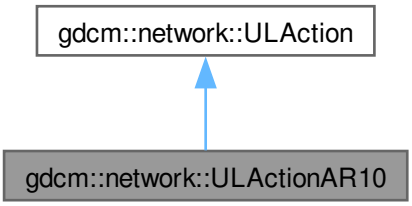
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

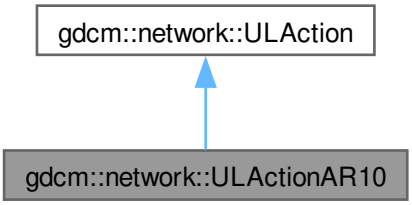
12.343 gdcm::network::ULActionAR10 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR10:



Collaboration diagram for gdcm::network::ULActionAR10:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.343.1 Member Function Documentation

12.343.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR10::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

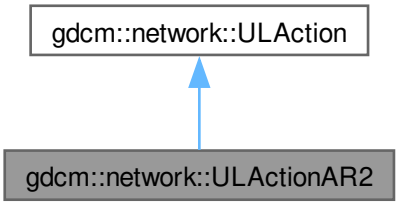
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

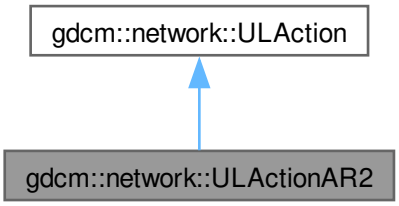
12.344 gdcm::network::ULActionAR2 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR2:



Collaboration diagram for gdcm::network::ULActionAR2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.344.1 Member Function Documentation

12.344.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

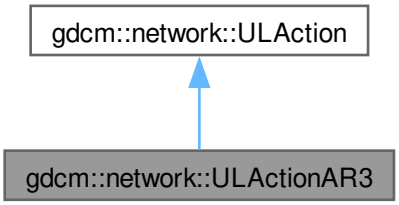
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

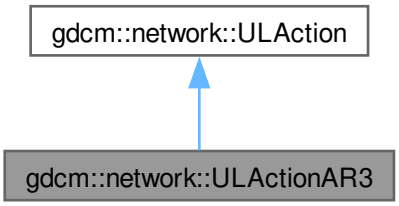
12.345 gdcm::network::ULActionAR3 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR3:



Collaboration diagram for gdcm::network::ULActionAR3:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.345.1 Member Function Documentation

12.345.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR3::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

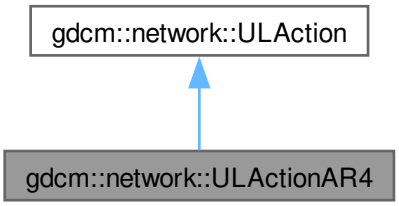
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

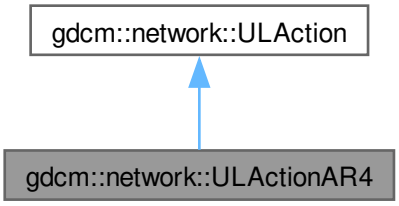
12.346 gdcm::network::ULActionAR4 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR4:



Collaboration diagram for gdcm::network::ULActionAR4:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.346.1 Member Function Documentation

12.346.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR4::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

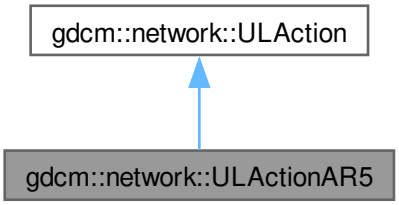
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

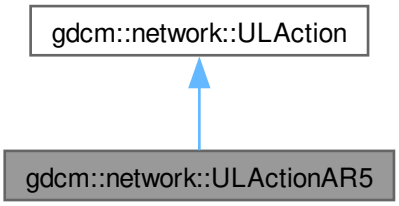
12.347 gdcm::network::ULActionAR5 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR5:



Collaboration diagram for gdcm::network::ULActionAR5:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.347.1 Member Function Documentation

12.347.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR5::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

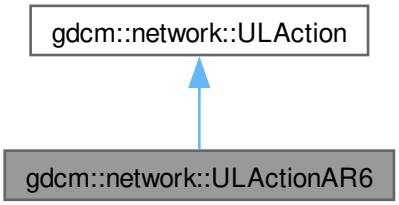
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

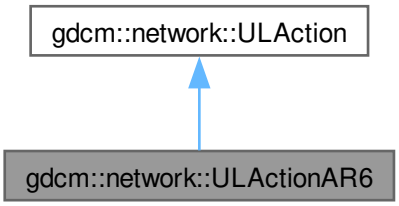
12.348 gdcm::network::ULActionAR6 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR6:



Collaboration diagram for gdcm::network::ULActionAR6:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.348.1 Member Function Documentation

12.348.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR6::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

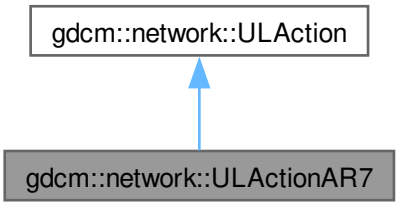
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

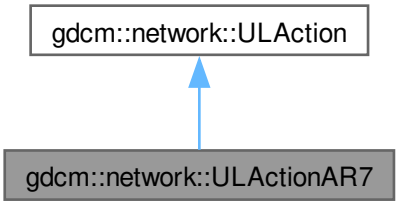
12.349 gdcm::network::ULActionAR7 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR7:



Collaboration diagram for gdcm::network::ULActionAR7:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.349.1 Member Function Documentation

12.349.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR7::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

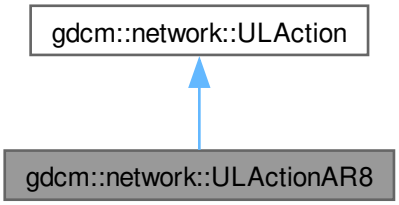
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

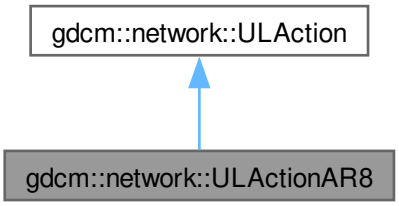
12.350 gdcm::network::ULActionAR8 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR8:



Collaboration diagram for gdcm::network::ULActionAR8:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.350.1 Member Function Documentation

12.350.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR8::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

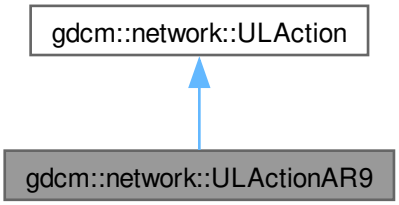
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

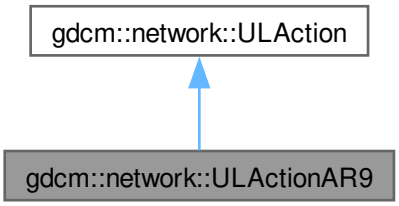
12.351 gdcm::network::ULActionAR9 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR9:



Collaboration diagram for gdcm::network::ULActionAR9:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.351.1 Member Function Documentation

12.351.1.1 PerformAction()

```
EStateID gdcm::network::ULActionAR9::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

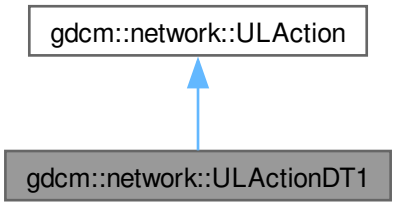
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

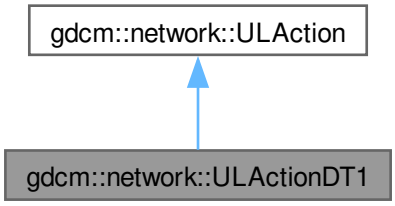
12.352 gdcm::network::ULActionDT1 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for gdcm::network::ULActionDT1:



Collaboration diagram for gdcm::network::ULActionDT1:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩ Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.352.1 Member Function Documentation

12.352.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT1::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

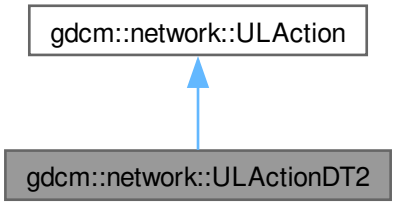
The documentation for this class was generated from the following file:

- gdcmULActionDT.h

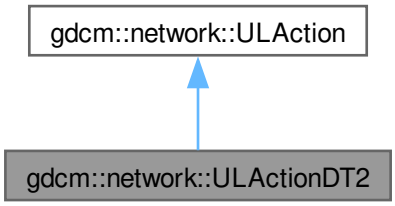
12.353 gdcm::network::ULActionDT2 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for gdcm::network::ULActionDT2:



Collaboration diagram for gdcm::network::ULActionDT2:



Public Member Functions

- EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingFor↩Event, EEventID &outRaisedEvent) override

Public Member Functions inherited from gdcm::network::ULAction

- ULAction ()=default
- ULAction (const ULAction &inAction)=delete
- virtual ~ULAction ()=default
- void operator= (const ULAction &)=delete

12.353.1 Member Function Documentation

12.353.1.1 PerformAction()

```
EStateID gdcm::network::ULActionDT2::PerformAction (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent)  [override], [virtual]
```

Implements gdcm::network::ULAction.

The documentation for this class was generated from the following file:

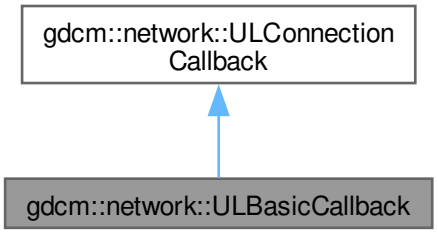
- gdcmULActionDT.h

12.354 gdcm::network::ULBasicCallback Class Reference

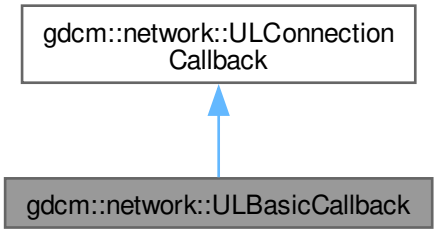
ULBasicCallback.

```
#include <gdcmULBasicCallback.h>
```

Inheritance diagram for gdcm::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



Public Member Functions

- ULBasicCallback ()=default
- ~ULBasicCallback () override=default
- std::vector< DataSet > const & GetDataSets () const
- std::vector< DataSet > const & GetResponses () const
- void HandleDataSet (const DataSet &inDataSet) override
- void HandleResponse (const DataSet &inDataSet) override

Public Member Functions inherited from gdcm::network::ULConnectionCallback

- ULConnectionCallback ()
- virtual ~ULConnectionCallback ()=default
- bool DataSetHandles () const
- void ResetHandledDataSet ()
- void SetImplicitFlag (const bool imp)

Additional Inherited Members

Protected Member Functions inherited from gdcm::network::ULConnectionCallback

- void DataSetHandled ()

Protected Attributes inherited from gdcm::network::ULConnectionCallback

- bool mImplicit

12.354.1 Detailed Description

ULBasicCallback.

This is the most basic of callbacks for how the ULConnectionManager handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to disk, etc. This class is NOT THREAD SAFE. Access the dataset vector after the entire set of datasets has been returned by the ULConnectionManager.

12.354.2 Constructor & Destructor Documentation

12.354.2.1 ULBasicCallback()

```
gdcm::network::ULBasicCallback::ULBasicCallback () [default]
```

12.354.2.2 ~ULBasicCallback()

```
gdcm::network::ULBasicCallback::~ULBasicCallback () [override], [default]
```

12.354.3 Member Function Documentation

12.354.3.1 GetDataSets()

```
std::vector< DataSet > const & gdcm::network::ULBasicCallback::GetDataSets () const
```

12.354.3.2 GetResponses()

```
std::vector< DataSet > const & gdcm::network::ULBasicCallback::GetResponses () const
```

12.354.3.3 HandleDataSet()

```
void gdcm::network::ULBasicCallback::HandleDataSet (
            const DataSet & inDataSet) [override], [virtual]
```

Implements gdcm::network::ULConnectionCallback.

12.354.3.4 HandleResponse()

```
void gdcm::network::ULBasicCallback::HandleResponse (
            const DataSet & inDataSet) [override], [virtual]
```

Implements gdcm::network::ULConnectionCallback.

The documentation for this class was generated from the following file:

- gdcmULBasicCallback.h

12.355 gdcm::network::ULConnection Class Reference

ULConnection.

```
#include <gdcmULConnection.h>
```

Public Member Functions

- ULConnection (const ULConnection &)=delete
- ULConnection (const ULConnectionInfo &inUserInformation)
- virtual ~ULConnection ()
- void AddAcceptedPresentationContext (const PresentationContextAC &inPC)
- PresentationContextRQ FindContext (const DataElement &de) const
- std::vector< PresentationContextAC > & GetAcceptedPresentationContexts ()
- std::vector< PresentationContextAC > const & GetAcceptedPresentationContexts () const
- const ULConnectionInfo & GetConnectionInfo () const
- uint32_t GetMaxPDUSize () const
- const PresentationContextAC * GetPresentationContextACByID (uint8_t id) const
- uint8_t GetPresentationContextIDFromPresentationContext (PresentationContextRQ const &pc) const

  return 0 upon error
- const PresentationContextRQ * GetPresentationContextRQByID (uint8_t id) const
- std::vector< PresentationContextRQ > const & GetPresentationContexts () const
- std::iostream * GetProtocol ()
- EStateID GetState () const
- ARTIMTimer & GetTimer ()
- bool InitializeConnection ()

  used to establish scu connections
- bool InitializeIncomingConnection ()

  used to establish scp connections
- void operator= (const ULConnection &)=delete
- void SetMaxPDUSize (uint32_t inSize)
- void SetPresentationContexts (const std::vector< PresentationContext > &inContexts)
- void SetPresentationContexts (const std::vector< PresentationContextRQ > &inContexts)
- void SetState (const EStateID &inState)
- void StopProtocol ()

Friends

- class ULActionAE6
- class ULConnectionManager

12.355.1 Detailed Description

ULConnection.

This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

The ULConnectionManager tells the ULConnection what data can actually be sent.

This class is done this way so that it can be eventually be replaced with a ULSecureConnection, if such a protocol is warranted, so that all data that passes through can be managed through a secure connection. For now, this class provides a simple pass-through mechanism to the socket itself.

So, for instance, a gdcm object will be passes to this object, and it will then get passed along the connection, if that connection is in the proper state to do so.

For right now, this class is not directly intended to be inherited from, but the potential for future ULSecureConnection warrants the addition, rather than having everything be managed from within the ULConnectionManager (or this class) without a wrapper.

12.355.2 Constructor & Destructor Documentation

12.355.2.1 ULConnection() [1/2]

```
gdcm::network::ULConnection::ULConnection (
            const ULConnectionInfo & inUserInformation)
```

Referenced by ULConnection(), and operator=().

12.355.2.2 ~ULConnection()

```
virtual gdcm::network::ULConnection::~ULConnection ()  [virtual]
```

12.355.2.3 ULConnection() [2/2]

```
gdcm::network::ULConnection::ULConnection (
            const ULConnection & )  [delete]
```

References ULConnection().

12.355.3 Member Function Documentation

12.355.3.1 AddAcceptedPresentationContext()

```
void gdcm::network::ULConnection::AddAcceptedPresentationContext (
            const PresentationContextAC & inPC)
```

12.355.3.2 FindContext()

```
PresentationContextRQ gdcm::network::ULConnection::FindContext (
            const DataElement & de) const
```

12.355.3.3 GetAcceptedPresentationContexts() [1/2]

```
std::vector< PresentationContextAC > & gdcm::network::ULConnection::GetAcceptedPresentation↩
Contexts ()
```

12.355.3.4 GetAcceptedPresentationContexts() [2/2]

```
std::vector< PresentationContextAC > const & gdcm::network::ULConnection::GetAcceptedPresentation↩
Contexts () const
```

12.355.3.5 GetConnectionInfo()

```
const ULConnectionInfo & gdcm::network::ULConnection::GetConnectionInfo () const
```

12.355.3.6 GetMaxPDUSize()

```
uint32_t gdcm::network::ULConnection::GetMaxPDUSize () const
```

12.355.3.7 GetPresentationContextACByID()

```
const PresentationContextAC * gdcm::network::ULConnection::GetPresentationContextACByID (
            uint8_t id) const
```

12.355.3.8 GetPresentationContextIDFromPresentationContext()

```
uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (
            PresentationContextRQ const & pc) const
```

return 0 upon error

12.355.3.9 GetPresentationContextRQByID()

```
const PresentationContextRQ * gdcm::network::ULConnection::GetPresentationContextRQByID (
            uint8_t id) const
```

12.355.3.10 GetPresentationContexts()

```
std::vector< PresentationContextRQ > const & gdcm::network::ULConnection::GetPresentationContexts
() const
```

12.355.3.11 GetProtocol()

```
std::iostream * gdcm::network::ULConnection::GetProtocol ()
```

12.355.3.12 GetState()

```
EStateID gdcm::network::ULConnection::GetState () const
```

12.355.3.13 GetTimer()

```
ARTIMTimer & gdcm::network::ULConnection::GetTimer ()
```

12.355.3.14 InitializeConnection()

```
bool gdcm::network::ULConnection::InitializeConnection ()
```

used to establish scu connections

12.355.3.15 InitializeIncomingConnection()

```
bool gdcm::network::ULConnection::InitializeIncomingConnection ()
```

used to establish scp connections

12.355.3.16 operator=()

```
void gdcm::network::ULConnection::operator= (
            const ULConnection & )  [delete]
```

References ULConnection().

12.355.3.17 SetMaxPDUSize()

```
void gdcm::network::ULConnection::SetMaxPDUSize (
            uint32_t inSize)
```

12.355.3.18 SetPresentationContexts() [1/2]

```
void gdcm::network::ULConnection::SetPresentationContexts (
            const std::vector< PresentationContext > & inContexts)
```

12.355.3.19 SetPresentationContexts() [2/2]

```
void gdcm::network::ULConnection::SetPresentationContexts (
            const std::vector< PresentationContextRQ > & inContexts)
```

12.355.3.20 SetState()

```
void gdcm::network::ULConnection::SetState (
            const EStateID & inState)
```

12.355.3.21 StopProtocol()

```
void gdcm::network::ULConnection::StopProtocol ()
```

12.355.4 Friends And Related Symbol Documentation

12.355.4.1 ULActionAE6

```
friend class ULActionAE6  [friend]
```

References ULActionAE6.

Referenced by ULActionAE6.

12.355.4.2 ULConnectionManager

```
friend class ULConnectionManager  [friend]
```

References ULConnectionManager.

Referenced by ULConnectionManager.

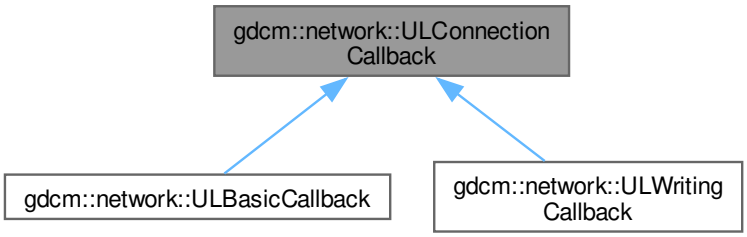
The documentation for this class was generated from the following file:

- gdcmULConnection.h

12.356 gdcm::network::ULConnectionCallback Class Reference

```
#include <gdcmULConnectionCallback.h>
```

Inheritance diagram for gdcm::network::ULConnectionCallback:



Public Member Functions

- ULConnectionCallback ()
- virtual ~ULConnectionCallback ()=default
- bool DataSetHandles () const
- virtual void HandleDataSet (const DataSet &inDataSet)=0
- virtual void HandleResponse (const DataSet &inDataSet)=0
- void ResetHandledDataSet ()
- void SetImplicitFlag (const bool imp)

Protected Member Functions

- void DataSetHandled ()

Protected Attributes

- bool mImplicit

12.356.1 Detailed Description

When a dataset comes back from a query/move/etc, the result can either be stored entirely in memory, or could be stored on disk. This class provides a mechanism to indicate what the ULConnectionManager should do with datasets that are produced through query results. The ULConnectionManager will call the HandleDataSet function during the course of receiving datasets. Particular implementations should fill in what that function does, including updating progress, etc. NOTE: since cmove requires that multiple event loops be employed, the callback function MUST set mHandledData↩Set to true. otherwise, the cmove event loop handler will not know data was received, and proceed to end the loop prematurely.

12.356.2 Constructor & Destructor Documentation

12.356.2.1 ULConnectionCallback()

```
gdcm::network::ULConnectionCallback::ULConnectionCallback () [inline]
```

References mImplicit.

12.356.2.2 ~ULConnectionCallback()

```
virtual gdcm::network::ULConnectionCallback::~ULConnectionCallback () [virtual], [default]
```

12.356.3 Member Function Documentation

12.356.3.1 DataSetHandled()

```
void gdcm::network::ULConnectionCallback::DataSetHandled () [inline], [protected]
```

12.356.3.2 DataSetHandles()

```
bool gdcm::network::ULConnectionCallback::DataSetHandles () const [inline]
```

12.356.3.3 HandleDataSet()

```
virtual void gdcm::network::ULConnectionCallback::HandleDataSet (
            const DataSet & inDataSet) [pure virtual]
```

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

12.356.3.4 HandleResponse()

```
virtual void gdcm::network::ULConnectionCallback::HandleResponse (
            const DataSet & inDataSet) [pure virtual]
```

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

12.356.3.5 ResetHandledDataSet()

```
void gdcm::network::ULConnectionCallback::ResetHandledDataSet () [inline]
```

12.356.3.6 SetImplicitFlag()

```
void gdcm::network::ULConnectionCallback::SetImplicitFlag (
            const bool imp)  [inline]
```

References mImplicit.

12.356.4 Member Data Documentation

12.356.4.1 mImplicit

```
bool gdcm::network::ULConnectionCallback::mImplicit  [protected]
```

Referenced by ULConnectionCallback(), and SetImplicitFlag().

The documentation for this class was generated from the following file:

- gdcmULConnectionCallback.h

12.357 gdcm::network::ULConnectionInfo Class Reference

ULConnectionInfo.

```
#include <gdcmULConnectionInfo.h>
```

Public Member Functions

- ULConnectionInfo ()
- const char ∗ GetCalledAETitle () const
- std::string GetCalledComputerName () const
- unsigned long GetCalledIPAddress () const
- int GetCalledIPPort () const
- const char ∗ GetCallingAETitle () const
- unsigned long GetMaxPDULength () const
- bool Initialize (UserInformation const &inUserInformation, const char ∗inCalledAETitle, const char ∗inCalling↩ AETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void SetMaxPDULength (unsigned long inMaxPDULength)

12.357.1 Detailed Description

ULConnectionInfo.

this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

12.357.2 Constructor & Destructor Documentation

12.357.2.1 ULConnectionInfo()

```
gdcm::network::ULConnectionInfo::ULConnectionInfo ()
```

12.357.3 Member Function Documentation

12.357.3.1 GetCalledAETitle()

```
const char * gdcm::network::ULConnectionInfo::GetCalledAETitle () const
```

12.357.3.2 GetCalledComputerName()

```
std::string gdcm::network::ULConnectionInfo::GetCalledComputerName () const
```

12.357.3.3 GetCalledIPAddress()

```
unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress () const
```

12.357.3.4 GetCalledIPPort()

```
int gdcm::network::ULConnectionInfo::GetCalledIPPort () const
```

12.357.3.5 GetCallingAETitle()

```
const char * gdcm::network::ULConnectionInfo::GetCallingAETitle () const
```

12.357.3.6 GetMaxPDULength()

```
unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength () const
```

12.357.3.7 Initialize()

```
bool gdcm::network::ULConnectionInfo::Initialize (
            UserInformation const & inUserInformation,
            const char * inCalledAETitle,
            const char * inCallingAETitle,
            unsigned long inCalledIPAddress,
            int inCalledIPPort,
            std::string inCalledComputerName)
```

12.357.3.8 SetMaxPDULength()

```
void gdcm::network::ULConnectionInfo::SetMaxPDULength (
            unsigned long inMaxPDULength)
```

The documentation for this class was generated from the following file:

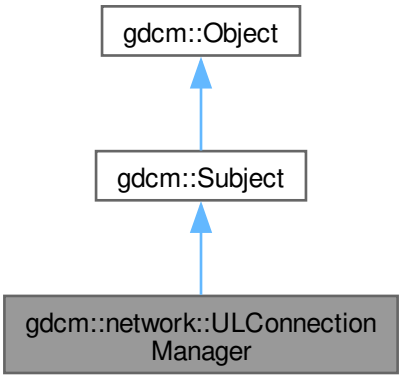
- gdcmULConnectionInfo.h

12.358 gdcm::network::ULConnectionManager Class Reference

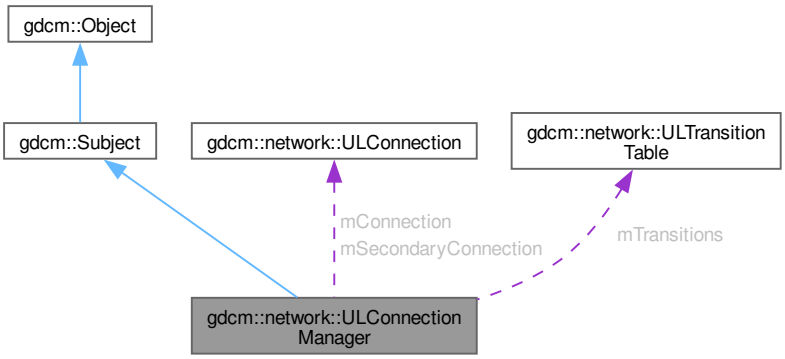
ULConnectionManager.

```
#include <gdcmULConnectionManager.h>
```

Inheritance diagram for gdcm::network::ULConnectionManager:



Collaboration diagram for gdcm::network::ULConnectionManager:



Public Member Functions

- ULConnectionManager ()
- ~ULConnectionManager () override
- bool BreakConnection (const double &inTimeout)
- void BreakConnectionNow ()
- bool EstablishConnection (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, std::vector< PresentationContext > const &pcVector)
- bool EstablishConnectionMove (const std::string &inAETitle, const std::string &inConnectAETitle, const std::string &inComputerName, long inIPAddress, uint16_t inConnectPort, double inTimeout, uint16_t inReturnPort, std::vector< PresentationContext > const &pcVector)
- std::vector< PresentationDataValue > SendEcho ()
- std::vector< DataSet > SendFind (const BaseRootQuery *inRootQuery)
- void SendFind (const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendMove (const BaseRootQuery *inRootQuery)
- bool SendMove (const BaseRootQuery *inRootQuery, ULConnectionCallback *inCallback)

  return false upon error
- std::vector< DataSet > SendNAction (const BaseQuery *inQuery)
- void SendNAction (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNCreate (const BaseQuery *inQuery)
- void SendNCreate (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNDelete (const BaseQuery *inQuery)
- void SendNDelete (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNEventReport (const BaseQuery *inQuery)
- void SendNEventReport (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNGet (const BaseQuery *inQuery)
- void SendNGet (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendNSet (const BaseQuery *inQuery)
- void SendNSet (const BaseQuery *inQuery, ULConnectionCallback *inCallback)
- std::vector< DataSet > SendStore (const File &file, std::istream *pStream=nullptr, std::streampos dataSetOffset=0)

- void SendStore (const File &file, ULConnectionCallback *inCallback, std::istream *pStream=nullptr, std::streampos dataSetOffset=0)

  callback based API

Public Member Functions inherited from gdcm::Subject

- Subject ()
- ~Subject () override
- unsigned long AddObserver (const Event &event, Command *)
- unsigned long AddObserver (const Event &event, Command *) const
- Command * GetCommand (unsigned long tag)
- bool HasObserver (const Event &event) const
- void InvokeEvent (const Event &)
- void InvokeEvent (const Event &) const
- void RemoveAllObservers ()
- void RemoveObserver (unsigned long tag)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Protected Member Functions

- ULConnectionManager (const ULConnectionManager &inCM)
- EStateID RunEventLoop (ULEvent &inEvent, ULConnection *inWhichConnection, ULConnectionCallback *inCallback, const bool &startWaiting)
- EStateID RunMoveEventLoop (ULEvent &inEvent, ULConnectionCallback *inCallback)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Protected Attributes

- ULConnection * mConnection
- ULConnection * mSecondaryConnection
- ULTransitionTable mTransitions

12.358.1 Detailed Description

ULConnectionManager.

The ULConnectionManager performs actions on the ULConnection given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

Its inputs are ULEvents, and it performs ULActions.

12.358.2 Constructor & Destructor Documentation

12.358.2.1 ULConnectionManager() [1/2]

```
gdcm::network::ULConnectionManager::ULConnectionManager (
            const ULConnectionManager & inCM)  [protected]
```

References ULConnectionManager().

Referenced by ULConnectionManager().

12.358.2.2 ULConnectionManager() [2/2]

```
gdcm::network::ULConnectionManager::ULConnectionManager ()
```

12.358.2.3 ~ULConnectionManager()

```
gdcm::network::ULConnectionManager::~ULConnectionManager ()  [override]
```

12.358.3 Member Function Documentation

12.358.3.1 BreakConnection()

```
bool gdcm::network::ULConnectionManager::BreakConnection (
            const double & inTimeout)
```

12.358.3.2 BreakConnectionNow()

```
void gdcm::network::ULConnectionManager::BreakConnectionNow ()
```

12.358.3.3 EstablishConnection()

```
bool gdcm::network::ULConnectionManager::EstablishConnection (
            const std::string & inAETitle,
            const std::string & inConnectAETitle,
            const std::string & inComputerName,
            long inIPAddress,
            uint16_t inConnectPort,
            double inTimeout,
            std::vector< PresentationContext > const & pcVector)
```

returns true if a connection of the given AETitle (ie, 'this' program) is able to connect to the given AETitle and Port in a certain amount of time providing the connection type will establish the proper exchange syntax with a server; if a different functionality is required, a different connection should be established. returns false if the connection type is 'move'– have to give a return port for move to work as specified.

12.358.3.4 EstablishConnectionMove()

```
bool gdcm::network::ULConnectionManager::EstablishConnectionMove (
            const std::string & inAETitle,
            const std::string & inConnectAETitle,
            const std::string & inComputerName,
            long inIPAddress,
            uint16_t inConnectPort,
            double inTimeout,
            uint16_t inReturnPort,
            std::vector< PresentationContext > const & pcVector)
```

returns true for above reasons, but contains the special 'move' port

12.358.3.5 RunEventLoop()

```
EStateID gdcm::network::ULConnectionManager::RunEventLoop (
            ULEvent & inEvent,
            ULConnection * inWhichConnection,
            ULConnectionCallback * inCallback,
            const bool & startWaiting)  [protected]
```

12.358.3.6 RunMoveEventLoop()

```
EStateID gdcm::network::ULConnectionManager::RunMoveEventLoop (
            ULEvent & inEvent,
            ULConnectionCallback * inCallback)  [protected]
```

12.358.3.7 SendEcho()

```
std::vector< PresentationDataValue > gdcm::network::ULConnectionManager::SendEcho ()
```

12.358.3.8 SendFind() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendFind (
            const BaseRootQuery * inRootQuery)
```

12.358.3.9 SendFind() [2/2]

```
void gdcm::network::ULConnectionManager::SendFind (
            const BaseRootQuery * inRootQuery,
            ULConnectionCallback * inCallback)
```

12.358.3.10 SendMove() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendMove (
            const BaseRootQuery * inRootQuery)
```

12.358.3.11 SendMove() [2/2]

```
bool gdcm::network::ULConnectionManager::SendMove (
            const BaseRootQuery * inRootQuery,
            ULConnectionCallback * inCallback)
```

return false upon error

12.358.3.12 SendNAction() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNAction (
            const BaseQuery * inQuery)
```

12.358.3.13 SendNAction() [2/2]

```
void gdcm::network::ULConnectionManager::SendNAction (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback)
```

12.358.3.14 SendNCreate() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNCreate (
            const BaseQuery * inQuery)
```

12.358.3.15 SendNCreate() [2/2]

```
void gdcm::network::ULConnectionManager::SendNCreate (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback)
```

12.358.3.16 SendNDelete() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNDelete (
            const BaseQuery * inQuery)
```

12.358.3.17 SendNDelete() [2/2]

```
void gdcm::network::ULConnectionManager::SendNDelete (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback)
```

12.358.3.18 SendNEventReport() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNEventReport (
            const BaseQuery * inQuery)
```

12.358.3.19 SendNEventReport() [2/2]

```
void gdcm::network::ULConnectionManager::SendNEventReport (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback)
```

12.358.3.20 SendNGet() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNGet (
            const BaseQuery * inQuery)
```

12.358.3.21 SendNGet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNGet (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback)
```

12.358.3.22 SendNSet() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendNSet (
            const BaseQuery * inQuery)
```

12.358.3.23 SendNSet() [2/2]

```
void gdcm::network::ULConnectionManager::SendNSet (
            const BaseQuery * inQuery,
            ULConnectionCallback * inCallback)
```

12.358.3.24 SendStore() [1/2]

```
std::vector< DataSet > gdcm::network::ULConnectionManager::SendStore (
            const File & file,
            std::istream * pStream = nullptr,
            std::streampos dataSetOffset = 0)
```

12.358.3.25 SendStore() [2/2]

```
void gdcm::network::ULConnectionManager::SendStore (
            const File & file,
            ULConnectionCallback * inCallback,
            std::istream * pStream = nullptr,
            std::streampos dataSetOffset = 0)
```

callback based API

12.358.4 Member Data Documentation

12.358.4.1 mConnection

```
ULConnection* gdcm::network::ULConnectionManager::mConnection  [protected]
```

12.358.4.2 mSecondaryConnection

```
ULConnection* gdcm::network::ULConnectionManager::mSecondaryConnection  [protected]
```

12.358.4.3 mTransitions

```
ULTransitionTable gdcm::network::ULConnectionManager::mTransitions  [protected]
```

The documentation for this class was generated from the following file:

- gdcmULConnectionManager.h

12.359 gdcm::network::ULEvent Class Reference

ULEvent.

```
#include <gdcmULEvent.h>
```

Public Member Functions

- ULEvent (const EEventID &inEventID, BasePDU *inBasePDU, std::istream *iStream=nullptr, std::streampos posDataSet=0)
- ULEvent (const EEventID &inEventID, std::vector< BasePDU * > inBasePDU, std::istream *iStream=nullptr, std::streampos posDataSet=0)
- ~ULEvent ()
- std::streampos GetDataSetPos () const
- EEventID GetEvent () const
- std::istream * GetIStream () const
- std::vector< BasePDU * > const & GetPDUs () const
- void SetEvent (const EEventID &inEvent)
- void SetPDU (std::vector< BasePDU * > const &inPDU)

12.359.1 Detailed Description

ULEvent.

base class for network events.

An event consists of the event ID and the data associated with that event.

Note that once a PDU is created, it is now the responsibility of the associated event to destroy it!

12.359.2 Constructor & Destructor Documentation

12.359.2.1 ULEvent() [1/2]

```
gdcm::network::ULEvent::ULEvent (
            const EEventID & inEventID,
            std::vector< BasePDU * > inBasePDU,
            std::istream * iStream = nullptr,
            std::streampos posDataSet = 0)  [inline]
```

12.359.2.2 ULEvent() [2/2]

```
gdcm::network::ULEvent::ULEvent (
            const EEventID & inEventID,
            BasePDU * inBasePDU,
            std::istream * iStream = nullptr,
            std::streampos posDataSet = 0)  [inline]
```

12.359.2.3 ~ULEvent()

```
gdcm::network::ULEvent::~ULEvent () [inline]
```

12.359.3 Member Function Documentation

12.359.3.1 GetDataSetPos()

```
std::streampos gdcm::network::ULEvent::GetDataSetPos () const [inline]
```

12.359.3.2 GetEvent()

```
EEventID gdcm::network::ULEvent::GetEvent () const [inline]
```

12.359.3.3 GetIStream()

```
std::istream * gdcm::network::ULEvent::GetIStream () const [inline]
```

12.359.3.4 GetPDUs()

```
std::vector< BasePDU * > const & gdcm::network::ULEvent::GetPDUs () const [inline]
```

12.359.3.5 SetEvent()

```
void gdcm::network::ULEvent::SetEvent (
            const EEventID & inEvent) [inline]
```

12.359.3.6 SetPDU()

```
void gdcm::network::ULEvent::SetPDU (
            std::vector< BasePDU * > const & inPDU) [inline]
```

The documentation for this class was generated from the following file:

- gdcmULEvent.h

12.360 gdcm::network::ULTransitionTable Class Reference

ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

```
#include <gdcmULTransitionTable.h>
```

Public Member Functions

- ULTransitionTable ()
- void HandleEvent (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent) const
- void PrintTable () const

12.360.1 Detailed Description

ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Based roughly on the solutions in player2.cpp in the boost examples and this so question: `http://stackoverflow.com/questions/1647631/c-state-machine-design`

The transition table is constructed of TableRows. Each row is based on an event, and an event handler in the TransitionTable object takes a given event, and then finds the given row.

Then, given the current state of the connection, determines the appropriate action to take and then the state to transition to next.

12.360.2 Constructor & Destructor Documentation

12.360.2.1 ULTransitionTable()

```
gdcm::network::ULTransitionTable::ULTransitionTable ()
```

12.360.3 Member Function Documentation

12.360.3.1 HandleEvent()

```
void gdcm::network::ULTransitionTable::HandleEvent (
            Subject * s,
            ULEvent & inEvent,
            ULConnection & inConnection,
            bool & outWaitingForEvent,
            EEventID & outRaisedEvent) const
```

12.360.3.2 PrintTable()

```
void gdcm::network::ULTransitionTable::PrintTable () const
```

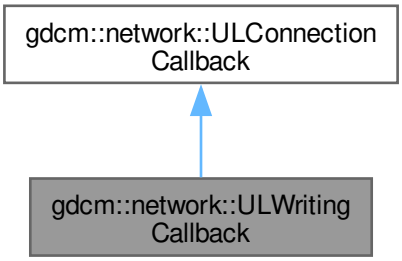
The documentation for this class was generated from the following file:

- gdcmULTransitionTable.h

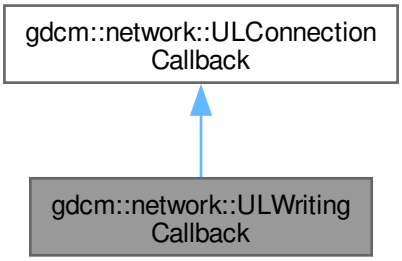
12.361 gdcm::network::ULWritingCallback Class Reference

```
#include <gdcmULWritingCallback.h>
```

Inheritance diagram for gdcm::network::ULWritingCallback:



Collaboration diagram for gdcm::network::ULWritingCallback:



Public Member Functions

- ULWritingCallback ()=default
- ~ULWritingCallback () override=default
- void HandleDataSet (const DataSet &inDataSet) override
- void HandleResponse (const DataSet &inDataSet) override
- void SetDirectory (const std::string &inDirectoryName)

  provide the directory into which all files are written.

Public Member Functions inherited from gdcm::network::ULConnectionCallback

- ULConnectionCallback ()
- virtual ~ULConnectionCallback ()=default
- bool DataSetHandles () const
- void ResetHandledDataSet ()
- void SetImplicitFlag (const bool imp)

Additional Inherited Members

Protected Member Functions inherited from gdcm::network::ULConnectionCallback

- void DataSetHandled ()

Protected Attributes inherited from gdcm::network::ULConnectionCallback

- bool mImplicit

12.361.1 Constructor & Destructor Documentation

12.361.1.1 ULWritingCallback()

```
gdcm::network::ULWritingCallback::ULWritingCallback () [default]
```

12.361.1.2 ~ULWritingCallback()

```
gdcm::network::ULWritingCallback::~ULWritingCallback () [override], [default]
```

12.361.2 Member Function Documentation

12.361.2.1 HandleDataSet()

```
void gdcm::network::ULWritingCallback::HandleDataSet (
            const DataSet & inDataSet) [override], [virtual]
```

Implements gdcm::network::ULConnectionCallback.

12.361.2.2 HandleResponse()

```
void gdcm::network::ULWritingCallback::HandleResponse (
            const DataSet & inDataSet) [override], [virtual]
```

Implements gdcm::network::ULConnectionCallback.

12.361.2.3 SetDirectory()

```
void gdcm::network::ULWritingCallback::SetDirectory (
            const std::string & inDirectoryName)  [inline]
```

provide the directory into which all files are written.

The documentation for this class was generated from the following file:

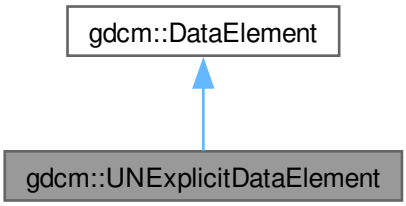
- gdcmULWritingCallback.h

12.362 gdcm::UNExplicitDataElement Class Reference

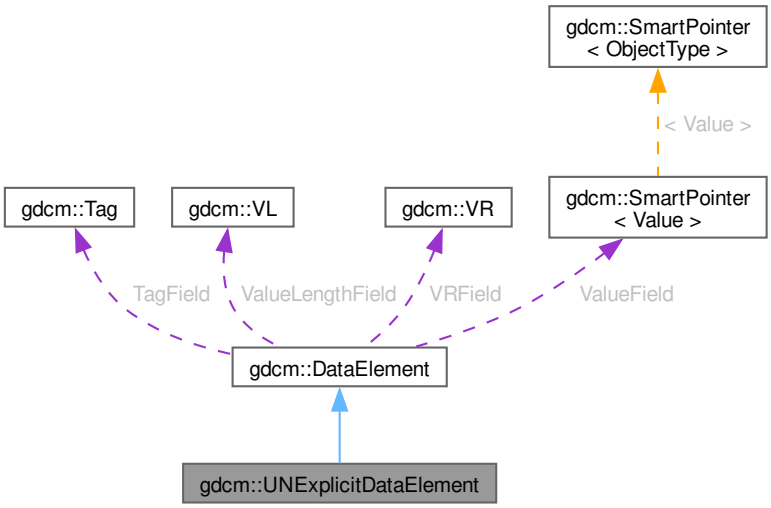
Class to read/write a DataElement as UNExplicit Data Element.

```
#include <gdcmUNExplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitDataElement:



Collaboration diagram for gdcm::UNExplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const

- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.362.1 Detailed Description

Class to read/write a DataElement as UNExplicit Data Element.

Note

bla

12.362.2 Member Function Documentation

12.362.2.1 GetLength()

```
VL gdcm::UNExplicitDataElement::GetLength () const
```

12.362.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitDataElement::Read (
            std::istream & is)
```

12.362.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitDataElement::ReadPreValue (
            std::istream & is)
```

12.362.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitDataElement::ReadValue (
            std::istream & is,
            bool readvalues = true)
```

12.362.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitDataElement::ReadWithLength (
            std::istream & is,
            VL & length)
```

The documentation for this class was generated from the following file:

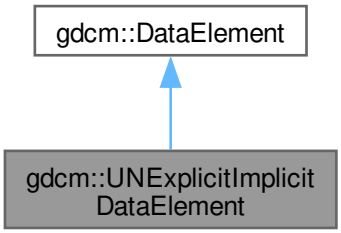
- gdcmUNExplicitDataElement.h

12.363 gdcm::UNExplicitImplicitDataElement Class Reference

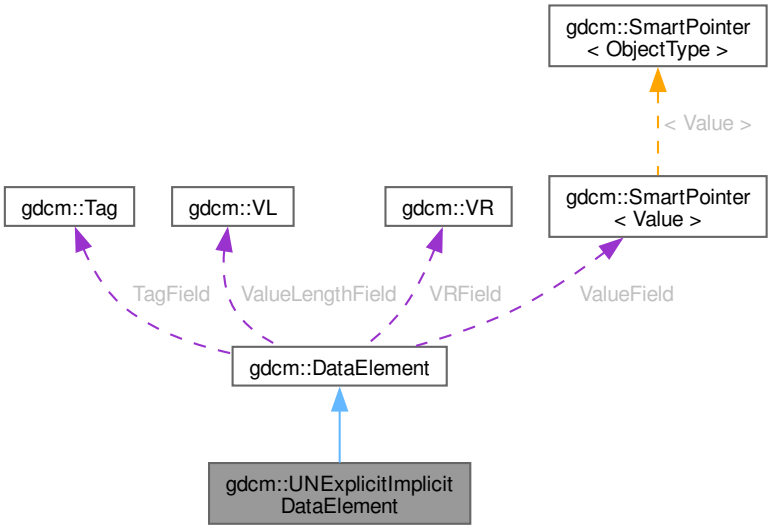
Class to read/write a DataElement as ExplicitImplicit Data Element.

```
#include <gdcmUNExplicitImplicitDataElement.h>
```

Inheritance diagram for gdcm::UNExplicitImplicitDataElement:



Collaboration diagram for gdcm::UNExplicitImplicitDataElement:



Public Member Functions

- VL GetLength () const
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is)

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

Get Tag.

- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.363.1 Detailed Description

Class to read/write a DataElement as ExplicitImplicit Data Element.

This class gather two known bugs:

1. GDCM 1.2.0 would rewrite VR=UN Value Length on 2 bytes instead of 4 bytes
2. GDCM 1.2.0 would also rewrite DataElement as Implicit when the VR would not be known this would only happen in some very rare cases. gdcm 2.X design could handle bug #1 or #2 exclusively, this class can now handle file which have both issues. See: gdcmData/TheralysGDCM120Bug.dcm

12.363.2 Member Function Documentation

12.363.2.1 GetLength()

```
VL gdcm::UNExplicitImplicitDataElement::GetLength () const
```

12.363.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitImplicitDataElement::Read (
            std::istream & is)
```

12.363.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitImplicitDataElement::ReadPreValue (
            std::istream & is)
```

12.363.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::UNExplicitImplicitDataElement::ReadValue (
            std::istream & is)
```

The documentation for this class was generated from the following file:

- gdcmUNExplicitImplicitDataElement.h

12.364 gdcm::Unpacker12Bits Class Reference

Pack/Unpack 12 bits pixel into 16bits.

```
#include <gdcmUnpacker12Bits.h>
```

Static Public Member Functions

- static bool Pack (char ∗out, const char ∗in, size_t n)
- static bool Unpack (char ∗out, const char ∗in, size_t n)

12.364.1 Detailed Description

Pack/Unpack 12 bits pixel into 16bits.

- You can only pack an even number of 16bits, which means a multiple of 4 (expressed in bytes)
- You can only unpack a multiple of 3 bytes

This class has no purpose in general purpose DICOM implementation. However to be able to cope with some early ACR-NEMA file generated by a well-known private vendor, one would need to unpack 12bits Stored Pixel Value into a more standard 16bits Stored Pixel Value.

See also

Rescaler

12.364.2 Member Function Documentation

12.364.2.1 Pack()

```
bool gdcm::Unpacker12Bits::Pack (
            char * out,
            const char * in,
            size_t n)  [static]
```

Pack an array of 16bits where all values are 12bits into a pack form. n is the length in bytes of array in, out will be a fake 8bits array of size (n / 2) ∗ 3

12.364.2.2 Unpack()

```
bool gdcm::Unpacker12Bits::Unpack (
            char * out,
            const char * in,
            size_t n)  [static]
```

Unpack an array of 'packed' 12bits data into a more conventional 16bits array. n is the length in bytes of array in, out will be a 16bits array of size (n / 3) ∗ 2

The documentation for this class was generated from the following file:

- gdcmUnpacker12Bits.h

12.365 gdcm::Usage Class Reference

Usage.

```
#include <gdcmUsage.h>
```

Public Types

- enum UsageType {
  Mandatory ,
  Conditional ,
  UserOption ,
  Invalid }

Public Member Functions

- Usage (UsageType type=Invalid)
- operator UsageType () const

Static Public Member Functions

- static const char ∗ GetUsageString (UsageType type)
- static UsageType GetUsageType (const char ∗type)

Friends

- std::ostream & operator<< (std::ostream &os, const Usage &vr)

12.365.1 Detailed Description

Usage.

Note

A.1.3 IOD Module Table and Functional Group Macro Table This Section of each IOD defines in a tabular form the Modules comprising the IOD. The following information must be specified for each Module in the table:

- The name of the Module or Functional Group
- A reference to the Section in Annex C which defines the Module or Functional Group
- The usage of the Module or Functional Group; whether it is:
- Mandatory (see A.1.3.1) , abbreviated M
- Conditional (see A.1.3.2) , abbreviated C
- User Option (see A.1.3.3) , abbreviated U The Modules referenced are defined in Annex C. A.1.3.1 MANDATORY MODULES For each IOD, Mandatory Modules shall be supported per the definitions, semantics and requirements defined in Annex C.

A.1.3.2 CONDITIONAL MODULES Conditional Modules are Mandatory Modules if specific conditions are met. If the specified conditions are not met, this Module shall not be supported; that is, no information defined in that Module shall be sent. A.1.3.3 USER OPTION MODULES User Option Modules may or may not be supported. If an optional Module is supported, the Attribute Types specified in the Modules in Annex C shall be supported.

12.365.2 Member Enumeration Documentation

12.365.2.1 UsageType

```
enum gdcm::Usage::UsageType
```

Enumerator

Mandatory	
Conditional	
UserOption	
Invalid	

12.365.3 Constructor & Destructor Documentation

12.365.3.1 Usage()

```
gdcm::Usage::Usage (
            UsageType type = Invalid)  [inline]
```

References Invalid.

Referenced by operator<<.

12.365.4 Member Function Documentation

12.365.4.1 GetUsageString()

```
const char * gdcm::Usage::GetUsageString (
            UsageType type)  [static]
```

Referenced by operator<<.

12.365.4.2 GetUsageType()

```
UsageType gdcm::Usage::GetUsageType (
            const char * type)  [static]
```

12.365.4.3 operator UsageType()

```
gdcm::Usage::operator UsageType () const  [inline]
```

12.365.5 Friends And Related Symbol Documentation

12.365.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Usage & vr)  [friend]
```

References Usage(), and GetUsageString().

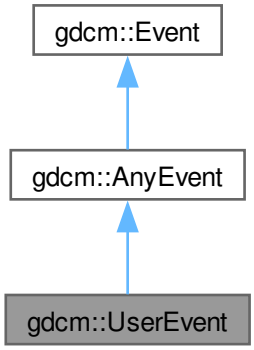
The documentation for this class was generated from the following file:

- gdcmUsage.h

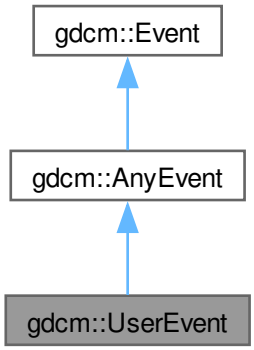
12.366 gdcm::UserEvent Class Reference

```
#include <gdcmEvent.h>
```

Inheritance diagram for gdcm::UserEvent:



Collaboration diagram for gdcm::UserEvent:



Additional Inherited Members

Public Member Functions inherited from gdcm::Event

- Event ()

- Event (const Event &)
- virtual ~Event ()
- virtual bool CheckEvent (const Event *) const =0
- virtual const char * GetEventName () const =0
- virtual Event * MakeObject () const =0
- void operator= (const Event &)=delete
- virtual void Print (std::ostream &os) const

The documentation for this class was generated from the following file:

- gdcmEvent.h

12.367 gdcm::network::UserInformation Class Reference

UserInformation.

```
#include <gdcmUserInformation.h>
```

Public Member Functions

- UserInformation ()
- UserInformation (const UserInformation &)=delete
- ~UserInformation ()
- void AddRoleSelectionSub (RoleSelectionSub const &r)
- void AddSOPClassExtendedNegociationSub (SOPClassExtendedNegociationSub const &s)
- MaximumLengthSub & GetMaximumLengthSub ()
- const MaximumLengthSub & GetMaximumLengthSub () const
- UserInformation & operator= (const UserInformation &)
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

12.367.1 Detailed Description

UserInformation.

Table 9-16 USER INFORMATION ITEM FIELDS

TODO what is the goal of :

Table 9-20 USER INFORMATION ITEM FIELDS

12.367.2 Constructor & Destructor Documentation

12.367.2.1 UserInformation() [1/2]

```
gdcm::network::UserInformation::UserInformation ()
```

Referenced by UserInformation(), and operator=().

12.367.2.2 ~UserInformation()

```
gdcm::network::UserInformation::~UserInformation ()
```

12.367.2.3 UserInformation() [2/2]

```
gdcm::network::UserInformation::UserInformation (
            const UserInformation & )  [delete]
```

References UserInformation().

12.367.3 Member Function Documentation

12.367.3.1 AddRoleSelectionSub()

```
void gdcm::network::UserInformation::AddRoleSelectionSub (
            RoleSelectionSub const & r)
```

12.367.3.2 AddSOPClassExtendedNegociationSub()

```
void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (
            SOPClassExtendedNegociationSub const & s)
```

12.367.3.3 GetMaximumLengthSub() [1/2]

```
MaximumLengthSub & gdcm::network::UserInformation::GetMaximumLengthSub ()  [inline]
```

12.367.3.4 GetMaximumLengthSub() [2/2]

```
const MaximumLengthSub & gdcm::network::UserInformation::GetMaximumLengthSub () const  [inline]
```

12.367.3.5 operator=()

```
UserInformation & gdcm::network::UserInformation::operator= (
            const UserInformation & )
```

References UserInformation().

12.367.3.6 Print()

```
void gdcm::network::UserInformation::Print (
            std::ostream & os) const
```

12.367.3.7 Read()

```
std::istream & gdcm::network::UserInformation::Read (
            std::istream & is)
```

12.367.3.8 Size()

```
size_t gdcm::network::UserInformation::Size () const
```

12.367.3.9 Write()

```
const std::ostream & gdcm::network::UserInformation::Write (
            std::ostream & os) const
```

The documentation for this class was generated from the following file:

- gdcmUserInformation.h

12.368 gdcm::UUIDGenerator Class Reference

Class for generating unique UUID.

```
#include <gdcmUUIDGenerator.h>
```

Public Member Functions

- const char * Generate ()

Static Public Member Functions

- static bool IsValid (const char *uid)

  Find out if the string is a valid UUID or not.

12.368.1 Detailed Description

Class for generating unique UUID.

generate DCE 1.1 uid

12.368.2 Member Function Documentation

12.368.2.1 Generate()

```
const char * gdcm::UUIDGenerator::Generate ()
```

Return the generated uuid NOT THREAD SAFE

12.368.2.2 IsValid()

```
bool gdcm::UUIDGenerator::IsValid (
            const char * uid)  [static]
```

Find out if the string is a valid UUID or not.

The documentation for this class was generated from the following file:

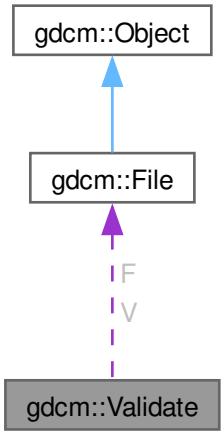
- gdcmUUIDGenerator.h

12.369 gdcm::Validate Class Reference

Validate class.

```
#include <gdcmValidate.h>
```

Collaboration diagram for gdcm::Validate:



Public Member Functions

- Validate ()
- ~Validate ()
- const File & GetValidatedFile ()
- void SetFile (File const &f)
- void Validation ()

Protected Attributes

- const File * F
- File V

12.369.1 Detailed Description

Validate class.

12.369.2 Constructor & Destructor Documentation

12.369.2.1 Validate()

```
gdcm::Validate::Validate ()
```

12.369.2.2 ~Validate()

```
gdcm::Validate::~Validate ()
```

12.369.3 Member Function Documentation

12.369.3.1 GetValidatedFile()

```
const File & gdcm::Validate::GetValidatedFile ()  [inline]
```

References V.

12.369.3.2 SetFile()

```
void gdcm::Validate::SetFile (
            File const & f)  [inline]
```

References F.

12.369.3.3 Validation()

```
void gdcm::Validate::Validation ()
```

12.369.4 Member Data Documentation

12.369.4.1 F

```
const File* gdcm::Validate::F  [protected]
```

Referenced by SetFile().

12.369.4.2 V

```
File gdcm::Validate::V  [protected]
```

Referenced by GetValidatedFile().

The documentation for this class was generated from the following file:

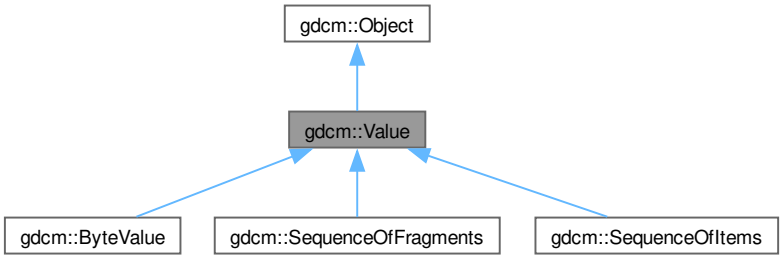
- gdcmValidate.h

12.370 gdcm::Value Class Reference

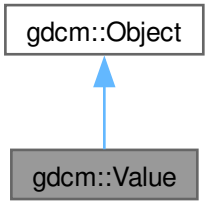
Class to represent the value of a Data Element.

```
#include <gdcmValue.h>
```

Inheritance diagram for gdcm::Value:



Collaboration diagram for gdcm::Value:



Public Member Functions

- Value ()=default
- ~Value () override=default
- virtual void Clear ()=0
- virtual VL GetLength () const =0
- virtual bool operator== (const Value &val) const =0
- virtual void SetLength (VL l)=0

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)
- virtual void Print (std::ostream &) const

Protected Member Functions

- virtual void SetLengthOnly (VL l)

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Friends

- class DataElement

12.370.1 Detailed Description

Class to represent the value of a Data Element.

Note

VALUE: A component of a Value Field. A Value Field may consist of one or more of these components.

12.370.2 Constructor & Destructor Documentation

12.370.2.1 Value()

```
gdcm::Value::Value () [default]
```

Referenced by gdcm::ByteValue::operator==(), gdcm::SequenceOfFragments::operator==(), gdcm::SequenceOfItems::operator==(), and operator==().

12.370.2.2 ~Value()

```
gdcm::Value::~Value () [override], [default]
```

12.370.3 Member Function Documentation

12.370.3.1 Clear()

```
virtual void gdcm::Value::Clear ()  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfFragments, and gdcm::SequenceOfItems.

12.370.3.2 GetLength()

```
virtual VL gdcm::Value::GetLength () const  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfFragments, and gdcm::SequenceOfItems.

Referenced by gdcm::DataSet::InsertDataElement(), and gdcm::DataElement::SetValue().

12.370.3.3 operator==()

```
virtual bool gdcm::Value::operator== (
            const Value & val) const  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfFragments, and gdcm::SequenceOfItems.

References Value().

12.370.3.4 SetLength()

```
virtual void gdcm::Value::SetLength (
            VL l)  [pure virtual]
```

Implemented in gdcm::ByteValue, gdcm::SequenceOfFragments, and gdcm::SequenceOfItems.

12.370.3.5 SetLengthOnly()

```
virtual void gdcm::Value::SetLengthOnly (
            VL l)  [protected], [virtual]
```

Reimplemented in gdcm::ByteValue.

12.370.4 Friends And Related Symbol Documentation

12.370.4.1 DataElement

```
friend class DataElement  [friend]
```

References DataElement.

Referenced by DataElement.

The documentation for this class was generated from the following file:

- gdcmValue.h

12.371 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference

Class to dispatch template calls.

```
#include <gdcmValueIO.h>
```

Static Public Member Functions

- static std::istream & Read (std::istream &is, Value &v, bool readvalues)
- static const std::ostream & Write (std::ostream &os, const Value &v)

12.371.1 Detailed Description

template<typename TDE, typename TSwap, typename TType = uint8_t>
class gdcm::ValueIO< TDE, TSwap, TType >

Class to dispatch template calls.

12.371.2 Member Function Documentation

12.371.2.1 Read()

```
template<typename TDE, typename TSwap, typename TType = uint8_t>
std::istream & gdcm::ValueIO< TDE, TSwap, TType >::Read (
            std::istream & is,
            Value & v,
            bool readvalues)  [static]
```

12.371.2.2 Write()

```
template<typename TDE, typename TSwap, typename TType = uint8_t>
const std::ostream & gdcm::ValueIO< TDE, TSwap, TType >::Write (
            std::ostream & os,
            const Value & v)  [static]
```

The documentation for this class was generated from the following file:

- gdcmValueIO.h

12.372 gdcm::MrProtocol::Vector3 Struct Reference

```
#include <gdcmMrProtocol.h>
```

Public Attributes

- double dCor
- double dSag
- double dTra

12.372.1 Member Data Documentation

12.372.1.1 dCor

```
double gdcm::MrProtocol::Vector3::dCor
```

12.372.1.2 dSag

```
double gdcm::MrProtocol::Vector3::dSag
```

12.372.1.3 dTra

```
double gdcm::MrProtocol::Vector3::dTra
```

The documentation for this struct was generated from the following file:

- gdcmMrProtocol.h

12.373 gdcm::Version Class Reference

major/minor and build version

```
#include <gdcmVersion.h>
```

Public Member Functions

- Version ()=default
- ~Version ()=default
- void Print (std::ostream &os=std::cout) const

Static Public Member Functions

- static int GetBuildVersion ()
- static int GetMajorVersion ()
- static int GetMinorVersion ()
- static const char * GetVersion ()

Friends

- std::ostream & operator<< (std::ostream &_os, const Version &v)

12.373.1 Detailed Description

major/minor and build version

12.373.2 Constructor & Destructor Documentation

12.373.2.1 Version()

```
gdcm::Version::Version () [default]
```

Referenced by operator<<.

12.373.2.2 ~Version()

```
gdcm::Version::~Version () [default]
```

12.373.3 Member Function Documentation

12.373.3.1 GetBuildVersion()

```
int gdcm::Version::GetBuildVersion () [static]
```

12.373.3.2 GetMajorVersion()

```
int gdcm::Version::GetMajorVersion () [static]
```

12.373.3.3 GetMinorVersion()

```
int gdcm::Version::GetMinorVersion () [static]
```

12.373.3.4 GetVersion()

```
const char * gdcm::Version::GetVersion () [static]
```

12.373.3.5 Print()

```
void gdcm::Version::Print (
            std::ostream & os = std::cout) const
```

Referenced by operator<<.

12.373.4 Friends And Related Symbol Documentation

12.373.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & _os,
            const Version & v) [friend]
```

References Version(), and Print().

The documentation for this class was generated from the following file:

- gdcmVersion.h

12.374 gdcm::VL Class Reference

Value Length.

```
#include <gdcmVL.h>
```

Public Types

- typedef uint32_t Type

Public Member Functions

- VL (uint32_t vl=0)
- VL GetLength () const
- bool IsOdd () const

  Return whether or not the VL is odd or not.
- bool IsUndefined () const
- operator uint32_t () const
- VL & operator++ ()
- VL operator++ (int)
- VL & operator+= (VL const &vl)

  += operator
- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & Read16 (std::istream &is)
- void SetToUndefined ()
- template<typename TSwap>
  const std::ostream & Write (std::ostream &os) const
- template<typename TSwap>
  const std::ostream & Write16 (std::ostream &os) const

Static Public Member Functions

- static uint16_t GetVL16Max ()
- static uint32_t GetVL32Max ()

Friends

- std::ostream & operator<< (std::ostream &os, const VL &vl)

12.374.1 Detailed Description

Value Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

Examples

BasicImageAnonymizer.cs, DecompressImage.cs, ReadAndDumpDICOMDIR2.cxx, and rle2img.cxx.

12.374.2 Member Typedef Documentation

12.374.2.1 Type

```
typedef uint32_t gdcm::VL::Type
```

12.374.3 Constructor & Destructor Documentation

12.374.3.1 VL()

```
gdcm::VL::VL (
            uint32_t vl = 0)  [inline]
```

Referenced by GetLength(), operator++(), operator++(), operator+=(), and operator<<.

12.374.4 Member Function Documentation

12.374.4.1 GetLength()

```
VL gdcm::VL::GetLength () const  [inline]
```

Examples

ReadAndDumpDICOMDIR2.cxx.

References VL().

Referenced by gdcm::FileMetaInformation::GetFullLength(), and gdcm::Item::Write().

12.374.4.2 GetVL16Max()

```
uint16_t gdcm::VL::GetVL16Max () [inline], [static]
```

12.374.4.3 GetVL32Max()

```
uint32_t gdcm::VL::GetVL32Max () [inline], [static]
```

12.374.4.4 IsOdd()

```
bool gdcm::VL::IsOdd () const [inline]
```

Return whether or not the VL is odd or not.

References IsUndefined().

Referenced by Write(), and Write16().

12.374.4.5 IsUndefined()

```
bool gdcm::VL::IsUndefined () const [inline]
```

Referenced by IsOdd().

12.374.4.6 operator uint32_t()

```
gdcm::VL::operator uint32_t () const [inline]
```

12.374.4.7 operator++() [1/2]

```
VL & gdcm::VL::operator++ () [inline]
```

References VL().

12.374.4.8 operator++() [2/2]

```
VL gdcm::VL::operator++ (
            int ) [inline]
```

References VL().

12.374.4.9 operator+=()

```
VL & gdcm::VL::operator+= (
            VL const & vl)  [inline]
```

+= operator

References VL().

12.374.4.10 Read()

```
template<typename TSwap>
std::istream & gdcm::VL::Read (
            std::istream & is)  [inline]
```

12.374.4.11 Read16()

```
template<typename TSwap>
std::istream & gdcm::VL::Read16 (
            std::istream & is)  [inline]
```

12.374.4.12 SetToUndefined()

```
void gdcm::VL::SetToUndefined ()  [inline]
```

12.374.4.13 Write()

```
template<typename TSwap>
const std::ostream & gdcm::VL::Write (
            std::ostream & os) const  [inline]
```

References IsOdd().

Referenced by gdcm::Fragment::Write(), gdcm::Item::Write(), gdcm::SequenceOfFragments::Write(), and gdcm::SequenceOfItems::Write(

12.374.4.14 Write16()

```
template<typename TSwap>
const std::ostream & gdcm::VL::Write16 (
            std::ostream & os) const  [inline]
```

References IsOdd().

12.374.5 Friends And Related Symbol Documentation

12.374.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const VL & vl)  [friend]
```

References VL().

The documentation for this class was generated from the following file:

- gdcmVL.h

12.375 gdcm::VM Class Reference

Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

```
#include <gdcmVM.h>
```

Public Types

- enum VMType {
  VM0 = 0 ,
  VM1 = 1 ,
  VM2 = 2 ,
  VM3 = 4 ,
  VM4 = 8 ,
  VM5 = 16 ,
  VM6 = 32 ,
  VM8 = 64 ,
  VM9 = 128 ,
  VM10 = 256 ,
  VM12 = 512 ,
  VM16 = 1024 ,
  VM18 = 2048 ,
  VM24 = 4096 ,
  VM28 = 8192 ,
  VM32 = 16384 ,
  VM35 = 32768 ,
  VM99 = 65536 ,
  VM256 = 131072 ,
  VM1_2 = VM1 | VM2 ,
  VM1_3 = VM1 | VM2 | VM3 ,
  VM1_4 = VM1 | VM2 | VM3 | VM4 ,
  VM1_5 = VM1 | VM2 | VM3 | VM4 | VM5 ,
  VM1_8 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 ,
  VM1_32 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 ,

VM1_99 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 ,
VM1_n = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM2_2n = VM2 | VM4 | VM6 | VM8 | VM16 | VM24 | VM32 | VM256 ,
VM2_n = VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM3_4 = VM3 | VM4 ,
VM3_3n = VM3 | VM6 | VM9 | VM24 | VM99 | VM256 ,
VM3_n = VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM4_4n = VM4 | VM16 | VM24 | VM32 | VM256 ,
VM6_6n = VM6 | VM12 | VM18 | VM24 ,
VM6_n = VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256 ,
VM7_7n ,
VM30_30n ,
VM47_47n ,
VM_END = VM1_n + 1 }

Public Member Functions

- VM (VMType type=VM0)
- bool Compatible (VM const &vm) const
- unsigned int GetLength () const
- operator VMType () const

Static Public Member Functions

- static size_t GetNumberOfElementsFromArray (const char ∗array, size_t length)
- static const char ∗ GetVMString (VMType vm)
- static VMType GetVMType (const char ∗vm)
- static VMType GetVMTypeFromLength (size_t length, unsigned int size)
- static bool IsValid (int vm1, VMType vm2)

Static Protected Member Functions

- static unsigned int GetIndex (VMType vm)

Friends

- std::ostream & operator<< (std::ostream &os, const VM &vm)

12.375.1 Detailed Description

Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Some private dict define some more: 4-4n 1-4 1-5 256 9 3-4

even more:

7-7n 10 18 12 35 47_47n 30_30n 28

6-6n

12.375.2 Member Enumeration Documentation

12.375.2.1 VMType

enum gdcm::VM::VMType

Enumerator

VM0	
VM1	
VM2	
VM3	
VM4	
VM5	
VM6	
VM8	
VM9	
VM10	
VM12	
VM16	
VM18	
VM24	
VM28	
VM32	
VM35	
VM99	
VM256	
VM1_2	
VM1_3	
VM1_4	
VM1_5	
VM1_8	
VM1_32	
VM1_99	
VM1_n	
VM2_2n	

VM2_n	
VM3_4	
VM3_3n	
VM3_n	
VM4_4n	
VM6_6n	
VM6_n	
VM7_7n	
VM30_30n	
VM47_47n	
VM_END	

12.375.3 Constructor & Destructor Documentation

12.375.3.1 VM()

```
gdcm::VM::VM (
            VMType type = VM0)  [inline]
```

References VM0.

Referenced by Compatible(), GetLength(), and operator<<.

12.375.4 Member Function Documentation

12.375.4.1 Compatible()

```
bool gdcm::VM::Compatible (
            VM const & vm) const
```

WARNING: Implementation deficiency The Compatible function is poorly implemented, the reference vm should be coming from the dictionary, while the passed in value is the value guess from the file.

References VM().

12.375.4.2 GetIndex()

```
unsigned int gdcm::VM::GetIndex (
            VMType vm)  [static], [protected]
```

12.375.4.3 GetLength()

```
unsigned int gdcm::VM::GetLength () const
```

References VM(), and operator<<.

12.375.4.4 GetNumberOfElementsFromArray()

```
size_t gdcm::VM::GetNumberOfElementsFromArray (
            const char * array,
            size_t length)  [static]
```

12.375.4.5 GetVMString()

```
const char * gdcm::VM::GetVMString (
            VMType vm)  [static]
```

Return the string as written in the official DICOM dict from a custom enum type

Referenced by operator<<.

12.375.4.6 GetVMType()

```
VMType gdcm::VM::GetVMType (
            const char * vm)  [static]
```

12.375.4.7 GetVMTypeFromLength()

```
VMType gdcm::VM::GetVMTypeFromLength (
            size_t length,
            unsigned int size)  [static]
```

12.375.4.8 IsValid()

```
bool gdcm::VM::IsValid (
            int vm1,
            VMType vm2)  [static]
```

Check if vm1 is valid compare to vm2, i.e vm1 is element of vm2 vm1 is typically deduce from counting in a ValueField

12.375.4.9 operator VMType()

```
gdcm::VM::operator VMType () const  [inline]
```

12.375.5 Friends And Related Symbol Documentation

12.375.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const VM & vm)  [friend]
```

References VM(), and GetVMString().

Referenced by GetLength().

The documentation for this class was generated from the following file:

- gdcmVM.h

12.376 gdcm::VMToLength< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- gdcmVM.h

12.377 gdcm::VR Class Reference

VR class.

```
#include <gdcmVR.h>
```

Public Types

- enum VRType : long long {
  INVALID = 0 ,
  AE = 1 ,
  AS = 2 ,
  AT = 4 ,
  CS = 8 ,
  DA = 16 ,
  DS = 32 ,
  DT = 64 ,
  FD = 128 ,
  FL = 256 ,
  IS = 512 ,
  LO = 1024 ,
  LT = 2048 ,
  OB = 4096 ,
  OD = 134217728 ,

OF = 8192 ,
OL = 268435456 ,
OV = 2147483648 ,
OW = 16384 ,
PN = 32768 ,
SH = 65536 ,
SL = 131072 ,
SQ = 262144 ,
SS = 524288 ,
ST = 1048576 ,
SV = 4294967296 ,
TM = 2097152 ,
UC = 536870912 ,
UI = 4194304 ,
UL = 8388608 ,
UN = 16777216 ,
UR = 1073741824 ,
US = 33554432 ,
UT = 67108864 ,
UV = 8589934592 ,
OB_OW = OB | OW ,
US_SS = US | SS ,
US_SS_OW = US | SS | OW ,
US_OW = US | OW ,
VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM | UI | UL | US ,
VL32 = OB | OW | OD | OF | OL | OV | SQ | SV | UC | UN | UR | UT | UV ,
VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UC | UI | UR | UT ,
VRBINARY = AT | FL | FD | OB | OD | OF | OL | OV | OW | SL | SQ | SS | SV | UL | UN | US | UV ,
VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OV | OD | OW | OB | UN ,
VRALL = VRASCII | VRBINARY ,
VR_END = UV+1 }

Public Member Functions

- VR (VRType vr=INVALID)
- bool Compatible (VR const &vr) const
- int GetLength () const
- unsigned int GetSize () const
- unsigned int GetSizeof () const
- bool IsDual () const
- bool IsVRFile () const
- operator VRType () const
- std::istream & Read (std::istream &is)
- const std::ostream & Write (std::ostream &os) const

Static Public Member Functions

- static bool CanDisplay (VRType vr)
- static uint32_t GetLength (VRType vr)
- static const char * GetVRString (VRType vr)
- static const char * GetVRStringFromFile (VRType vr)

- static VRType GetVRType (const char *vr)
- static VRType GetVRTypeFromFile (const char *vr)
- static bool IsASCII (VRType vr)
- static bool IsASCII2 (VRType vr)
- static bool IsBinary (VRType vr)
- static bool IsBinary2 (VRType vr)
- static bool IsSwap (const char *vr)
- static bool IsValid (const char *vr)
- static bool IsValid (const char *vr1, VRType vr2)

Friends

- std::ostream & operator<< (std::ostream &os, const VR &vr)

12.377.1 Detailed Description

VR class.

This is adapted from DICOM standard The biggest difference is the INVALID VR and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict

Note

VALUE REPRESENTATION (VR) Specifies the data type and format of the Value(s) contained in the Value Field of a Data Element. VALUE REPRESENTATION FIELD: The field where the Value Representation of a Data Element is stored in the encoding of a Data Element structure with explicit VR.

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and SimplePrint.cs.

12.377.2 Member Enumeration Documentation

12.377.2.1 VRType

```
enum gdcm::VR::VRType : long long
```

Enumerator

INVALID	
AE	
AS	
AT	

CS	
DA	
DS	
DT	
FD	
FL	
IS	
LO	
LT	
OB	
OD	
OF	
OL	
OV	
OW	
PN	
SH	
SL	
SQ	
SS	
ST	
SV	
TM	
UC	
UI	
UL	
UN	
UR	
US	
UT	
UV	
OB_OW	
US_SS	
US_SS_OW	
US_OW	
VL16	
VL32	
VRASCII	
VRBINARY	

VR_VM1	
VRALL	
VR_END	

Examples

NewSequence.cs, and SimplePrint.cs.

12.377.3 Constructor & Destructor Documentation

12.377.3.1 VR()

```
gdcm::VR::VR (
            VRType vr = INVALID)  [inline]
```

References INVALID.

Referenced by Compatible(), and operator<<.

12.377.4 Member Function Documentation

12.377.4.1 CanDisplay()

```
bool gdcm::VR::CanDisplay (
            VRType vr)  [static]
```

12.377.4.2 Compatible()

```
bool gdcm::VR::Compatible (
            VR const & vr) const
```

Examples

SimplePrint.cs.

References VR().

12.377.4.3 GetLength() [1/2]

```
int gdcm::VR::GetLength () const  [inline]
```

References GetLength().

Referenced by GetLength().

12.377.4.4 GetLength() [2/2]

```
uint32_t gdcm::VR::GetLength (
            VRType vr)  [inline], [static]
```

References VL32.

12.377.4.5 GetSize()

```
unsigned int gdcm::VR::GetSize () const  [inline]
```

References AE, AS, AT, CS, DA, DS, DT, FD, FL, INVALID, IS, LO, LT, OB, OB_OW, OD, OF, OL, OV, OW, PN, SH, SL, SQ, SS, ST, SV, TM, UC, UI, UL, UN, UR, US, US_OW, US_SS, US_SS_OW, UT, UV, VL16, VL32, VR_END, VR_VM1, VRALL, VRASCII, VRBINARY, and VRTypeTemplateCase.

12.377.4.6 GetSizeof()

```
unsigned int gdcm::VR::GetSizeof () const
```

12.377.4.7 GetVRString()

```
const char * gdcm::VR::GetVRString (
            VRType vr)  [static]
```

Referenced by operator<<, and Write().

12.377.4.8 GetVRStringFromFile()

```
const char * gdcm::VR::GetVRStringFromFile (
            VRType vr)  [static]
```

12.377.4.9 GetVRType()

```
VRType gdcm::VR::GetVRType (
            const char * vr)  [static]
```

12.377.4.10 GetVRTypeFromFile()

```
VRType gdcm::VR::GetVRTypeFromFile (
            const char * vr)  [static]
```

Referenced by Read().

12.377.4.11 IsASCII()

```
bool gdcm::VR::IsASCII (
            VRType vr)  [static]
```

12.377.4.12 IsASCII2()

```
bool gdcm::VR::IsASCII2 (
            VRType vr)  [static]
```

12.377.4.13 IsBinary()

```
bool gdcm::VR::IsBinary (
            VRType vr)  [static]
```

12.377.4.14 IsBinary2()

```
bool gdcm::VR::IsBinary2 (
            VRType vr)  [static]
```

12.377.4.15 IsDual()

```
bool gdcm::VR::IsDual () const
```

Referenced by Write().

12.377.4.16 IsSwap()

```
bool gdcm::VR::IsSwap (
            const char * vr)  [static]
```

12.377.4.17 IsValid() [1/2]

```
bool gdcm::VR::IsValid (
            const char * vr)  [static]
```

12.377.4.18 IsValid() [2/2]

```
bool gdcm::VR::IsValid (
            const char * vr1,
            VRType vr2)  [static]
```

12.377.4.19 IsVRFile()

```
bool gdcm::VR::IsVRFile () const
```

Referenced by gdcm::DataElement::SetVR().

12.377.4.20 operator VRType()

```
gdcm::VR::operator VRType () const  [inline]
```

12.377.4.21 Read()

```
std::istream & gdcm::VR::Read (
            std::istream & is)  [inline]
```

References gdcmDebugMacro, GetVRTypeFromFile(), INVALID, VL32, and VR_END.

12.377.4.22 Write()

```
const std::ostream & gdcm::VR::Write (
            std::ostream & os) const  [inline]
```

References gdcmAssertAlwaysMacro, GetVRString(), INVALID, IsDual(), and VL32.

12.377.5 Friends And Related Symbol Documentation

12.377.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const VR & vr)  [friend]
```

References VR(), and GetVRString().

The documentation for this class was generated from the following file:

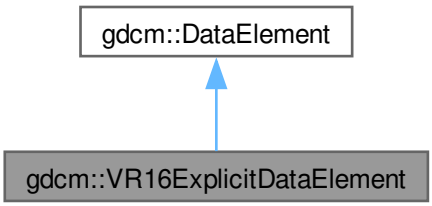
- gdcmVR.h

12.378 gdcm::VR16ExplicitDataElement Class Reference

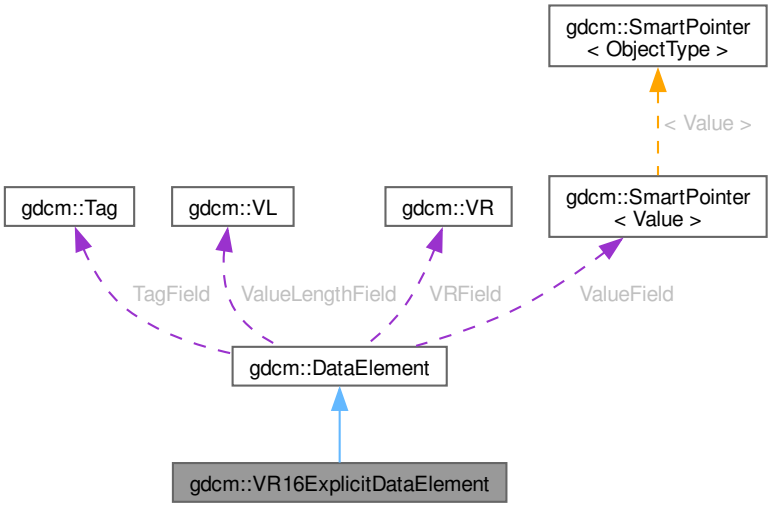
Class to read/write a DataElement as Explicit Data Element.

```
#include <gdcmVR16ExplicitDataElement.h>
```

Inheritance diagram for gdcm::VR16ExplicitDataElement:



Collaboration diagram for gdcm::VR16ExplicitDataElement:



Public Member Functions

- VL GetLength () const

- template<typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TSwap>
  std::istream & ReadPreValue (std::istream &is)
- template<typename TSwap>
  std::istream & ReadValue (std::istream &is, bool readvalues=true)
- template<typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)

Public Member Functions inherited from gdcm::DataElement

- DataElement (const DataElement &_val)
- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- void Clear ()

  Clear Data Element (make Value empty and invalidate Tag & VR).
- void Empty ()

  Make Data Element empty (no Value).
- const ByteValue * GetByteValue () const
- template<typename TDE>
  VL GetLength () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const SequenceOfFragments * GetSequenceOfFragments () const
- Tag & GetTag ()
- const Tag & GetTag () const

  Get Tag.
- Value & GetValue ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- VL & GetVL ()
- const VL & GetVL () const

  Get VL.
- VR const & GetVR () const
- bool IsEmpty () const

  Check if Data Element is empty.
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- bool operator< (const DataElement &de) const
- DataElement & operator= (const DataElement &)=default
- bool operator== (const DataElement &de) const
- template<typename TDE, typename TSwap>
  std::istream & Read (std::istream &is)
- template<typename TDE, typename TSwap>
  std::istream & ReadOrSkip (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadPreValue (std::istream &is, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadValue (std::istream &is, std::set< Tag > const &skiptags)

- template<typename TDE, typename TSwap>
  std::istream & ReadValueWithLength (std::istream &is, VL &length, std::set< Tag > const &skiptags)
- template<typename TDE, typename TSwap>
  std::istream & ReadWithLength (std::istream &is, VL &length)
- void SetByteValue (const char *array, VL length)
- void SetTag (const Tag &t)
- void SetValue (Value const &vl)
- void SetVL (const VL &vl)
- void SetVLToUndefined ()
- void SetVR (VR const &vr)
- template<typename TDE, typename TSwap>
  const std::ostream & Write (std::ostream &os) const

Additional Inherited Members

Protected Types inherited from gdcm::DataElement

- typedef SmartPointer< Value > ValuePtr

Protected Member Functions inherited from gdcm::DataElement

- void SetValueFieldLength (VL vl, bool readvalues)

Protected Attributes inherited from gdcm::DataElement

- Tag TagField
- ValuePtr ValueField
- VL ValueLengthField
- VR VRField

12.378.1 Detailed Description

Class to read/write a DataElement as Explicit Data Element.

Note

This class support 16 bits when finding an unknown VR: For instance: Siemens_CT_Sensation64_has_VR_RT.↩ dcm

12.378.2 Member Function Documentation

12.378.2.1 GetLength()

```
VL gdcm::VR16ExplicitDataElement::GetLength () const
```

12.378.2.2 Read()

```
template<typename TSwap>
std::istream & gdcm::VR16ExplicitDataElement::Read (
            std::istream & is)
```

12.378.2.3 ReadPreValue()

```
template<typename TSwap>
std::istream & gdcm::VR16ExplicitDataElement::ReadPreValue (
            std::istream & is)
```

12.378.2.4 ReadValue()

```
template<typename TSwap>
std::istream & gdcm::VR16ExplicitDataElement::ReadValue (
            std::istream & is,
            bool readvalues = true)
```

12.378.2.5 ReadWithLength()

```
template<typename TSwap>
std::istream & gdcm::VR16ExplicitDataElement::ReadWithLength (
            std::istream & is,
            VL & length)
```

The documentation for this class was generated from the following file:

- gdcmVR16ExplicitDataElement.h

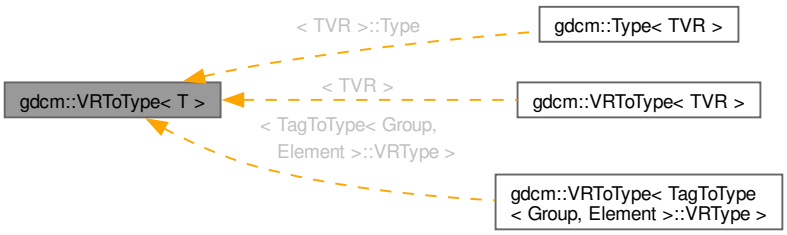
12.379 gdcm::VRToEncoding< T > Struct Template Reference

The documentation for this struct was generated from the following file:

- gdcmVR.h

12.380 gdcm::VRToType< T > Struct Template Reference

Inheritance diagram for gdcm::VRToType< T >:



12.380.1 Detailed Description

template<long long T>
struct gdcm::VRToType< T >

Examples

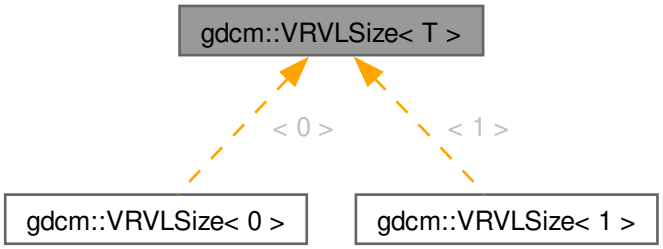
DumpGEMSMovieGroup.cxx.

The documentation for this struct was generated from the following file:

- gdcmVR.h

12.381 gdcm::VRVLSize< T > Class Template Reference

Inheritance diagram for gdcm::VRVLSize< T >:



The documentation for this class was generated from the following file:

- gdcmAttribute.h

12.382 gdcm::VRVLSize< 0 > Class Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::VRVLSize< 0 >:

Collaboration diagram for gdcm::VRVLSize< 0 >:

Static Public Member Functions

- static uint16_t Read (std::istream &_is)
- static void Write (std::ostream &os)

12.382.1 Member Function Documentation

12.382.1.1 Read()

```
uint16_t gdcm::VRVLSize< 0 >::Read (
            std::istream & _is)  [inline], [static]
```

12.382.1.2 Write()

```
void gdcm::VRVLSize< 0 >::Write (
            std::ostream & os)  [inline], [static]
```

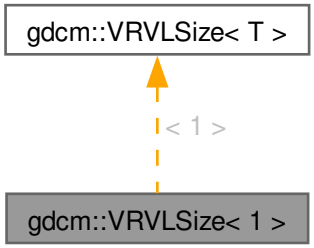
The documentation for this class was generated from the following file:

- gdcmAttribute.h

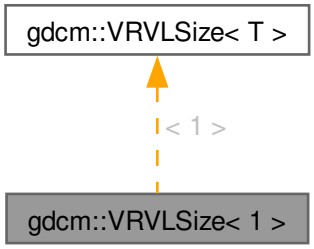
12.383 gdcm::VRVLSize< 1 > Class Reference

```
#include <gdcmAttribute.h>
```

Inheritance diagram for gdcm::VRVLSize< 1 >:



Collaboration diagram for gdcm::VRVLSize< 1 >:



Static Public Member Functions

- static uint32_t Read (std::istream &_is)
- static void Write (std::ostream &os)

12.383.1 Member Function Documentation

12.383.1.1 Read()

```
uint32_t gdcm::VRVLSize< 1 >::Read (
            std::istream & _is)  [inline], [static]
```

12.383.1.2 Write()

```
void gdcm::VRVLSize< 1 >::Write (
            std::ostream & os)  [inline], [static]
```

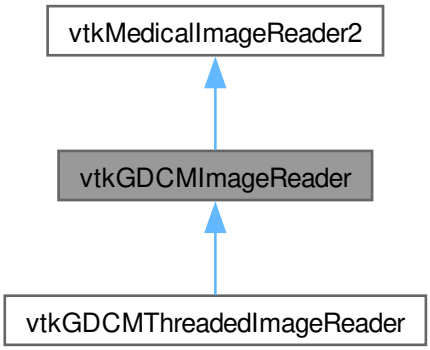
The documentation for this class was generated from the following file:

- gdcmAttribute.h

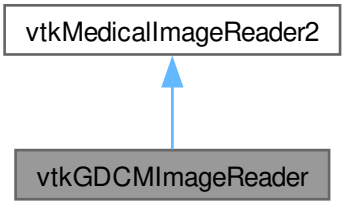
12.384 vtkGDCMImageReader Class Reference

```
#include <vtkGDCMImageReader.h>
```

Inheritance diagram for vtkGDCMImageReader:



Collaboration diagram for vtkGDCMImageReader:



Public Member Functions

- virtual int CanReadFile (const char ∗fname)
- virtual const char ∗ GetDescriptiveName ()
- virtual const char ∗ GetFileExtensions ()
- vtkImageData ∗ GetIconImage ()
- vtkImageData ∗ GetOverlay (int i)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetCurve (vtkPolyData ∗pd)
- virtual void SetFileNames (vtkStringArray ∗)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗pd)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (ApplyLookupTable, int)

- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LossyFlag, int)
- vtkTypeMacro (vtkGDCMImageReader, vtkMedicalImageReader2)

Static Public Member Functions

- static vtkGDCMImageReader ∗ New ()

Protected Member Functions

- vtkGDCMImageReader ()
- ∼vtkGDCMImageReader ()
- void ExecuteData (vtkDataObject ∗out)
- void ExecuteInformation ()
- void FillMedicalImageInformation (const gdcm::ImageReader &reader)
- int LoadSingleFile (const char ∗filename, char ∗pointer, unsigned long &outlen)
- int RequestDataCompat ()
- int RequestInformationCompat ()
- void SetFilePattern (const char ∗)
- void SetFilePrefix (const char ∗)
- vtkGetStringMacro (FilePattern)
- vtkGetStringMacro (FilePrefix)
- vtkSetVector6Macro (ImageOrientationPatient, double)

Protected Attributes

- int ApplyInverseVideo
- int ApplyLookupTable
- int ApplyPlanarConfiguration
- int ApplyShiftScale
- int ApplyYBRToRGB
- vtkPolyData ∗ Curve
- vtkMatrix4x4 ∗ DirectionCosines
- vtkStringArray ∗ FileNames
- int ForceRescale
- int IconDataScalarType
- int IconImageDataExtent [6]
- int IconNumberOfScalarComponents
- int ImageFormat
- double ImageOrientationPatient [6]
- double ImagePositionPatient [3]
- int LoadIconImage
- int LoadOverlays
- int LossyFlag
- vtkMedicalImageProperties ∗ MedicalImageProperties
- int NumberOfIconImages
- int NumberOfOverlays
- int PlanarConfiguration
- double Scale
- double Shift

12.384.1 Detailed Description

Examples

AWTMedical3.java, Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MIPViewer.java, MPRViewer.java, MPRViewer2.java, MagnifyFile.cxx, MetaImageMD5Activiz.cs, ReadSeriesIntoVTK.java, RefCounting.cs, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, and offscreenimage.cxx.

12.384.2 Constructor & Destructor Documentation

12.384.2.1 vtkGDCMImageReader()

```
vtkGDCMImageReader::vtkGDCMImageReader () [protected]
```

Examples

HelloActiviz2.cs.

References vtkGDCMImageReader().

Referenced by vtkGDCMImageReader(), ~vtkGDCMImageReader(), New(), vtkGetStringMacro(), vtkTypeMacro(), and vtkGDCMThreadedImageReader::vtkTypeMacro().

12.384.2.2 ~vtkGDCMImageReader()

```
vtkGDCMImageReader::~vtkGDCMImageReader () [protected]
```

References vtkGDCMImageReader().

12.384.3 Member Function Documentation

12.384.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader::CanReadFile (
            const char * fname) [virtual]
```

Examples

AWTMedical3.java, and MetaImageMD5Activiz.cs.

12.384.3.2 ExecuteData()

```
void vtkGDCMImageReader::ExecuteData (
            vtkDataObject * out)  [protected]
```

12.384.3.3 ExecuteInformation()

```
void vtkGDCMImageReader::ExecuteInformation ()  [protected]
```

12.384.3.4 FillMedicalImageInformation()

```
void vtkGDCMImageReader::FillMedicalImageInformation (
            const gdcm::ImageReader & reader)  [protected]
```

References FillMedicalImageInformation().

Referenced by FillMedicalImageInformation().

12.384.3.5 GetDescriptiveName()

```
virtual const char * vtkGDCMImageReader::GetDescriptiveName ()  [inline], [virtual]
```

12.384.3.6 GetFileExtensions()

```
virtual const char * vtkGDCMImageReader::GetFileExtensions ()  [inline], [virtual]
```

12.384.3.7 GetIconImage()

```
vtkImageData * vtkGDCMImageReader::GetIconImage ()
```

12.384.3.8 GetOverlay()

```
vtkImageData * vtkGDCMImageReader::GetOverlay (
            int i)
```

12.384.3.9 LoadSingleFile()

```
int vtkGDCMImageReader::LoadSingleFile (
            const char * filename,
            char * pointer,
            unsigned long & outlen)  [protected]
```

12.384.3.10 New()

```
vtkGDCMImageReader * vtkGDCMImageReader::New () [static]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, HelloActiviz.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, MagnifyFile.cxx, MetaImageMD5Activiz.cs, RefCounting.cs, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, offscreenimage.cxx, and reslicesphere.cxx.

References vtkGDCMImageReader().

12.384.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader::PrintSelf (
            ostream & os,
            vtkIndent indent) [virtual]
```

Reimplemented in vtkGDCMThreadedImageReader.

12.384.3.12 RequestDataCompat()

```
int vtkGDCMImageReader::RequestDataCompat () [protected]
```

References RequestDataCompat().

Referenced by RequestDataCompat().

12.384.3.13 RequestInformationCompat()

```
int vtkGDCMImageReader::RequestInformationCompat () [protected]
```

References RequestInformationCompat().

Referenced by RequestInformationCompat().

12.384.3.14 SetCurve()

```
virtual void vtkGDCMImageReader::SetCurve (
            vtkPolyData * pd) [virtual]
```

References SetCurve().

Referenced by SetCurve().

12.384.3.15 SetFileNames()

```
virtual void vtkGDCMImageReader::SetFileNames (
            vtkStringArray * )  [virtual]
```

Examples

AWTMedical3.java, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, ReadSeriesIntoVTK.java, and gdcmorthoplanes.cxx.

12.384.3.16 SetFilePattern()

```
void vtkGDCMImageReader::SetFilePattern (
            const char * )  [inline], [protected]
```

12.384.3.17 SetFilePrefix()

```
void vtkGDCMImageReader::SetFilePrefix (
            const char * )  [inline], [protected]
```

12.384.3.18 SetMedicalImageProperties()

```
virtual void vtkGDCMImageReader::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd)  [virtual]
```

12.384.3.19 vtkBooleanMacro() [1/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            ApplyLookupTable ,
            int )
```

References ApplyLookupTable.

12.384.3.20 vtkBooleanMacro() [2/5]

```
int vtkGDCMImageReader::vtkBooleanMacro (
            ApplyYBRToRGB ,
            int )
```

References ApplyYBRToRGB, and vtkBooleanMacro().

12.384.3.21 vtkBooleanMacro() [3/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LoadIconImage ,
            int )
```

References LoadIconImage.

12.384.3.22 vtkBooleanMacro() [4/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LoadOverlays ,
            int )
```

References LoadOverlays.

Referenced by vtkBooleanMacro().

12.384.3.23 vtkBooleanMacro() [5/5]

```
vtkGDCMImageReader::vtkBooleanMacro (
            LossyFlag ,
            int )
```

References LossyFlag.

12.384.3.24 vtkGetMacro() [1/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ApplyLookupTable ,
            int )
```

References ApplyLookupTable.

12.384.3.25 vtkGetMacro() [2/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ApplyYBRToRGB ,
            int )
```

References ApplyYBRToRGB, and vtkSetMacro().

12.384.3.26 vtkGetMacro() [3/11]

```
vtkGDCMImageReader::vtkGetMacro (
            ImageFormat ,
            int )
```

References ImageFormat, and vtkGetMacro().

12.384.3.27 vtkGetMacro() [4/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LoadIconImage ,
            int )
```

References LoadIconImage.

12.384.3.28 vtkGetMacro() [5/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LoadOverlays ,
            int )
```

References LoadOverlays.

Referenced by vtkGetMacro(), vtkGetMacro(), vtkGetMacro(), and vtkGetMacro().

12.384.3.29 vtkGetMacro() [6/11]

```
vtkGDCMImageReader::vtkGetMacro (
            LossyFlag ,
            int )
```

References LossyFlag.

12.384.3.30 vtkGetMacro() [7/11]

```
vtkGDCMImageReader::vtkGetMacro (
            NumberOfIconImages ,
            int )
```

References NumberOfIconImages.

12.384.3.31 vtkGetMacro() [8/11]

```
vtkGDCMImageReader::vtkGetMacro (
            NumberOfOverlays ,
            int )
```

References NumberOfOverlays.

12.384.3.32 vtkGetMacro() [9/11]

```
vtkGDCMImageReader::vtkGetMacro (
            PlanarConfiguration ,
            int )
```

References PlanarConfiguration, and vtkGetMacro().

12.384.3.33 vtkGetMacro() [10/11]

```
vtkGDCMImageReader::vtkGetMacro (
            Scale ,
            double )
```

References Scale, and vtkGetMacro().

12.384.3.34 vtkGetMacro() [11/11]

```
vtkGDCMImageReader::vtkGetMacro (
            Shift ,
            double )
```

References Shift, and vtkGetMacro().

12.384.3.35 vtkGetObjectMacro() [1/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            Curve ,
            vtkPolyData )
```

References Curve, and vtkGetObjectMacro().

12.384.3.36 vtkGetObjectMacro() [2/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4 )
```

References DirectionCosines.

Referenced by vtkGetObjectMacro().

12.384.3.37 vtkGetObjectMacro() [3/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray )
```

References FileNames.

12.384.3.38 vtkGetObjectMacro() [4/4]

```
vtkGDCMImageReader::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties )
```

References MedicalImageProperties.

12.384.3.39 vtkGetStringMacro() [1/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
            FilePattern )  [protected]
```

References vtkGDCMImageReader().

12.384.3.40 vtkGetStringMacro() [2/2]

```
vtkGDCMImageReader::vtkGetStringMacro (
            FilePrefix )  [protected]
```

12.384.3.41 vtkGetVector3Macro()

```
vtkGDCMImageReader::vtkGetVector3Macro (
            ImagePositionPatient ,
            double )
```

References ImagePositionPatient, and vtkGetVector3Macro().

Referenced by vtkGetVector3Macro().

12.384.3.42 vtkGetVector6Macro()

```
vtkGDCMImageReader::vtkGetVector6Macro (
            ImageOrientationPatient ,
            double )
```

References ImageOrientationPatient, and vtkGetVector6Macro().

Referenced by vtkGetVector6Macro().

12.384.3.43 vtkSetMacro() [1/4]

```
vtkGDCMImageReader::vtkSetMacro (
            ApplyLookupTable ,
            int )
```

References ApplyLookupTable.

12.384.3.44 vtkSetMacro() [2/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LoadIconImage ,
            int )
```

References LoadIconImage.

12.384.3.45 vtkSetMacro() [3/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LoadOverlays ,
            int )
```

References LoadOverlays.

Referenced by vtkGetMacro().

12.384.3.46 vtkSetMacro() [4/4]

```
vtkGDCMImageReader::vtkSetMacro (
            LossyFlag ,
            int )
```

References LossyFlag.

12.384.3.47 vtkSetVector6Macro()

```
vtkGDCMImageReader::vtkSetVector6Macro (
            ImageOrientationPatient ,
            double )  [protected]
```

References ImageOrientationPatient, and vtkSetVector6Macro().

Referenced by vtkSetVector6Macro().

12.384.3.48 vtkTypeMacro()

```
vtkGDCMImageReader::vtkTypeMacro (
            vtkGDCMImageReader ,
            vtkMedicalImageReader2 )
```

References vtkGDCMImageReader().

12.384.4 Member Data Documentation

12.384.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader::ApplyInverseVideo  [protected]
```

12.384.4.2 ApplyLookupTable

```
int vtkGDCMImageReader::ApplyLookupTable  [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

12.384.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader::ApplyPlanarConfiguration  [protected]
```

12.384.4.4 ApplyShiftScale

```
int vtkGDCMImageReader::ApplyShiftScale  [protected]
```

12.384.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader::ApplyYBRToRGB  [protected]
```

Referenced by vtkBooleanMacro(), and vtkGetMacro().

12.384.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader::Curve  [protected]
```

Referenced by vtkGetObjectMacro().

12.384.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines [protected]
```

Referenced by vtkGetObjectMacro().

12.384.4.8 FileNames

```
vtkStringArray* vtkGDCMImageReader::FileNames [protected]
```

Referenced by vtkGetObjectMacro().

12.384.4.9 ForceRescale

```
int vtkGDCMImageReader::ForceRescale [protected]
```

12.384.4.10 IconDataScalarType

```
int vtkGDCMImageReader::IconDataScalarType [protected]
```

12.384.4.11 IconImageDataExtent

```
int vtkGDCMImageReader::IconImageDataExtent[6] [protected]
```

12.384.4.12 IconNumberOfScalarComponents

```
int vtkGDCMImageReader::IconNumberOfScalarComponents [protected]
```

12.384.4.13 ImageFormat

```
int vtkGDCMImageReader::ImageFormat [protected]
```

Referenced by vtkGetMacro().

12.384.4.14 ImageOrientationPatient

```
double vtkGDCMImageReader::ImageOrientationPatient[6] [protected]
```

Referenced by vtkGetVector6Macro(), and vtkSetVector6Macro().

12.384.4.15 ImagePositionPatient

```
double vtkGDCMImageReader::ImagePositionPatient[3]  [protected]
```

Referenced by vtkGetVector3Macro().

12.384.4.16 LoadIconImage

```
int vtkGDCMImageReader::LoadIconImage  [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

12.384.4.17 LoadOverlays

```
int vtkGDCMImageReader::LoadOverlays  [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

12.384.4.18 LossyFlag

```
int vtkGDCMImageReader::LossyFlag  [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

12.384.4.19 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties  [protected]
```

Referenced by vtkGetObjectMacro().

12.384.4.20 NumberOfIconImages

```
int vtkGDCMImageReader::NumberOfIconImages  [protected]
```

Referenced by vtkGetMacro().

12.384.4.21 NumberOfOverlays

```
int vtkGDCMImageReader::NumberOfOverlays  [protected]
```

Referenced by vtkGetMacro().

12.384.4.22 PlanarConfiguration

```
int vtkGDCMImageReader::PlanarConfiguration [protected]
```

Referenced by vtkGetMacro().

12.384.4.23 Scale

```
double vtkGDCMImageReader::Scale [protected]
```

Referenced by vtkGetMacro(), and vtkGDCMThreadedImageReader::vtkSetMacro().

12.384.4.24 Shift

```
double vtkGDCMImageReader::Shift [protected]
```

Referenced by vtkGetMacro(), and vtkGDCMThreadedImageReader::vtkSetMacro().

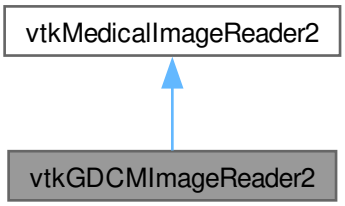
The documentation for this class was generated from the following file:

- vtkGDCMImageReader.h

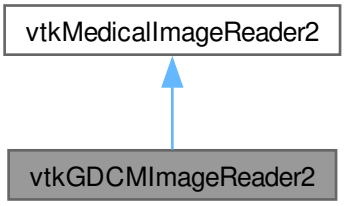
12.385 vtkGDCMImageReader2 Class Reference

```
#include <vtkGDCMImageReader2.h>
```

Inheritance diagram for vtkGDCMImageReader2:



Collaboration diagram for vtkGDCMImageReader2:



Public Member Functions

- virtual int CanReadFile (const char ∗fname)
- virtual const char ∗ GetDescriptiveName ()
- virtual const char ∗ GetFileExtensions ()
- vtkImageData ∗ GetIconImage ()
- vtkAlgorithmOutput ∗ GetIconImagePort ()
- vtkImageData ∗ GetOverlay (int i)
- vtkAlgorithmOutput ∗ GetOverlayPort (int index)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetCurve (vtkPolyData ∗pd)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗pd)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (ApplyLookupTable, int)
- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LossyFlag, int)
- vtkTypeMacro (vtkGDCMImageReader2, vtkMedicalImageReader2)

Static Public Member Functions

- static vtkGDCMImageReader2 ∗ New ()

Protected Member Functions

- vtkGDCMImageReader2 ()
- ∼vtkGDCMImageReader2 ()
- void FillMedicalImageInformation (const gdcm::ImageReader &reader)
- int LoadSingleFile (const char ∗filename, char ∗pointer, unsigned long &outlen)
- int ProcessRequest (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗output↩ Vector)
- int RequestData (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗output↩ Vector)
- int RequestDataCompat ()
- int RequestInformation (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector)
- int RequestInformationCompat ()
- void SetFilePattern (const char ∗)
- void SetFilePrefix (const char ∗)
- vtkGetStringMacro (FilePattern)
- vtkGetStringMacro (FilePrefix)
- vtkSetVector6Macro (ImageOrientationPatient, double)

Protected Attributes

- int ApplyInverseVideo
- int ApplyLookupTable
- int ApplyPlanarConfiguration
- int ApplyShiftScale
- int ApplyYBRToRGB
- vtkPolyData ∗ Curve
- vtkMatrix4x4 ∗ DirectionCosines
- int ForceRescale
- int IconDataScalarType
- int IconImageDataExtent [6]
- int IconNumberOfScalarComponents
- int ImageFormat
- double ImageOrientationPatient [6]
- double ImagePositionPatient [3]
- int LoadIconImage
- int LoadOverlays
- int LossyFlag
- int NumberOfIconImages
- int NumberOfOverlays
- int PlanarConfiguration
- double Scale
- double Shift

12.385.1 Detailed Description

Examples

Compute3DSpacing.cxx.

12.385.2 Constructor & Destructor Documentation

12.385.2.1 vtkGDCMImageReader2()

```
vtkGDCMImageReader2::vtkGDCMImageReader2 ()  [protected]
```

References vtkGDCMImageReader2().

Referenced by vtkGDCMImageReader2(), ~vtkGDCMImageReader2(), New(), vtkGetStringMacro(), and vtkTypeMacro().

12.385.2.2 ~vtkGDCMImageReader2()

```
vtkGDCMImageReader2::~vtkGDCMImageReader2 ()  [protected]
```

References vtkGDCMImageReader2().

12.385.3 Member Function Documentation

12.385.3.1 CanReadFile()

```
virtual int vtkGDCMImageReader2::CanReadFile (
            const char * fname)  [virtual]
```

12.385.3.2 FillMedicalImageInformation()

```
void vtkGDCMImageReader2::FillMedicalImageInformation (
            const gdcm::ImageReader & reader)  [protected]
```

References FillMedicalImageInformation().

Referenced by FillMedicalImageInformation().

12.385.3.3 GetDescriptiveName()

```
virtual const char * vtkGDCMImageReader2::GetDescriptiveName ()  [inline], [virtual]
```

12.385.3.4 GetFileExtensions()

```
virtual const char * vtkGDCMImageReader2::GetFileExtensions ()  [inline], [virtual]
```

12.385.3.5 GetIconImage()

```
vtkImageData * vtkGDCMImageReader2::GetIconImage ()
```

12.385.3.6 GetIconImagePort()

```
vtkAlgorithmOutput * vtkGDCMImageReader2::GetIconImagePort ()
```

12.385.3.7 GetOverlay()

```
vtkImageData * vtkGDCMImageReader2::GetOverlay (
            int i)
```

12.385.3.8 GetOverlayPort()

```
vtkAlgorithmOutput * vtkGDCMImageReader2::GetOverlayPort (
            int index)
```

12.385.3.9 LoadSingleFile()

```
int vtkGDCMImageReader2::LoadSingleFile (
            const char * filename,
            char * pointer,
            unsigned long & outlen)  [protected]
```

References LoadSingleFile().

Referenced by LoadSingleFile().

12.385.3.10 New()

```
vtkGDCMImageReader2 * vtkGDCMImageReader2::New ()  [static]
```

Examples

Compute3DSpacing.cxx.

References vtkGDCMImageReader2().

12.385.3.11 PrintSelf()

```
virtual void vtkGDCMImageReader2::PrintSelf (
            ostream & os,
            vtkIndent indent)  [virtual]
```

12.385.3.12 ProcessRequest()

```
int vtkGDCMImageReader2::ProcessRequest (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector)  [protected]
```

References ProcessRequest().

Referenced by ProcessRequest().

12.385.3.13 RequestData()

```
int vtkGDCMImageReader2::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector)  [protected]
```

References RequestData().

Referenced by RequestData().

12.385.3.14 RequestDataCompat()

```
int vtkGDCMImageReader2::RequestDataCompat ()  [protected]
```

References RequestDataCompat().

Referenced by RequestDataCompat().

12.385.3.15 RequestInformation()

```
int vtkGDCMImageReader2::RequestInformation (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector)  [protected]
```

References RequestInformation().

Referenced by RequestInformation().

12.385.3.16 RequestInformationCompat()

```
int vtkGDCMImageReader2::RequestInformationCompat ()  [protected]
```

References RequestInformationCompat().

Referenced by RequestInformationCompat().

12.385.3.17 SetCurve()

```
virtual void vtkGDCMImageReader2::SetCurve (
            vtkPolyData * pd)  [virtual]
```

References SetCurve().

Referenced by SetCurve().

12.385.3.18 SetFilePattern()

```
void vtkGDCMImageReader2::SetFilePattern (
            const char * )  [inline], [protected]
```

12.385.3.19 SetFilePrefix()

```
void vtkGDCMImageReader2::SetFilePrefix (
            const char * )  [inline], [protected]
```

References SetFilePrefix().

Referenced by SetFilePrefix().

12.385.3.20 SetMedicalImageProperties()

```
virtual void vtkGDCMImageReader2::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd)  [virtual]
```

12.385.3.21 vtkBooleanMacro() [1/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            ApplyLookupTable ,
            int )
```

References ApplyLookupTable.

12.385.3.22 vtkBooleanMacro() [2/5]

```
int vtkGDCMImageReader2::vtkBooleanMacro (
            ApplyYBRToRGB ,
            int )
```

References ApplyYBRToRGB, and vtkBooleanMacro().

12.385.3.23 vtkBooleanMacro() [3/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LoadIconImage ,
            int )
```

References LoadIconImage.

12.385.3.24 vtkBooleanMacro() [4/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LoadOverlays ,
            int )
```

References LoadOverlays.

Referenced by vtkBooleanMacro().

12.385.3.25 vtkBooleanMacro() [5/5]

```
vtkGDCMImageReader2::vtkBooleanMacro (
            LossyFlag ,
            int )
```

References LossyFlag.

12.385.3.26 vtkGetMacro() [1/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ApplyLookupTable ,
            int )
```

References ApplyLookupTable.

12.385.3.27 vtkGetMacro() [2/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ApplyYBRToRGB ,
            int )
```

References ApplyYBRToRGB, and vtkSetMacro().

12.385.3.28 vtkGetMacro() [3/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            ImageFormat ,
            int )
```

References ImageFormat, and vtkGetMacro().

12.385.3.29 vtkGetMacro() [4/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LoadIconImage ,
            int )
```

References LoadIconImage.

12.385.3.30 vtkGetMacro() [5/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LoadOverlays ,
            int )
```

References LoadOverlays.

Referenced by vtkGetMacro(), vtkGetMacro(), vtkGetMacro(), and vtkGetMacro().

12.385.3.31 vtkGetMacro() [6/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            LossyFlag ,
            int )
```

References LossyFlag.

12.385.3.32 vtkGetMacro() [7/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            NumberOfIconImages ,
            int )
```

References NumberOfIconImages.

12.385.3.33 vtkGetMacro() [8/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            NumberOfOverlays ,
            int )
```

References NumberOfOverlays.

12.385.3.34 vtkGetMacro() [9/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            PlanarConfiguration ,
            int )
```

References PlanarConfiguration, and vtkGetMacro().

12.385.3.35 vtkGetMacro() [10/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            Scale ,
            double )
```

References Scale, and vtkGetMacro().

12.385.3.36 vtkGetMacro() [11/11]

```
vtkGDCMImageReader2::vtkGetMacro (
            Shift ,
            double )
```

References Shift, and vtkGetMacro().

12.385.3.37 vtkGetObjectMacro() [1/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
            Curve ,
            vtkPolyData )
```

References Curve, and vtkGetObjectMacro().

12.385.3.38 vtkGetObjectMacro() [2/2]

```
vtkGDCMImageReader2::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4 )
```

References DirectionCosines.

Referenced by vtkGetObjectMacro().

12.385.3.39 vtkGetStringMacro() [1/2]

```
vtkGDCMImageReader2::vtkGetStringMacro (
            FilePattern )  [protected]
```

References vtkGDCMImageReader2().

12.385.3.40 vtkGetStringMacro() [2/2]

```
vtkGDCMImageReader2::vtkGetStringMacro (
            FilePrefix )  [protected]
```

12.385.3.41 vtkGetVector3Macro()

```
vtkGDCMImageReader2::vtkGetVector3Macro (
            ImagePositionPatient ,
            double )
```

References ImagePositionPatient, and vtkGetVector3Macro().

Referenced by vtkGetVector3Macro().

12.385.3.42 vtkGetVector6Macro()

```
vtkGDCMImageReader2::vtkGetVector6Macro (
            ImageOrientationPatient ,
            double )
```

References ImageOrientationPatient, and vtkGetVector6Macro().

Referenced by vtkGetVector6Macro().

12.385.3.43 vtkSetMacro() [1/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            ApplyLookupTable ,
            int )
```

References ApplyLookupTable.

12.385.3.44 vtkSetMacro() [2/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LoadIconImage ,
            int )
```

References LoadIconImage.

12.385.3.45 vtkSetMacro() [3/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LoadOverlays ,
            int )
```

References LoadOverlays.

Referenced by vtkGetMacro().

12.385.3.46 vtkSetMacro() [4/4]

```
vtkGDCMImageReader2::vtkSetMacro (
            LossyFlag ,
            int )
```

References LossyFlag.

12.385.3.47 vtkSetVector6Macro()

```
vtkGDCMImageReader2::vtkSetVector6Macro (
            ImageOrientationPatient ,
            double )  [protected]
```

References ImageOrientationPatient, and vtkSetVector6Macro().

Referenced by vtkSetVector6Macro().

12.385.3.48 vtkTypeMacro()

```
vtkGDCMImageReader2::vtkTypeMacro (
            vtkGDCMImageReader2 ,
            vtkMedicalImageReader2 )
```

References vtkGDCMImageReader2().

12.385.4 Member Data Documentation

12.385.4.1 ApplyInverseVideo

```
int vtkGDCMImageReader2::ApplyInverseVideo  [protected]
```

12.385.4.2 ApplyLookupTable

```
int vtkGDCMImageReader2::ApplyLookupTable  [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

12.385.4.3 ApplyPlanarConfiguration

```
int vtkGDCMImageReader2::ApplyPlanarConfiguration  [protected]
```

12.385.4.4 ApplyShiftScale

```
int vtkGDCMImageReader2::ApplyShiftScale  [protected]
```

12.385.4.5 ApplyYBRToRGB

```
int vtkGDCMImageReader2::ApplyYBRToRGB  [protected]
```

Referenced by vtkBooleanMacro(), and vtkGetMacro().

12.385.4.6 Curve

```
vtkPolyData* vtkGDCMImageReader2::Curve  [protected]
```

Referenced by vtkGetObjectMacro().

12.385.4.7 DirectionCosines

```
vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines  [protected]
```

Referenced by vtkGetObjectMacro().

12.385.4.8 ForceRescale

```
int vtkGDCMImageReader2::ForceRescale  [protected]
```

12.385.4.9 IconDataScalarType

```
int vtkGDCMImageReader2::IconDataScalarType  [protected]
```

12.385.4.10 IconImageDataExtent

```
int vtkGDCMImageReader2::IconImageDataExtent[6]  [protected]
```

12.385.4.11 IconNumberOfScalarComponents

```
int vtkGDCMImageReader2::IconNumberOfScalarComponents  [protected]
```

12.385.4.12 ImageFormat

```
int vtkGDCMImageReader2::ImageFormat  [protected]
```

Referenced by vtkGetMacro().

12.385.4.13 ImageOrientationPatient

```
double vtkGDCMImageReader2::ImageOrientationPatient[6]  [protected]
```

Referenced by vtkGetVector6Macro(), and vtkSetVector6Macro().

12.385.4.14 ImagePositionPatient

```
double vtkGDCMImageReader2::ImagePositionPatient[3]  [protected]
```

Referenced by vtkGetVector3Macro().

12.385.4.15 LoadIconImage

```
int vtkGDCMImageReader2::LoadIconImage [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

12.385.4.16 LoadOverlays

```
int vtkGDCMImageReader2::LoadOverlays [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

12.385.4.17 LossyFlag

```
int vtkGDCMImageReader2::LossyFlag [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

12.385.4.18 NumberOfIconImages

```
int vtkGDCMImageReader2::NumberOfIconImages [protected]
```

Referenced by vtkGetMacro().

12.385.4.19 NumberOfOverlays

```
int vtkGDCMImageReader2::NumberOfOverlays [protected]
```

Referenced by vtkGetMacro().

12.385.4.20 PlanarConfiguration

```
int vtkGDCMImageReader2::PlanarConfiguration [protected]
```

Referenced by vtkGetMacro().

12.385.4.21 Scale

```
double vtkGDCMImageReader2::Scale [protected]
```

Referenced by vtkGetMacro().

12.385.4.22 Shift

```
double vtkGDCMImageReader2::Shift  [protected]
```

Referenced by vtkGetMacro().

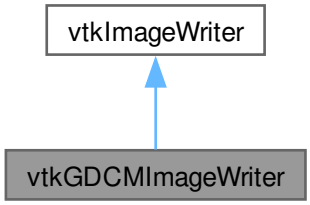
The documentation for this class was generated from the following file:

- vtkGDCMImageReader2.h

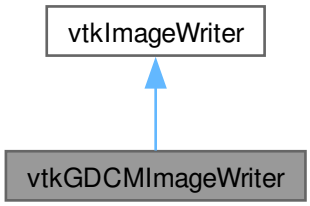
12.386 vtkGDCMImageWriter Class Reference

```
#include <vtkGDCMImageWriter.h>
```

Inheritance diagram for vtkGDCMImageWriter:



Collaboration diagram for vtkGDCMImageWriter:



Public Types

- enum CompressionTypes {
  NO_COMPRESSION = 0 ,
  JPEG_COMPRESSION ,
  JPEG2000_COMPRESSION ,
  JPEGLS_COMPRESSION ,
  RLE_COMPRESSION }

Public Member Functions

- virtual const char ∗ GetDescriptiveName ()
- virtual const char ∗ GetFileExtensions ()
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetDirectionCosines (vtkMatrix4x4 ∗matrix)
- virtual void SetDirectionCosinesFromImageOrientationPatient (const double dircos[6])
- virtual void SetFileNames (vtkStringArray ∗)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗)
- vtkBooleanMacro (FileLowerLeft, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkGetMacro (CompressionType, int)
- vtkGetMacro (FileLowerLeft, int)
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetStringMacro (SeriesUID)
- vtkGetStringMacro (StudyUID)
- vtkSetMacro (CompressionType, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (ImageFormat, int)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (PlanarConfiguration, int)
- vtkSetMacro (Scale, double)
- vtkSetMacro (Shift, double)
- vtkSetStringMacro (SeriesUID)
- vtkSetStringMacro (StudyUID)
- vtkTypeMacro (vtkGDCMImageWriter, vtkImageWriter)
- virtual void Write ()

Static Public Member Functions

- static vtkGDCMImageWriter ∗ New ()

Protected Member Functions

- vtkGDCMImageWriter ()
- ~vtkGDCMImageWriter ()
- virtual char * GetFileName ()
- int WriteGDCMData (vtkImageData *data, int timeStep)
- void WriteSlice (vtkImageData *data)

12.386.1 Detailed Description

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, MagnifyFile.cxx, RefCounting.cs, and gdcmorthoplanes.cxx.

12.386.2 Member Enumeration Documentation

12.386.2.1 CompressionTypes

```
enum vtkGDCMImageWriter::CompressionTypes
```

Enumerator

NO_COMPRESSION	
JPEG_COMPRESSION	
JPEG2000_COMPRESSION	
JPEGLS_COMPRESSION	
RLE_COMPRESSION	

12.386.3 Constructor & Destructor Documentation

12.386.3.1 vtkGDCMImageWriter()

```
vtkGDCMImageWriter::vtkGDCMImageWriter () [protected]
```

Referenced by GetFileName(), New(), and vtkTypeMacro().

12.386.3.2 ~vtkGDCMImageWriter()

```
vtkGDCMImageWriter::~vtkGDCMImageWriter () [protected]
```

12.386.4 Member Function Documentation

12.386.4.1 GetDescriptiveName()

```
virtual const char * vtkGDCMImageWriter::GetDescriptiveName () [inline], [virtual]
```

12.386.4.2 GetFileExtensions()

```
virtual const char * vtkGDCMImageWriter::GetFileExtensions () [inline], [virtual]
```

12.386.4.3 GetFileName()

```
virtual char * vtkGDCMImageWriter::GetFileName () [protected], [virtual]
```

References vtkGDCMImageWriter().

12.386.4.4 New()

```
vtkGDCMImageWriter * vtkGDCMImageWriter::New () [static]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, HelloActiviz.cs, HelloVTKWorld.cs, HelloVTKWorld2.cs, MagnifyFile.cxx, RefCounting.cs, and gdcmorthoplanes.cxx.

References vtkGDCMImageWriter().

12.386.4.5 PrintSelf()

```
virtual void vtkGDCMImageWriter::PrintSelf (
            ostream & os,
            vtkIndent indent) [virtual]
```

12.386.4.6 SetDirectionCosines()

```
virtual void vtkGDCMImageWriter::SetDirectionCosines (
            vtkMatrix4x4 * matrix) [virtual]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MagnifyFile.cxx, and gdcmorthoplanes.cxx.

12.386.4.7 SetDirectionCosinesFromImageOrientationPatient()

```
virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (
            const double dircos[6])  [virtual]
```

12.386.4.8 SetFileNames()

```
virtual void vtkGDCMImageWriter::SetFileNames (
            vtkStringArray * )  [virtual]
```

Examples

ConvertMultiFrameToSingleFrame.cxx, and CreateFakePET.cxx.

12.386.4.9 SetMedicalImageProperties()

```
virtual void vtkGDCMImageWriter::SetMedicalImageProperties (
            vtkMedicalImageProperties * )  [virtual]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MagnifyFile.cxx, and gdcmorthoplanes.cxx.

12.386.4.10 vtkBooleanMacro() [1/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
            FileLowerLeft ,
            int )
```

12.386.4.11 vtkBooleanMacro() [2/2]

```
vtkGDCMImageWriter::vtkBooleanMacro (
            LossyFlag ,
            int )
```

12.386.4.12 vtkGetMacro() [1/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            CompressionType ,
            int )
```

12.386.4.13 vtkGetMacro() [2/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            FileLowerLeft ,
            int )
```

12.386.4.14 vtkGetMacro() [3/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            ImageFormat ,
            int )
```

12.386.4.15 vtkGetMacro() [4/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            LossyFlag ,
            int )
```

12.386.4.16 vtkGetMacro() [5/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            PlanarConfiguration ,
            int )
```

12.386.4.17 vtkGetMacro() [6/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            Scale ,
            double )
```

12.386.4.18 vtkGetMacro() [7/7]

```
vtkGDCMImageWriter::vtkGetMacro (
            Shift ,
            double )
```

12.386.4.19 vtkGetObjectMacro() [1/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            DirectionCosines ,
            vtkMatrix4x4 )
```

12.386.4.20 vtkGetObjectMacro() [2/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray )
```

12.386.4.21 vtkGetObjectMacro() [3/3]

```
vtkGDCMImageWriter::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties )
```

12.386.4.22 vtkGetStringMacro() [1/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
            SeriesUID )
```

12.386.4.23 vtkGetStringMacro() [2/2]

```
vtkGDCMImageWriter::vtkGetStringMacro (
            StudyUID )
```

12.386.4.24 vtkSetMacro() [1/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            CompressionType ,
            int )
```

12.386.4.25 vtkSetMacro() [2/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            FileLowerLeft ,
            int )
```

12.386.4.26 vtkSetMacro() [3/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            ImageFormat ,
            int )
```

12.386.4.27 vtkSetMacro() [4/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            LossyFlag ,
            int )
```

12.386.4.28 vtkSetMacro() [5/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            PlanarConfiguration ,
            int )
```

12.386.4.29 vtkSetMacro() [6/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            Scale ,
            double )
```

12.386.4.30 vtkSetMacro() [7/7]

```
vtkGDCMImageWriter::vtkSetMacro (
            Shift ,
            double )
```

12.386.4.31 vtkSetStringMacro() [1/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
            SeriesUID )
```

12.386.4.32 vtkSetStringMacro() [2/2]

```
vtkGDCMImageWriter::vtkSetStringMacro (
            StudyUID )
```

12.386.4.33 vtkTypeMacro()

```
vtkGDCMImageWriter::vtkTypeMacro (
            vtkGDCMImageWriter ,
            vtkImageWriter )
```

References vtkGDCMImageWriter().

12.386.4.34 Write()

```
virtual void vtkGDCMImageWriter::Write ()  [virtual]
```

Examples

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakePET.cxx, CreateFakeRTDOSE.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, MagnifyFile.cxx, and gdcmorthoplanes.cxx.

12.386.4.35 WriteGDCMData()

```
int vtkGDCMImageWriter::WriteGDCMData (
            vtkImageData * data,
            int timeStep)  [protected]
```

12.386.4.36 WriteSlice()

```
void vtkGDCMImageWriter::WriteSlice (
            vtkImageData * data)  [protected]
```

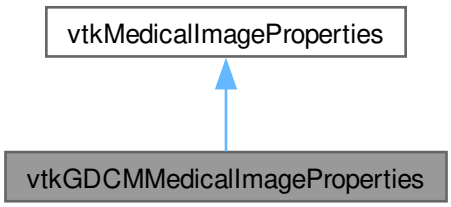
The documentation for this class was generated from the following file:

- vtkGDCMImageWriter.h

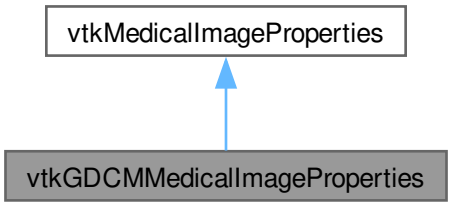
12.387 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

Inheritance diagram for vtkGDCMMedicalImageProperties:



Collaboration diagram for vtkGDCMMedicalImageProperties:



Public Member Functions

- virtual void Clear ()
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkGDCMMedicalImageProperties, vtkMedicalImageProperties)

Static Public Member Functions

- static vtkGDCMMedicalImageProperties * New ()

Protected Member Functions

- vtkGDCMMedicalImageProperties ()
- ~vtkGDCMMedicalImageProperties ()
- gdcm::File const & GetFile (unsigned int t)
- void PushBackFile (gdcm::File const &f)

Friends

- class vtkGDCMImageReader
- class vtkGDCMImageReader2
- class vtkGDCMImageWriter

12.387.1 Constructor & Destructor Documentation

12.387.1.1 vtkGDCMMedicalImageProperties()

```
vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties () [protected]
```

Referenced by GetFile(), New(), and vtkTypeMacro().

12.387.1.2 ~vtkGDCMMedicalImageProperties()

```
vtkGDCMMedicalImageProperties::~vtkGDCMMedicalImageProperties () [protected]
```

12.387.2 Member Function Documentation

12.387.2.1 Clear()

```
virtual void vtkGDCMMedicalImageProperties::Clear () [virtual]
```

12.387.2.2 GetFile()

```
gdcm::File const & vtkGDCMMedicalImageProperties::GetFile (
            unsigned int t) [protected]
```

References vtkGDCMMedicalImageProperties().

12.387.2.3 New()

```
vtkGDCMMedicalImageProperties * vtkGDCMMedicalImageProperties::New () [static]
```

References vtkGDCMMedicalImageProperties().

12.387.2.4 PrintSelf()

```
void vtkGDCMMedicalImageProperties::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.387.2.5 PushBackFile()

```
void vtkGDCMMedicalImageProperties::PushBackFile (
            gdcm::File const & f) [protected]
```

12.387.2.6 vtkTypeMacro()

```
vtkGDCMMedicalImageProperties::vtkTypeMacro (
            vtkGDCMMedicalImageProperties ,
            vtkMedicalImageProperties )
```

References vtkGDCMMedicalImageProperties().

12.387.3 Friends And Related Symbol Documentation

12.387.3.1 vtkGDCMImageReader

```
friend class vtkGDCMImageReader [friend]
```

References vtkGDCMImageReader.

Referenced by vtkGDCMImageReader.

12.387.3.2 vtkGDCMImageReader2

```
friend class vtkGDCMImageReader2 [friend]
```

References vtkGDCMImageReader2.

Referenced by vtkGDCMImageReader2.

12.387.3.3 vtkGDCMImageWriter

```
friend class vtkGDCMImageWriter [friend]
```

References vtkGDCMImageWriter.

Referenced by vtkGDCMImageWriter.

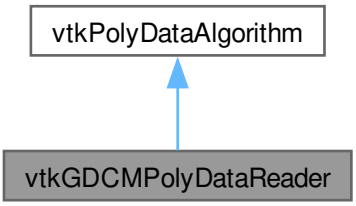
The documentation for this class was generated from the following file:

- vtkGDCMMedicalImageProperties.h

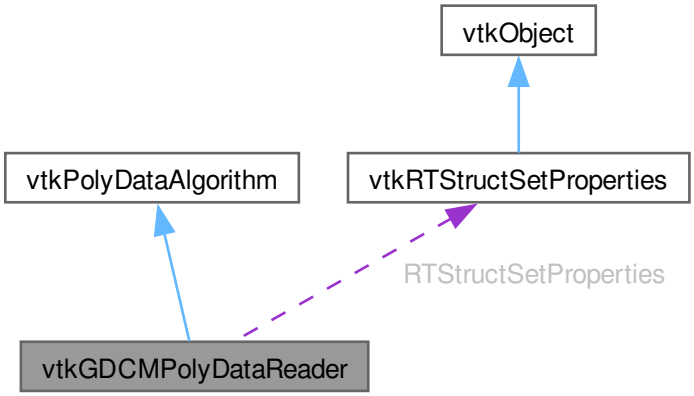
12.388 vtkGDCMPolyDataReader Class Reference

```
#include <vtkGDCMPolyDataReader.h>
```

Inheritance diagram for vtkGDCMPolyDataReader:



Collaboration diagram for vtkGDCMPolyDataReader:



Public Member Functions

- virtual void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (RTStructSetProperties, vtkRTStructSetProperties)
- vtkGetStringMacro (FileName)
- vtkSetStringMacro (FileName)
- vtkTypeMacro (vtkGDCMPolyDataReader, vtkPolyDataAlgorithm)

Static Public Member Functions

- static vtkGDCMPolyDataReader ∗ New ()

Protected Member Functions

- vtkGDCMPolyDataReader ()
- ∼vtkGDCMPolyDataReader ()
- void FillMedicalImageInformation (const gdcm::Reader &reader)
- int RequestData (vtkInformation ∗, vtkInformationVector ∗∗, vtkInformationVector ∗)
- int RequestData_HemodynamicWaveformStorage (gdcm::Reader const &reader, vtkInformationVector ∗output↩ Vector)
- int RequestData_RTStructureSetStorage (gdcm::Reader const &reader, vtkInformationVector ∗outputVector)
- int RequestInformation (vtkInformation ∗vtkNotUsed(request), vtkInformationVector ∗∗vtkNotUsed(inputVector), vtkInformationVector ∗outputVector)
- int RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const &reader)
- int RequestInformation_RTStructureSetStorage (gdcm::Reader const &reader)

Protected Attributes

- char * FileName
- vtkMedicalImageProperties * MedicalImageProperties
- vtkRTStructSetProperties * RTStructSetProperties

12.388.1 Detailed Description

Examples

GenerateRTSTRUCT.cxx, gdcmscene.cxx, and rtstructapp.cxx.

12.388.2 Constructor & Destructor Documentation

12.388.2.1 vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::vtkGDCMPolyDataReader () [protected]
```

Referenced by New(), RequestData_HemodynamicWaveformStorage(), and vtkTypeMacro().

12.388.2.2 ~vtkGDCMPolyDataReader()

```
vtkGDCMPolyDataReader::~vtkGDCMPolyDataReader () [protected]
```

12.388.3 Member Function Documentation

12.388.3.1 FillMedicalImageInformation()

```
void vtkGDCMPolyDataReader::FillMedicalImageInformation (
            const gdcm::Reader & reader) [protected]
```

12.388.3.2 New()

```
vtkGDCMPolyDataReader * vtkGDCMPolyDataReader::New () [static]
```

Examples

GenerateRTSTRUCT.cxx, gdcmscene.cxx, and rtstructapp.cxx.

References vtkGDCMPolyDataReader().

12.388.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataReader::PrintSelf (
            ostream & os,
            vtkIndent indent)  [virtual]
```

12.388.3.4 RequestData()

```
int vtkGDCMPolyDataReader::RequestData (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector * )  [protected]
```

12.388.3.5 RequestData_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (
            gdcm::Reader const & reader,
            vtkInformationVector * outputVector)  [protected]
```

References vtkGDCMPolyDataReader().

12.388.3.6 RequestData_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (
            gdcm::Reader const & reader,
            vtkInformationVector * outputVector)  [protected]
```

12.388.3.7 RequestInformation()

```
int vtkGDCMPolyDataReader::RequestInformation (
            vtkInformation * vtkNotUsedrequest,
            vtkInformationVector ** vtkNotUsedinputVector,
            vtkInformationVector * outputVector)  [protected]
```

12.388.3.8 RequestInformation_HemodynamicWaveformStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (
            gdcm::Reader const & reader)  [protected]
```

12.388.3.9 RequestInformation_RTStructureSetStorage()

```
int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (
            gdcm::Reader const & reader)  [protected]
```

12.388.3.10 vtkGetObjectMacro() [1/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
            MedicalImageProperties ,
            vtkMedicalImageProperties )
```

References MedicalImageProperties.

12.388.3.11 vtkGetObjectMacro() [2/2]

```
vtkGDCMPolyDataReader::vtkGetObjectMacro (
            RTStructSetProperties ,
            vtkRTStructSetProperties )
```

References RTStructSetProperties.

12.388.3.12 vtkGetStringMacro()

```
vtkGDCMPolyDataReader::vtkGetStringMacro (
            FileName )
```

References FileName.

12.388.3.13 vtkSetStringMacro()

```
vtkGDCMPolyDataReader::vtkSetStringMacro (
            FileName )
```

References FileName.

12.388.3.14 vtkTypeMacro()

```
vtkGDCMPolyDataReader::vtkTypeMacro (
            vtkGDCMPolyDataReader ,
            vtkPolyDataAlgorithm )
```

References vtkGDCMPolyDataReader().

12.388.4 Member Data Documentation

12.388.4.1 FileName

```
char* vtkGDCMPolyDataReader::FileName  [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.388.4.2 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties  [protected]
```

Referenced by vtkGetObjectMacro().

12.388.4.3 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties  [protected]
```

Referenced by vtkGetObjectMacro().

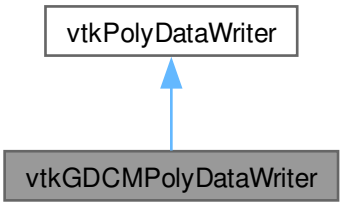
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataReader.h

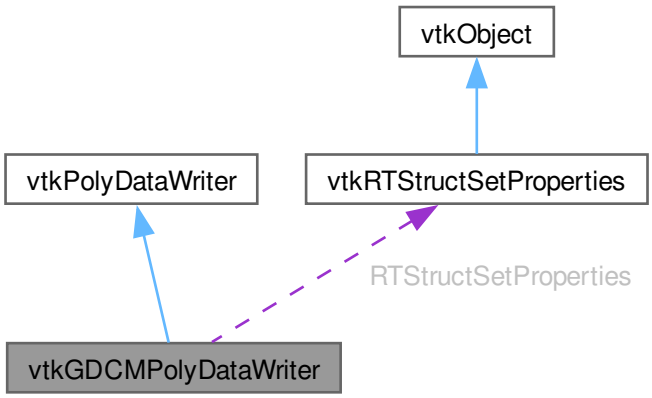
12.389 vtkGDCMPolyDataWriter Class Reference

```
#include <vtkGDCMPolyDataWriter.h>
```

Inheritance diagram for vtkGDCMPolyDataWriter:



Collaboration diagram for vtkGDCMPolyDataWriter:



Public Member Functions

- void InitializeRTStructSet (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtk↩ StringArray ∗inROINames, vtkStringArray ∗inROIAlgorithmName, vtkStringArray ∗inROIType)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗pd)
- void SetNumberOfInputPorts (int n)
- virtual void SetRTStructSetProperties (vtkRTStructSetProperties ∗pd)
- vtkTypeMacro (vtkGDCMPolyDataWriter, vtkPolyDataWriter)

Static Public Member Functions

- static vtkGDCMPolyDataWriter ∗ New ()

Protected Member Functions

- vtkGDCMPolyDataWriter ()
- ∼vtkGDCMPolyDataWriter ()
- void WriteData ()
- void WriteRTSTRUCTData (gdcm::File &file, int num)
- void WriteRTSTRUCTInfo (gdcm::File &file)

Protected Attributes

- vtkMedicalImageProperties ∗ MedicalImageProperties
- vtkRTStructSetProperties ∗ RTStructSetProperties

12.389.1 Detailed Description

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

12.389.2 Constructor & Destructor Documentation

12.389.2.1 vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter () [protected]
```

Referenced by New(), vtkTypeMacro(), and WriteRTSTRUCTData().

12.389.2.2 ~vtkGDCMPolyDataWriter()

```
vtkGDCMPolyDataWriter::~vtkGDCMPolyDataWriter () [protected]
```

12.389.3 Member Function Documentation

12.389.3.1 InitializeRTStructSet()

```
void vtkGDCMPolyDataWriter::InitializeRTStructSet (
            vtkStdString inDirectory,
            vtkStdString inStructLabel,
            vtkStdString inStructName,
            vtkStringArray * inROINames,
            vtkStringArray * inROIAlgorithmName,
            vtkStringArray * inROIType)
```

Examples

GenerateRTSTRUCT.cxx.

12.389.3.2 New()

```
vtkGDCMPolyDataWriter * vtkGDCMPolyDataWriter::New () [static]
```

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

References vtkGDCMPolyDataWriter().

12.389.3.3 PrintSelf()

```
virtual void vtkGDCMPolyDataWriter::PrintSelf (
            ostream & os,
            vtkIndent indent)  [virtual]
```

12.389.3.4 SetMedicalImageProperties()

```
virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (
            vtkMedicalImageProperties * pd)  [virtual]
```

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

12.389.3.5 SetNumberOfInputPorts()

```
void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (
            int n)
```

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

12.389.3.6 SetRTStructSetProperties()

```
virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (
            vtkRTStructSetProperties * pd)  [virtual]
```

Examples

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

12.389.3.7 vtkTypeMacro()

```
vtkGDCMPolyDataWriter::vtkTypeMacro (
            vtkGDCMPolyDataWriter ,
            vtkPolyDataWriter )
```

References vtkGDCMPolyDataWriter().

12.389.3.8 WriteData()

```
void vtkGDCMPolyDataWriter::WriteData () [protected]
```

12.389.3.9 WriteRTSTRUCTData()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (
            gdcm::File & file,
            int num) [protected]
```

References vtkGDCMPolyDataWriter().

12.389.3.10 WriteRTSTRUCTInfo()

```
void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (
            gdcm::File & file) [protected]
```

12.389.4 Member Data Documentation

12.389.4.1 MedicalImageProperties

```
vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]
```

12.389.4.2 RTStructSetProperties

```
vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties [protected]
```

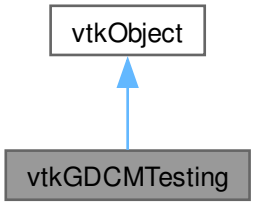
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataWriter.h

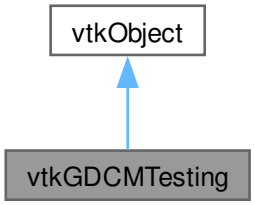
12.390 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Inheritance diagram for vtkGDCMTesting:



Collaboration diagram for vtkGDCMTesting:



Public Types

- typedef const char ∗const (∗ MD5MetaImagesType)[3]

Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkGDCMTesting, vtkObject)

Static Public Member Functions

- static const char ∗ GetGDCMDataRoot ()
- static const char ∗const ∗ GetMD5MetaImage (unsigned int file)
- static const char ∗ GetMHDMD5FromFile (const char ∗filepath)
- static unsigned int GetNumberOfMD5MetaImages ()
- static const char ∗ GetRAWMD5FromFile (const char ∗filepath)
- static const char ∗ GetVTKDataRoot ()
- static vtkGDCMTesting ∗ New ()

Protected Member Functions

- vtkGDCMTesting ()
- ∼vtkGDCMTesting ()

12.390.1 Detailed Description

Examples

HelloActiviz5.cs, HelloVTKWorld2.cs, MetaImageMD5Activiz.cs, ReadSeriesIntoVTK.java, and RefCounting.cs.

12.390.2 Member Typedef Documentation

12.390.2.1 MD5MetaImagesType

```
typedef const char* const(* vtkGDCMTesting::MD5MetaImagesType)[3]
```

12.390.3 Constructor & Destructor Documentation

12.390.3.1 vtkGDCMTesting()

```
vtkGDCMTesting::vtkGDCMTesting ()  [protected]
```

Referenced by ∼vtkGDCMTesting(), New(), and vtkTypeMacro().

12.390.3.2 ∼vtkGDCMTesting()

```
vtkGDCMTesting::∼vtkGDCMTesting ()  [protected]
```

References vtkGDCMTesting().

12.390.4 Member Function Documentation

12.390.4.1 GetGDCMDataRoot()

```
const char * vtkGDCMTesting::GetGDCMDataRoot ()  [static]
```

Examples

HelloActiviz5.cs, and ReadSeriesIntoVTK.java.

12.390.4.2 GetMD5MetaImage()

```
const char *const * vtkGDCMTesting::GetMD5MetaImage (
            unsigned int file)  [static]
```

12.390.4.3 GetMHDMD5FromFile()

```
const char * vtkGDCMTesting::GetMHDMD5FromFile (
            const char * filepath)  [static]
```

Examples

MetaImageMD5Activiz.cs.

12.390.4.4 GetNumberOfMD5MetaImages()

```
unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ()  [static]
```

12.390.4.5 GetRAWMD5FromFile()

```
const char * vtkGDCMTesting::GetRAWMD5FromFile (
            const char * filepath)  [static]
```

Examples

MetaImageMD5Activiz.cs.

12.390.4.6 GetVTKDataRoot()

```
const char * vtkGDCMTesting::GetVTKDataRoot ()  [static]
```

Examples

HelloActiviz5.cs, and HelloVTKWorld2.cs.

12.390.4.7 New()

```
vtkGDCMTesting * vtkGDCMTesting::New ()  [static]
```

Examples

RefCounting.cs.

References vtkGDCMTesting().

12.390.4.8 PrintSelf()

```
void vtkGDCMTesting::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.390.4.9 vtkTypeMacro()

```
vtkGDCMTesting::vtkTypeMacro (
            vtkGDCMTesting ,
            vtkObject )
```

References vtkGDCMTesting().

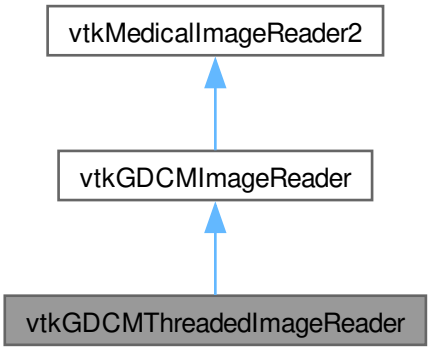
The documentation for this class was generated from the following file:

- vtkGDCMTesting.h

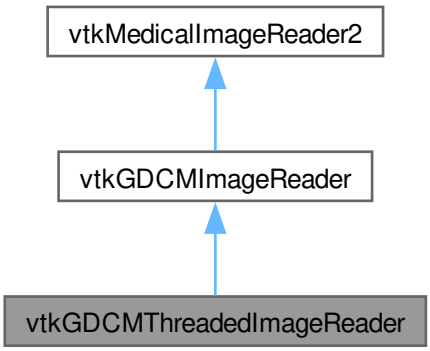
12.391 vtkGDCMThreadedImageReader Class Reference

```
#include <vtkGDCMThreadedImageReader.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader:



Collaboration diagram for vtkGDCMThreadedImageReader:



Public Member Functions

- virtual void PrintSelf (ostream &os, vtkIndent indent)
- vtkBooleanMacro (UseShiftScale, int)
- vtkGetMacro (UseShiftScale, int)
- vtkSetMacro (Scale, double)
- vtkSetMacro (Shift, double)
- vtkSetMacro (UseShiftScale, int)
- vtkTypeMacro (vtkGDCMThreadedImageReader, vtkGDCMImageReader)

Public Member Functions inherited from vtkGDCMImageReader

- virtual int CanReadFile (const char ∗fname)
- virtual const char ∗ GetDescriptiveName ()
- virtual const char ∗ GetFileExtensions ()
- vtkImageData ∗ GetIconImage ()
- vtkImageData ∗ GetOverlay (int i)
- virtual void SetCurve (vtkPolyData ∗pd)
- virtual void SetFileNames (vtkStringArray ∗)
- virtual void SetMedicalImageProperties (vtkMedicalImageProperties ∗pd)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (LoadIconImage, int)

- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (ApplyLookupTable, int)
- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LossyFlag, int)
- vtkTypeMacro (vtkGDCMImageReader, vtkMedicalImageReader2)

Static Public Member Functions

- static vtkGDCMThreadedImageReader ∗ New ()

Static Public Member Functions inherited from vtkGDCMImageReader

- static vtkGDCMImageReader ∗ New ()

Protected Member Functions

- vtkGDCMThreadedImageReader ()
- ∼vtkGDCMThreadedImageReader ()
- void ExecuteData (vtkDataObject ∗out)
- void ExecuteInformation ()
- void ReadFiles (unsigned int nfiles, const char ∗filenames[ ])
- void RequestDataCompat ()

Protected Member Functions inherited from vtkGDCMImageReader

- vtkGDCMImageReader ()
- ∼vtkGDCMImageReader ()
- void ExecuteData (vtkDataObject ∗out)
- void ExecuteInformation ()
- void FillMedicalImageInformation (const gdcm::ImageReader &reader)
- int LoadSingleFile (const char ∗filename, char ∗pointer, unsigned long &outlen)
- int RequestDataCompat ()
- int RequestInformationCompat ()
- void SetFilePattern (const char ∗)
- void SetFilePrefix (const char ∗)
- vtkGetStringMacro (FilePattern)
- vtkGetStringMacro (FilePrefix)
- vtkSetVector6Macro (ImageOrientationPatient, double)

Additional Inherited Members

Protected Attributes inherited from vtkGDCMImageReader

- int ApplyInverseVideo
- int ApplyLookupTable
- int ApplyPlanarConfiguration
- int ApplyShiftScale
- int ApplyYBRToRGB
- vtkPolyData * Curve
- vtkMatrix4x4 * DirectionCosines
- vtkStringArray * FileNames
- int ForceRescale
- int IconDataScalarType
- int IconImageDataExtent [6]
- int IconNumberOfScalarComponents
- int ImageFormat
- double ImageOrientationPatient [6]
- double ImagePositionPatient [3]
- int LoadIconImage
- int LoadOverlays
- int LossyFlag
- vtkMedicalImageProperties * MedicalImageProperties
- int NumberOfIconImages
- int NumberOfOverlays
- int PlanarConfiguration
- double Scale
- double Shift

12.391.1 Constructor & Destructor Documentation

12.391.1.1 vtkGDCMThreadedImageReader()

```
vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ()  [protected]
```

Referenced by New(), RequestDataCompat(), and vtkTypeMacro().

12.391.1.2 ~vtkGDCMThreadedImageReader()

```
vtkGDCMThreadedImageReader::~vtkGDCMThreadedImageReader ()  [protected]
```

12.391.2 Member Function Documentation

12.391.2.1 ExecuteData()

```
void vtkGDCMThreadedImageReader::ExecuteData (
            vtkDataObject * out)  [protected]
```

12.391.2.2 ExecuteInformation()

```
void vtkGDCMThreadedImageReader::ExecuteInformation () [protected]
```

12.391.2.3 New()

```
vtkGDCMThreadedImageReader * vtkGDCMThreadedImageReader::New () [static]
```

References vtkGDCMThreadedImageReader().

12.391.2.4 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader::PrintSelf (
            ostream & os,
            vtkIndent indent) [virtual]
```

Reimplemented from vtkGDCMImageReader.

12.391.2.5 ReadFiles()

```
void vtkGDCMThreadedImageReader::ReadFiles (
            unsigned int nfiles,
            const char * filenames[]) [protected]
```

12.391.2.6 RequestDataCompat()

```
void vtkGDCMThreadedImageReader::RequestDataCompat () [protected]
```

References vtkGDCMThreadedImageReader().

12.391.2.7 vtkBooleanMacro()

```
vtkGDCMThreadedImageReader::vtkBooleanMacro (
            UseShiftScale ,
            int )
```

12.391.2.8 vtkGetMacro()

```
vtkGDCMThreadedImageReader::vtkGetMacro (
            UseShiftScale ,
            int )
```

12.391.2.9 vtkSetMacro() [1/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            Scale ,
            double )
```

References vtkGDCMImageReader::Scale.

12.391.2.10 vtkSetMacro() [2/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            Shift ,
            double )
```

References vtkGDCMImageReader::Shift.

12.391.2.11 vtkSetMacro() [3/3]

```
vtkGDCMThreadedImageReader::vtkSetMacro (
            UseShiftScale ,
            int )
```

12.391.2.12 vtkTypeMacro()

```
vtkGDCMThreadedImageReader::vtkTypeMacro (
            vtkGDCMThreadedImageReader ,
            vtkGDCMImageReader )
```

References vtkGDCMImageReader::vtkGDCMImageReader(), and vtkGDCMThreadedImageReader().

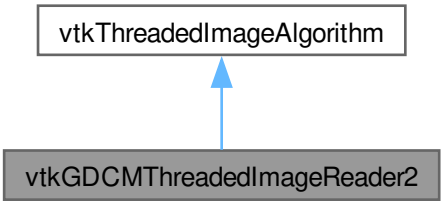
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader.h

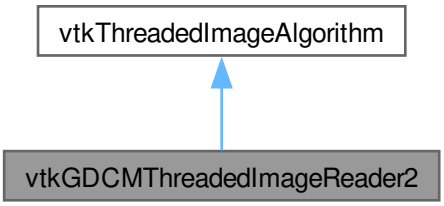
12.392 vtkGDCMThreadedImageReader2 Class Reference

```
#include <vtkGDCMThreadedImageReader2.h>
```

Inheritance diagram for vtkGDCMThreadedImageReader2:



Collaboration diagram for vtkGDCMThreadedImageReader2:



Public Member Functions

- virtual const char ∗ GetFileName (int i=0)
- virtual void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetFileName (const char ∗filename)
- virtual void SetFileNames (vtkStringArray ∗)
- int SplitExtent (int splitExt[6], int startExt[6], int num, int total)
- vtkBooleanMacro (FileLowerLeft, int)
- vtkBooleanMacro (LoadOverlays, int)
- vtkBooleanMacro (UseShiftScale, int)
- vtkGetMacro (DataScalarType, int)
- vtkGetMacro (FileLowerLeft, int)
- vtkGetMacro (LoadOverlays, int)

- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (NumberOfScalarComponents, int)
- vtkGetMacro (Scale, double)
- vtkGetMacro (Shift, double)
- vtkGetMacro (UseShiftScale, int)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetVector3Macro (DataOrigin, double)
- vtkGetVector3Macro (DataSpacing, double)
- vtkGetVector6Macro (DataExtent, int)
- vtkSetMacro (DataScalarType, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (NumberOfScalarComponents, int)
- vtkSetMacro (Scale, double)
- vtkSetMacro (Shift, double)
- vtkSetMacro (UseShiftScale, int)
- vtkSetVector3Macro (DataOrigin, double)
- vtkSetVector3Macro (DataSpacing, double)
- vtkSetVector6Macro (DataExtent, int)
- vtkTypeMacro (vtkGDCMThreadedImageReader2, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkGDCMThreadedImageReader2 * New ()

Protected Member Functions

- vtkGDCMThreadedImageReader2 ()
- ~vtkGDCMThreadedImageReader2 ()
- int RequestInformation (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector)
- void ThreadedRequestData (vtkInformation *request, vtkInformationVector **inputVector, vtkInformationVector *outputVector, vtkImageData ***inData, vtkImageData **outData, int outExt[6], int id)

12.392.1 Constructor & Destructor Documentation

12.392.1.1 vtkGDCMThreadedImageReader2()

```
vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ()  [protected]
```

Referenced by New(), ThreadedRequestData(), and vtkTypeMacro().

12.392.1.2 ~vtkGDCMThreadedImageReader2()

```
vtkGDCMThreadedImageReader2::~vtkGDCMThreadedImageReader2 ()  [protected]
```

12.392.2 Member Function Documentation

12.392.2.1 GetFileName()

```
virtual const char * vtkGDCMThreadedImageReader2::GetFileName (
            int i = 0)  [virtual]
```

12.392.2.2 New()

```
vtkGDCMThreadedImageReader2 * vtkGDCMThreadedImageReader2::New ()  [static]
```

References vtkGDCMThreadedImageReader2().

12.392.2.3 PrintSelf()

```
virtual void vtkGDCMThreadedImageReader2::PrintSelf (
            ostream & os,
            vtkIndent indent)  [virtual]
```

12.392.2.4 RequestInformation()

```
int vtkGDCMThreadedImageReader2::RequestInformation (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector)  [protected]
```

12.392.2.5 SetFileName()

```
virtual void vtkGDCMThreadedImageReader2::SetFileName (
            const char * filename)  [virtual]
```

12.392.2.6 SetFileNames()

```
virtual void vtkGDCMThreadedImageReader2::SetFileNames (
            vtkStringArray * )  [virtual]
```

12.392.2.7 SplitExtent()

```
int vtkGDCMThreadedImageReader2::SplitExtent (
            int splitExt[6],
            int startExt[6],
            int num,
            int total)
```

12.392.2.8 ThreadedRequestData()

```
void vtkGDCMThreadedImageReader2::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int outExt[6],
            int id)  [protected]
```

References vtkGDCMThreadedImageReader2().

12.392.2.9 vtkBooleanMacro() [1/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            FileLowerLeft ,
            int )
```

12.392.2.10 vtkBooleanMacro() [2/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            LoadOverlays ,
            int )
```

12.392.2.11 vtkBooleanMacro() [3/3]

```
vtkGDCMThreadedImageReader2::vtkBooleanMacro (
            UseShiftScale ,
            int )
```

12.392.2.12 vtkGetMacro() [1/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            DataScalarType ,
            int )
```

12.392.2.13 vtkGetMacro() [2/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            FileLowerLeft ,
            int )
```

12.392.2.14 vtkGetMacro() [3/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            LoadOverlays ,
            int )
```

12.392.2.15 vtkGetMacro() [4/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            NumberOfOverlays ,
            int )
```

12.392.2.16 vtkGetMacro() [5/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            NumberOfScalarComponents ,
            int )
```

12.392.2.17 vtkGetMacro() [6/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            Scale ,
            double )
```

12.392.2.18 vtkGetMacro() [7/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            Shift ,
            double )
```

12.392.2.19 vtkGetMacro() [8/8]

```
vtkGDCMThreadedImageReader2::vtkGetMacro (
            UseShiftScale ,
            int )
```

12.392.2.20 vtkGetObjectMacro()

```
vtkGDCMThreadedImageReader2::vtkGetObjectMacro (
            FileNames ,
            vtkStringArray )
```

12.392.2.21 vtkGetVector3Macro() [1/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
            DataOrigin ,
            double )
```

12.392.2.22 vtkGetVector3Macro() [2/2]

```
vtkGDCMThreadedImageReader2::vtkGetVector3Macro (
            DataSpacing ,
            double )
```

12.392.2.23 vtkGetVector6Macro()

```
vtkGDCMThreadedImageReader2::vtkGetVector6Macro (
            DataExtent ,
            int )
```

12.392.2.24 vtkSetMacro() [1/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            DataScalarType ,
            int )
```

12.392.2.25 vtkSetMacro() [2/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            FileLowerLeft ,
            int )
```

12.392.2.26 vtkSetMacro() [3/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            LoadOverlays ,
            int )
```

12.392.2.27 vtkSetMacro() [4/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            NumberOfScalarComponents ,
            int )
```

12.392.2.28 vtkSetMacro() [5/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            Scale ,
            double )
```

12.392.2.29 vtkSetMacro() [6/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            Shift ,
            double )
```

12.392.2.30 vtkSetMacro() [7/7]

```
vtkGDCMThreadedImageReader2::vtkSetMacro (
            UseShiftScale ,
            int )
```

12.392.2.31 vtkSetVector3Macro() [1/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
            DataOrigin ,
            double )
```

12.392.2.32 vtkSetVector3Macro() [2/2]

```
vtkGDCMThreadedImageReader2::vtkSetVector3Macro (
            DataSpacing ,
            double )
```

12.392.2.33 vtkSetVector6Macro()

```
vtkGDCMThreadedImageReader2::vtkSetVector6Macro (
            DataExtent ,
            int )
```

12.392.2.34 vtkTypeMacro()

```
vtkGDCMThreadedImageReader2::vtkTypeMacro (
            vtkGDCMThreadedImageReader2 ,
            vtkThreadedImageAlgorithm )
```

References vtkGDCMThreadedImageReader2().

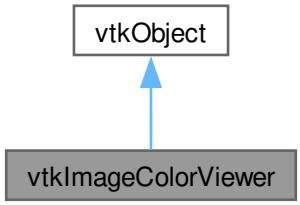
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader2.h

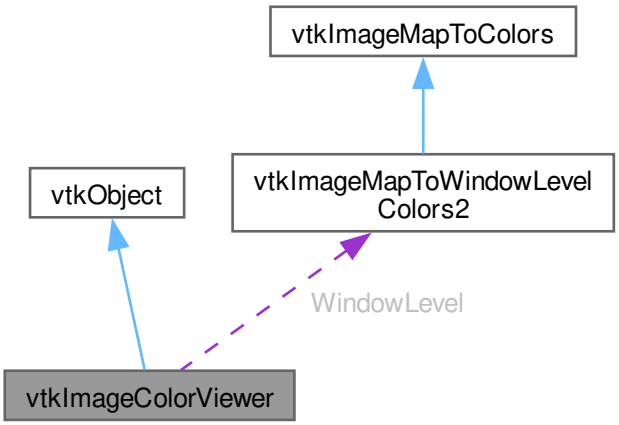
12.393 vtkImageColorViewer Class Reference

```
#include <vtkImageColorViewer.h>
```

Inheritance diagram for vtkImageColorViewer:



Collaboration diagram for vtkImageColorViewer:



Public Types

- enum {
  SLICE_ORIENTATION_YZ = 0 ,
  SLICE_ORIENTATION_XZ = 1 ,
  SLICE_ORIENTATION_XY = 2 }

Public Member Functions

- virtual void AddInput (vtkImageData *input)
- virtual void AddInputConnection (vtkAlgorithmOutput *input)
- virtual double GetColorLevel ()
- virtual double GetColorWindow ()
- virtual vtkImageData * GetInput ()
- virtual int GetOffScreenRendering ()
- double GetOverlayVisibility ()
- virtual int * GetPosition ()
- virtual int * GetSize ()
- virtual int GetSliceMax ()
- virtual int GetSliceMin ()
- virtual int * GetSliceRange ()
- virtual void GetSliceRange (int &min, int &max)
- virtual void GetSliceRange (int range[2])
- virtual const char * GetWindowName ()
- void PrintSelf (ostream &os, vtkIndent indent)
- virtual void Render (void)
- virtual void SetColorLevel (double s)
- virtual void SetColorWindow (double s)
- virtual void SetDisplayId (void *a)
- virtual void SetInput (vtkImageData *in)
- virtual void SetInputConnection (vtkAlgorithmOutput *input)
- virtual void SetOffScreenRendering (int)
- void SetOverlayVisibility (double vis)
- virtual void SetParentId (void *a)
- virtual void SetPosition (int a, int b)
- virtual void SetPosition (int a[2])
- virtual void SetRenderer (vtkRenderer *arg)
- virtual void SetRenderWindow (vtkRenderWindow *arg)
- virtual void SetSize (int a, int b)
- virtual void SetSize (int a[2])
- virtual void SetSlice (int s)
- virtual void SetSliceOrientation (int orientation)
- virtual void SetSliceOrientationToXY ()
- virtual void SetSliceOrientationToXZ ()
- virtual void SetSliceOrientationToYZ ()
- virtual void SetupInteractor (vtkRenderWindowInteractor *)
- virtual void SetWindowId (void *a)
- virtual void UpdateDisplayExtent ()
- vtkBooleanMacro (OffScreenRendering, int)
- vtkGetMacro (Slice, int)
- vtkGetMacro (SliceOrientation, int)
- vtkGetObjectMacro (ImageActor, vtkImageActor)
- vtkGetObjectMacro (InteractorStyle, vtkInteractorStyleImage)
- vtkGetObjectMacro (Renderer, vtkRenderer)
- vtkGetObjectMacro (RenderWindow, vtkRenderWindow)
- vtkGetObjectMacro (WindowLevel, vtkImageMapToWindowLevelColors2)
- vtkTypeMacro (vtkImageColorViewer, vtkObject)

Static Public Member Functions

- static vtkImageColorViewer ∗ New ()

Protected Member Functions

- vtkImageColorViewer ()
- ∼vtkImageColorViewer ()
- virtual void InstallPipeline ()
- virtual void UnInstallPipeline ()
- virtual void UpdateOrientation ()

Protected Attributes

- int FirstRender
- vtkImageActor ∗ ImageActor
- vtkRenderWindowInteractor ∗ Interactor
- vtkInteractorStyleImage ∗ InteractorStyle
- vtkImageActor ∗ OverlayImageActor
- vtkRenderer ∗ Renderer
- vtkRenderWindow ∗ RenderWindow
- int Slice
- int SliceOrientation
- vtkImageMapToWindowLevelColors2 ∗ WindowLevel

Friends

- class vtkImageColorViewerCallback

12.393.1 Detailed Description

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.393.2 Member Enumeration Documentation

12.393.2.1 anonymous enum

```
anonymous enum
```

Enumerator

SLICE_ORIENTATION_YZ	
SLICE_ORIENTATION_XZ	
SLICE_ORIENTATION_XY	

12.393.3 Constructor & Destructor Documentation

12.393.3.1 vtkImageColorViewer()

```
vtkImageColorViewer::vtkImageColorViewer ()  [protected]
```

Referenced by New(), vtkImageColorViewerCallback, and vtkTypeMacro().

12.393.3.2 ~vtkImageColorViewer()

```
vtkImageColorViewer::~vtkImageColorViewer ()  [protected]
```

12.393.4 Member Function Documentation

12.393.4.1 AddInput()

```
virtual void vtkImageColorViewer::AddInput (
            vtkImageData * input)  [virtual]
```

12.393.4.2 AddInputConnection()

```
virtual void vtkImageColorViewer::AddInputConnection (
            vtkAlgorithmOutput * input)  [virtual]
```

12.393.4.3 GetColorLevel()

```
virtual double vtkImageColorViewer::GetColorLevel ()  [virtual]
```

12.393.4.4 GetColorWindow()

```
virtual double vtkImageColorViewer::GetColorWindow ()  [virtual]
```

12.393.4.5 GetInput()

```
virtual vtkImageData * vtkImageColorViewer::GetInput ()  [virtual]
```

12.393.4.6 GetOffScreenRendering()

```
virtual int vtkImageColorViewer::GetOffScreenRendering () [virtual]
```

12.393.4.7 GetOverlayVisibility()

```
double vtkImageColorViewer::GetOverlayVisibility ()
```

12.393.4.8 GetPosition()

```
virtual int * vtkImageColorViewer::GetPosition () [virtual]
```

12.393.4.9 GetSize()

```
virtual int * vtkImageColorViewer::GetSize () [virtual]
```

12.393.4.10 GetSliceMax()

```
virtual int vtkImageColorViewer::GetSliceMax () [virtual]
```

12.393.4.11 GetSliceMin()

```
virtual int vtkImageColorViewer::GetSliceMin () [virtual]
```

12.393.4.12 GetSliceRange() [1/3]

```
virtual int * vtkImageColorViewer::GetSliceRange () [virtual]
```

12.393.4.13 GetSliceRange() [2/3]

```
virtual void vtkImageColorViewer::GetSliceRange (
            int & min,
            int & max) [virtual]
```

12.393.4.14 GetSliceRange() [3/3]

```
virtual void vtkImageColorViewer::GetSliceRange (
            int range[2]) [inline], [virtual]
```

References GetSliceRange().

Referenced by GetSliceRange().

12.393.4.15 GetWindowName()

```
virtual const char * vtkImageColorViewer::GetWindowName ()  [virtual]
```

12.393.4.16 InstallPipeline()

```
virtual void vtkImageColorViewer::InstallPipeline ()  [protected], [virtual]
```

12.393.4.17 New()

```
vtkImageColorViewer * vtkImageColorViewer::New ()  [static]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

References vtkImageColorViewer().

12.393.4.18 PrintSelf()

```
void vtkImageColorViewer::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.393.4.19 Render()

```
virtual void vtkImageColorViewer::Render (
            void )  [virtual]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.393.4.20 SetColorLevel()

```
virtual void vtkImageColorViewer::SetColorLevel (
            double s)  [virtual]
```

12.393.4.21 SetColorWindow()

```
virtual void vtkImageColorViewer::SetColorWindow (
            double s)  [virtual]
```

12.393.4.22 SetDisplayId()

```
virtual void vtkImageColorViewer::SetDisplayId (
            void * a)  [virtual]
```

12.393.4.23 SetInput()

```
virtual void vtkImageColorViewer::SetInput (
            vtkImageData * in)  [virtual]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.393.4.24 SetInputConnection()

```
virtual void vtkImageColorViewer::SetInputConnection (
            vtkAlgorithmOutput * input)  [virtual]
```

12.393.4.25 SetOffScreenRendering()

```
virtual void vtkImageColorViewer::SetOffScreenRendering (
            int )  [virtual]
```

12.393.4.26 SetOverlayVisibility()

```
void vtkImageColorViewer::SetOverlayVisibility (
            double vis)
```

12.393.4.27 SetParentId()

```
virtual void vtkImageColorViewer::SetParentId (
            void * a)  [virtual]
```

12.393.4.28 SetPosition() [1/2]

```
virtual void vtkImageColorViewer::SetPosition (
            int a,
            int b)  [virtual]
```

12.393.4.29 SetPosition() [2/2]

```
virtual void vtkImageColorViewer::SetPosition (
            int a[2])  [inline], [virtual]
```

References SetPosition().

Referenced by SetPosition().

12.393.4.30 SetRenderer()

```
virtual void vtkImageColorViewer::SetRenderer (
            vtkRenderer * arg)  [virtual]
```

12.393.4.31 SetRenderWindow()

```
virtual void vtkImageColorViewer::SetRenderWindow (
            vtkRenderWindow * arg)  [virtual]
```

12.393.4.32 SetSize() [1/2]

```
virtual void vtkImageColorViewer::SetSize (
            int a,
            int b)  [virtual]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.393.4.33 SetSize() [2/2]

```
virtual void vtkImageColorViewer::SetSize (
            int a[2])  [inline], [virtual]
```

References SetSize().

Referenced by SetSize().

12.393.4.34 SetSlice()

```
virtual void vtkImageColorViewer::SetSlice (
            int s)  [virtual]
```

12.393.4.35 SetSliceOrientation()

```
virtual void vtkImageColorViewer::SetSliceOrientation (
            int orientation)  [virtual]
```

Referenced by SetSliceOrientationToXY(), SetSliceOrientationToXZ(), and SetSliceOrientationToYZ().

12.393.4.36 SetSliceOrientationToXY()

```
virtual void vtkImageColorViewer::SetSliceOrientationToXY ()  [inline], [virtual]
```

References SetSliceOrientation(), and SLICE_ORIENTATION_XY.

12.393.4.37 SetSliceOrientationToXZ()

```
virtual void vtkImageColorViewer::SetSliceOrientationToXZ ()  [inline], [virtual]
```

References SetSliceOrientation(), and SLICE_ORIENTATION_XZ.

12.393.4.38 SetSliceOrientationToYZ()

```
virtual void vtkImageColorViewer::SetSliceOrientationToYZ ()  [inline], [virtual]
```

References SetSliceOrientation(), and SLICE_ORIENTATION_YZ.

12.393.4.39 SetupInteractor()

```
virtual void vtkImageColorViewer::SetupInteractor (
            vtkRenderWindowInteractor * )  [virtual]
```

Examples

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

12.393.4.40 SetWindowId()

```
virtual void vtkImageColorViewer::SetWindowId (
            void * a)  [virtual]
```

12.393.4.41 UnInstallPipeline()

```
virtual void vtkImageColorViewer::UnInstallPipeline ()  [protected], [virtual]
```

12.393.4.42 UpdateDisplayExtent()

```
virtual void vtkImageColorViewer::UpdateDisplayExtent ()  [virtual]
```

12.393.4.43 UpdateOrientation()

```
virtual void vtkImageColorViewer::UpdateOrientation ()  [protected], [virtual]
```

12.393.4.44 vtkBooleanMacro()

```
vtkImageColorViewer::vtkBooleanMacro (
            OffScreenRendering ,
            int )
```

12.393.4.45 vtkGetMacro() [1/2]

```
vtkImageColorViewer::vtkGetMacro (
            Slice ,
            int )
```

References Slice.

12.393.4.46 vtkGetMacro() [2/2]

```
vtkImageColorViewer::vtkGetMacro (
            SliceOrientation ,
            int )
```

References SliceOrientation.

12.393.4.47 vtkGetObjectMacro() [1/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            ImageActor ,
            vtkImageActor )
```

References ImageActor.

12.393.4.48 vtkGetObjectMacro() [2/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            InteractorStyle ,
            vtkInteractorStyleImage )
```

References InteractorStyle.

12.393.4.49 vtkGetObjectMacro() [3/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            Renderer ,
            vtkRenderer )
```

References Renderer.

12.393.4.50 vtkGetObjectMacro() [4/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            RenderWindow ,
            vtkRenderWindow )
```

References RenderWindow.

12.393.4.51 vtkGetObjectMacro() [5/5]

```
vtkImageColorViewer::vtkGetObjectMacro (
            WindowLevel ,
            vtkImageMapToWindowLevelColors2 )
```

References WindowLevel.

12.393.4.52 vtkTypeMacro()

```
vtkImageColorViewer::vtkTypeMacro (
            vtkImageColorViewer ,
            vtkObject )
```

References vtkImageColorViewer().

12.393.5 Friends And Related Symbol Documentation

12.393.5.1 vtkImageColorViewerCallback

```
friend class vtkImageColorViewerCallback  [friend]
```

References vtkImageColorViewer(), and vtkImageColorViewerCallback.

Referenced by vtkImageColorViewerCallback.

12.393.6 Member Data Documentation

12.393.6.1 FirstRender

```
int vtkImageColorViewer::FirstRender [protected]
```

12.393.6.2 ImageActor

```
vtkImageActor* vtkImageColorViewer::ImageActor [protected]
```

Referenced by vtkGetObjectMacro().

12.393.6.3 Interactor

```
vtkRenderWindowInteractor* vtkImageColorViewer::Interactor [protected]
```

12.393.6.4 InteractorStyle

```
vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle [protected]
```

Referenced by vtkGetObjectMacro().

12.393.6.5 OverlayImageActor

```
vtkImageActor* vtkImageColorViewer::OverlayImageActor [protected]
```

12.393.6.6 Renderer

```
vtkRenderer* vtkImageColorViewer::Renderer [protected]
```

Referenced by vtkGetObjectMacro().

12.393.6.7 RenderWindow

```
vtkRenderWindow* vtkImageColorViewer::RenderWindow [protected]
```

Referenced by vtkGetObjectMacro().

12.393.6.8 Slice

```
int vtkImageColorViewer::Slice [protected]
```

Referenced by vtkGetMacro().

12.393.6.9 SliceOrientation

```
int vtkImageColorViewer::SliceOrientation [protected]
```

Referenced by vtkGetMacro().

12.393.6.10 WindowLevel

```
vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel [protected]
```

Referenced by vtkGetObjectMacro().

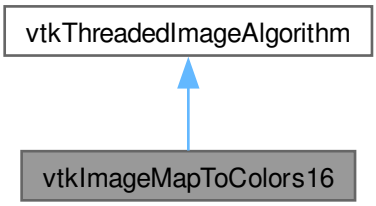
The documentation for this class was generated from the following file:

- vtkImageColorViewer.h

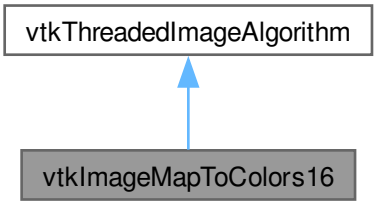
12.394 vtkImageMapToColors16 Class Reference

```
#include <vtkImageMapToColors16.h>
```

Inheritance diagram for vtkImageMapToColors16:



Collaboration diagram for vtkImageMapToColors16:



Public Member Functions

- virtual unsigned long GetMTime ()
- void PrintSelf (ostream &os, vtkIndent indent)
- virtual void SetLookupTable (vtkScalarsToColors ∗)
- void SetOutputFormatToLuminance ()
- void SetOutputFormatToLuminanceAlpha ()
- void SetOutputFormatToRGB ()
- void SetOutputFormatToRGBA ()
- vtkBooleanMacro (PassAlphaToOutput, int)
- vtkGetMacro (ActiveComponent, int)
- vtkGetMacro (OutputFormat, int)
- vtkGetMacro (PassAlphaToOutput, int)
- vtkGetObjectMacro (LookupTable, vtkScalarsToColors)
- vtkSetMacro (ActiveComponent, int)
- vtkSetMacro (OutputFormat, int)
- vtkSetMacro (PassAlphaToOutput, int)
- vtkTypeMacro (vtkImageMapToColors16, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageMapToColors16 ∗ New ()

Protected Member Functions

- vtkImageMapToColors16 ()
- ∼vtkImageMapToColors16 ()
- virtual int RequestData (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector)
- virtual int RequestInformation (vtkInformation ∗, vtkInformationVector ∗∗, vtkInformationVector ∗)
- void ThreadedRequestData (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector, vtkImageData ∗∗∗inData, vtkImageData ∗∗outData, int extent[6], int id)

Protected Attributes

- int ActiveComponent
- int DataWasPassed
- vtkScalarsToColors * LookupTable
- int OutputFormat
- int PassAlphaToOutput

12.394.1 Constructor & Destructor Documentation

12.394.1.1 vtkImageMapToColors16()

```
vtkImageMapToColors16::vtkImageMapToColors16 ()  [protected]
```

Referenced by New(), and vtkTypeMacro().

12.394.1.2 ~vtkImageMapToColors16()

```
vtkImageMapToColors16::~vtkImageMapToColors16 ()  [protected]
```

12.394.2 Member Function Documentation

12.394.2.1 GetMTime()

```
virtual unsigned long vtkImageMapToColors16::GetMTime ()  [virtual]
```

Referenced by vtkGetMacro().

12.394.2.2 New()

```
vtkImageMapToColors16 * vtkImageMapToColors16::New ()  [static]
```

References vtkImageMapToColors16().

12.394.2.3 PrintSelf()

```
void vtkImageMapToColors16::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.394.2.4 RequestData()

```
virtual int vtkImageMapToColors16::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector)  [protected], [virtual]
```

12.394.2.5 RequestInformation()

```
virtual int vtkImageMapToColors16::RequestInformation (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector * )  [protected], [virtual]
```

12.394.2.6 SetLookupTable()

```
virtual void vtkImageMapToColors16::SetLookupTable (
            vtkScalarsToColors * )  [virtual]
```

12.394.2.7 SetOutputFormatToLuminance()

```
void vtkImageMapToColors16::SetOutputFormatToLuminance ()  [inline]
```

References OutputFormat.

12.394.2.8 SetOutputFormatToLuminanceAlpha()

```
void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ()  [inline]
```

References OutputFormat.

12.394.2.9 SetOutputFormatToRGB()

```
void vtkImageMapToColors16::SetOutputFormatToRGB ()  [inline]
```

References OutputFormat.

12.394.2.10 SetOutputFormatToRGBA()

```
void vtkImageMapToColors16::SetOutputFormatToRGBA ()  [inline]
```

References OutputFormat.

12.394.2.11 ThreadedRequestData()

```
void vtkImageMapToColors16::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int extent[6],
            int id)  [protected]
```

12.394.2.12 vtkBooleanMacro()

```
vtkImageMapToColors16::vtkBooleanMacro (
            PassAlphaToOutput ,
            int )
```

References PassAlphaToOutput.

12.394.2.13 vtkGetMacro() [1/3]

```
vtkImageMapToColors16::vtkGetMacro (
            ActiveComponent ,
            int )
```

References ActiveComponent.

12.394.2.14 vtkGetMacro() [2/3]

```
vtkImageMapToColors16::vtkGetMacro (
            OutputFormat ,
            int )
```

References OutputFormat.

12.394.2.15 vtkGetMacro() [3/3]

```
vtkImageMapToColors16::vtkGetMacro (
            PassAlphaToOutput ,
            int )
```

References GetMTime(), and PassAlphaToOutput.

12.394.2.16 vtkGetObjectMacro()

```
vtkImageMapToColors16::vtkGetObjectMacro (
            LookupTable ,
            vtkScalarsToColors )
```

References LookupTable.

12.394.2.17 vtkSetMacro() [1/3]

```
vtkImageMapToColors16::vtkSetMacro (
            ActiveComponent ,
            int )
```

References ActiveComponent.

12.394.2.18 vtkSetMacro() [2/3]

```
vtkImageMapToColors16::vtkSetMacro (
            OutputFormat ,
            int )
```

References OutputFormat.

12.394.2.19 vtkSetMacro() [3/3]

```
vtkImageMapToColors16::vtkSetMacro (
            PassAlphaToOutput ,
            int )
```

References PassAlphaToOutput.

12.394.2.20 vtkTypeMacro()

```
vtkImageMapToColors16::vtkTypeMacro (
            vtkImageMapToColors16 ,
            vtkThreadedImageAlgorithm )
```

References vtkImageMapToColors16().

12.394.3 Member Data Documentation

12.394.3.1 ActiveComponent

```
int vtkImageMapToColors16::ActiveComponent  [protected]
```

Referenced by vtkGetMacro(), and vtkSetMacro().

12.394.3.2 DataWasPassed

```
int vtkImageMapToColors16::DataWasPassed [protected]
```

12.394.3.3 LookupTable

```
vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]
```

Referenced by vtkGetObjectMacro().

12.394.3.4 OutputFormat

```
int vtkImageMapToColors16::OutputFormat [protected]
```

Referenced by SetOutputFormatToLuminance(), SetOutputFormatToLuminanceAlpha(), SetOutputFormatToRGB(), SetOutputFormatToRGBA(), vtkGetMacro(), and vtkSetMacro().

12.394.3.5 PassAlphaToOutput

```
int vtkImageMapToColors16::PassAlphaToOutput [protected]
```

Referenced by vtkBooleanMacro(), vtkGetMacro(), and vtkSetMacro().

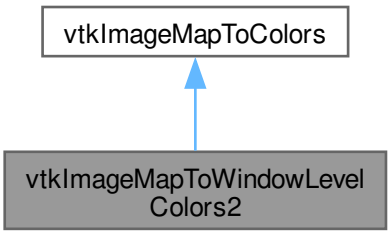
The documentation for this class was generated from the following file:

- vtkImageMapToColors16.h

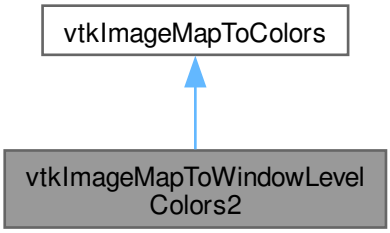
12.395 vtkImageMapToWindowLevelColors2 Class Reference

```
#include <vtkImageMapToWindowLevelColors2.h>
```

Inheritance diagram for vtkImageMapToWindowLevelColors2:



Collaboration diagram for vtkImageMapToWindowLevelColors2:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetMacro (Level, double)
- vtkGetMacro (Window, double)
- vtkSetMacro (Level, double)
- vtkSetMacro (Window, double)
- vtkTypeMacro (vtkImageMapToWindowLevelColors2, vtkImageMapToColors)

Static Public Member Functions

- static vtkImageMapToWindowLevelColors2 ∗ New ()

Protected Member Functions

- vtkImageMapToWindowLevelColors2 ()
- ∼vtkImageMapToWindowLevelColors2 ()
- virtual int RequestData (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector)
- virtual int RequestInformation (vtkInformation ∗, vtkInformationVector ∗∗, vtkInformationVector ∗)
- void ThreadedRequestData (vtkInformation ∗request, vtkInformationVector ∗∗inputVector, vtkInformationVector ∗outputVector, vtkImageData ∗∗∗inData, vtkImageData ∗∗outData, int extent[6], int id)

Protected Attributes

- double Level
- double Window

12.395.1 Constructor & Destructor Documentation

12.395.1.1 vtkImageMapToWindowLevelColors2()

```
vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ()  [protected]
```

Referenced by New(), and vtkTypeMacro().

12.395.1.2 ~vtkImageMapToWindowLevelColors2()

```
vtkImageMapToWindowLevelColors2::~vtkImageMapToWindowLevelColors2 ()  [protected]
```

12.395.2 Member Function Documentation

12.395.2.1 New()

```
vtkImageMapToWindowLevelColors2 * vtkImageMapToWindowLevelColors2::New ()  [static]
```

References vtkImageMapToWindowLevelColors2().

12.395.2.2 PrintSelf()

```
void vtkImageMapToWindowLevelColors2::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.395.2.3 RequestData()

```
virtual int vtkImageMapToWindowLevelColors2::RequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector)  [protected], [virtual]
```

12.395.2.4 RequestInformation()

```
virtual int vtkImageMapToWindowLevelColors2::RequestInformation (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector * )  [protected], [virtual]
```

12.395.2.5 ThreadedRequestData()

```
void vtkImageMapToWindowLevelColors2::ThreadedRequestData (
            vtkInformation * request,
            vtkInformationVector ** inputVector,
            vtkInformationVector * outputVector,
            vtkImageData *** inData,
            vtkImageData ** outData,
            int extent[6],
            int id)  [protected]
```

12.395.2.6 vtkGetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
            Level ,
            double )
```

References Level.

12.395.2.7 vtkGetMacro() [2/2]

```
vtkImageMapToWindowLevelColors2::vtkGetMacro (
            Window ,
            double )
```

References Window.

12.395.2.8 vtkSetMacro() [1/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
            Level ,
            double )
```

References Level.

12.395.2.9 vtkSetMacro() [2/2]

```
vtkImageMapToWindowLevelColors2::vtkSetMacro (
            Window ,
            double )
```

References Window.

12.395.2.10 vtkTypeMacro()

```
vtkImageMapToWindowLevelColors2::vtkTypeMacro (
            vtkImageMapToWindowLevelColors2 ,
            vtkImageMapToColors )
```

References vtkImageMapToWindowLevelColors2().

12.395.3 Member Data Documentation

12.395.3.1 Level

```
double vtkImageMapToWindowLevelColors2::Level  [protected]
```

Referenced by vtkGetMacro(), and vtkSetMacro().

12.395.3.2 Window

```
double vtkImageMapToWindowLevelColors2::Window  [protected]
```

Referenced by vtkGetMacro(), and vtkSetMacro().

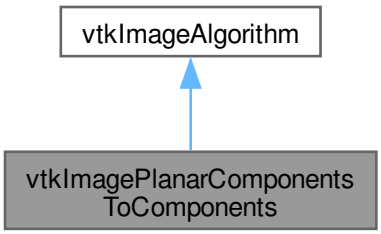
The documentation for this class was generated from the following file:

- vtkImageMapToWindowLevelColors2.h

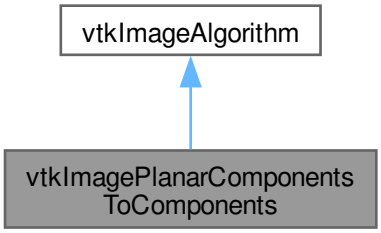
12.396 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkImagePlanarComponentsToComponents, vtkImageAlgorithm)

Static Public Member Functions

- static vtkImagePlanarComponentsToComponents * New ()

Protected Member Functions

- vtkImagePlanarComponentsToComponents ()
- ~vtkImagePlanarComponentsToComponents ()
- virtual int RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)

12.396.1 Constructor & Destructor Documentation

12.396.1.1 vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents () [protected]
```

Referenced by New(), RequestData(), and vtkTypeMacro().

12.396.1.2 ~vtkImagePlanarComponentsToComponents()

```
vtkImagePlanarComponentsToComponents::~vtkImagePlanarComponentsToComponents () [inline], [protected]
```

12.396.2 Member Function Documentation

12.396.2.1 New()

```
vtkImagePlanarComponentsToComponents * vtkImagePlanarComponentsToComponents::New () [static]
```

References vtkImagePlanarComponentsToComponents().

12.396.2.2 PrintSelf()

```
void vtkImagePlanarComponentsToComponents::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.396.2.3 RequestData()

```
virtual int vtkImagePlanarComponentsToComponents::RequestData (
            vtkInformation * ,
            vtkInformationVector ** ,
            vtkInformationVector * ) [protected], [virtual]
```

References vtkImagePlanarComponentsToComponents().

12.396.2.4 vtkTypeMacro()

```
vtkImagePlanarComponentsToComponents::vtkTypeMacro (
            vtkImagePlanarComponentsToComponents ,
            vtkImageAlgorithm )
```

References vtkImagePlanarComponentsToComponents().

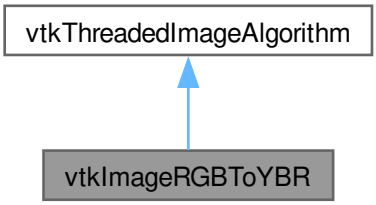
The documentation for this class was generated from the following file:

- vtkImagePlanarComponentsToComponents.h

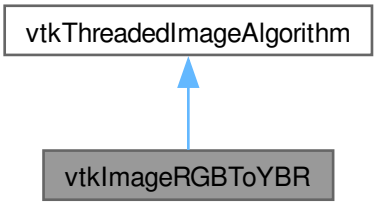
12.397 vtkImageRGBToYBR Class Reference

```
#include <vtkImageRGBToYBR.h>
```

Inheritance diagram for vtkImageRGBToYBR:



Collaboration diagram for vtkImageRGBToYBR:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkImageRGBToYBR, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageRGBToYBR * New ()

Protected Member Functions

- vtkImageRGBToYBR ()
- ∼vtkImageRGBToYBR ()
- void ThreadedExecute (vtkImageData ∗inData, vtkImageData ∗outData, int ext[6], int id)

12.397.1 Constructor & Destructor Documentation

12.397.1.1 vtkImageRGBToYBR()

```
vtkImageRGBToYBR::vtkImageRGBToYBR () [protected]
```

Referenced by New(), ThreadedExecute(), and vtkTypeMacro().

12.397.1.2 ∼vtkImageRGBToYBR()

```
vtkImageRGBToYBR::∼vtkImageRGBToYBR () [inline], [protected]
```

12.397.2 Member Function Documentation

12.397.2.1 New()

```
vtkImageRGBToYBR * vtkImageRGBToYBR::New () [static]
```

References vtkImageRGBToYBR().

12.397.2.2 PrintSelf()

```
void vtkImageRGBToYBR::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.397.2.3 ThreadedExecute()

```
void vtkImageRGBToYBR::ThreadedExecute (
            vtkImageData * inData,
            vtkImageData * outData,
            int ext[6],
            int id) [protected]
```

References vtkImageRGBToYBR().

12.397.2.4 vtkTypeMacro()

```
vtkImageRGBToYBR::vtkTypeMacro (
            vtkImageRGBToYBR ,
            vtkThreadedImageAlgorithm )
```

References vtkImageRGBToYBR().

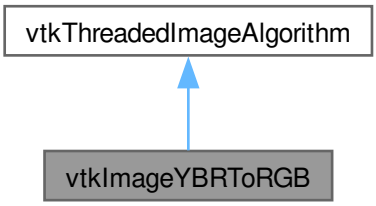
The documentation for this class was generated from the following file:

- vtkImageRGBToYBR.h

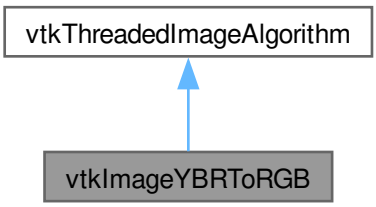
12.398 vtkImageYBRToRGB Class Reference

```
#include <vtkImageYBRToRGB.h>
```

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeMacro (vtkImageYBRToRGB, vtkThreadedImageAlgorithm)

Static Public Member Functions

- static vtkImageYBRToRGB * New ()

Protected Member Functions

- vtkImageYBRToRGB ()
- ~vtkImageYBRToRGB ()
- void ThreadedExecute (vtkImageData *inData, vtkImageData *outData, int ext[6], int id)

12.398.1 Constructor & Destructor Documentation

12.398.1.1 vtkImageYBRToRGB()

```
vtkImageYBRToRGB::vtkImageYBRToRGB () [protected]
```

Referenced by New(), ThreadedExecute(), and vtkTypeMacro().

12.398.1.2 ~vtkImageYBRToRGB()

```
vtkImageYBRToRGB::~vtkImageYBRToRGB () [inline], [protected]
```

12.398.2 Member Function Documentation

12.398.2.1 New()

```
vtkImageYBRToRGB * vtkImageYBRToRGB::New () [static]
```

References vtkImageYBRToRGB().

12.398.2.2 PrintSelf()

```
void vtkImageYBRToRGB::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.398.2.3 ThreadedExecute()

```
void vtkImageYBRToRGB::ThreadedExecute (
            vtkImageData * inData,
            vtkImageData * outData,
            int ext[6],
            int id)  [protected]
```

References vtkImageYBRToRGB().

12.398.2.4 vtkTypeMacro()

```
vtkImageYBRToRGB::vtkTypeMacro (
            vtkImageYBRToRGB ,
            vtkThreadedImageAlgorithm )
```

References vtkImageYBRToRGB().

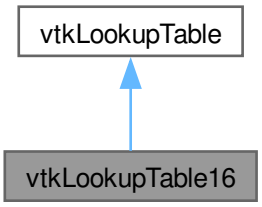
The documentation for this class was generated from the following file:

- vtkImageYBRToRGB.h

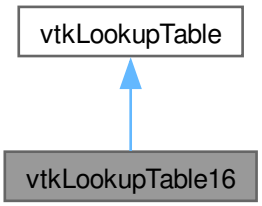
12.399 vtkLookupTable16 Class Reference

```
#include <vtkLookupTable16.h>
```

Inheritance diagram for vtkLookupTable16:



Collaboration diagram for vtkLookupTable16:



Public Member Functions

- void Build ()
- unsigned short ∗ GetPointer (const vtkIdType id)
- void PrintSelf (ostream &os, vtkIndent indent)
- void SetNumberOfTableValues (vtkIdType number)
- vtkTypeMacro (vtkLookupTable16, vtkLookupTable)
- unsigned char ∗ WritePointer (const vtkIdType id, const int number)

Static Public Member Functions

- static vtkLookupTable16 ∗ New ()

Protected Member Functions

- vtkLookupTable16 (int sze=256, int ext=256)
- ∼vtkLookupTable16 ()
- void MapScalarsThroughTable2 (void ∗input, unsigned char ∗output, int inputDataType, int numberOfValues, int inputIncrement, int outputFormat)

Protected Attributes

- vtkUnsignedShortArray ∗ Table16

12.399.1 Constructor & Destructor Documentation

12.399.1.1 vtkLookupTable16()

```
vtkLookupTable16::vtkLookupTable16 (
            int sze = 256,
            int ext = 256)  [protected]
```

Referenced by MapScalarsThroughTable2(), New(), and vtkTypeMacro().

12.399.1.2 ~vtkLookupTable16()

```
vtkLookupTable16::~vtkLookupTable16 () [protected]
```

12.399.2 Member Function Documentation

12.399.2.1 Build()

```
void vtkLookupTable16::Build ()
```

12.399.2.2 GetPointer()

```
unsigned short * vtkLookupTable16::GetPointer (
            const vtkIdType id) [inline]
```

References Table16.

12.399.2.3 MapScalarsThroughTable2()

```
void vtkLookupTable16::MapScalarsThroughTable2 (
            void * input,
            unsigned char * output,
            int inputDataType,
            int numberOfValues,
            int inputIncrement,
            int outputFormat) [protected]
```

References vtkLookupTable16().

12.399.2.4 New()

```
vtkLookupTable16 * vtkLookupTable16::New () [static]
```

References vtkLookupTable16().

12.399.2.5 PrintSelf()

```
void vtkLookupTable16::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.399.2.6 SetNumberOfTableValues()

```
void vtkLookupTable16::SetNumberOfTableValues (
            vtkIdType number)
```

References WritePointer().

12.399.2.7 vtkTypeMacro()

```
vtkLookupTable16::vtkTypeMacro (
            vtkLookupTable16 ,
            vtkLookupTable )
```

References vtkLookupTable16().

12.399.2.8 WritePointer()

```
unsigned char * vtkLookupTable16::WritePointer (
            const vtkIdType id,
            const int number)  [inline]
```

References Table16.

Referenced by SetNumberOfTableValues().

12.399.3 Member Data Documentation

12.399.3.1 Table16

```
vtkUnsignedShortArray* vtkLookupTable16::Table16  [protected]
```

Referenced by GetPointer(), and WritePointer().

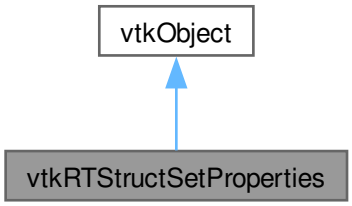
The documentation for this class was generated from the following file:

- vtkLookupTable16.h

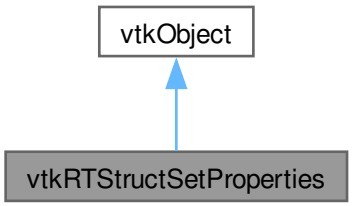
12.400 vtkRTStructSetProperties Class Reference

```
#include <vtkRTStructSetProperties.h>
```

Inheritance diagram for vtkRTStructSetProperties:



Collaboration diagram for vtkRTStructSetProperties:



Public Member Functions

- void AddContourReferencedFrameOfReference (vtkIdType pdnum, const char ∗classuid, const char ∗instanceuid)
- void AddReferencedFrameOfReference (const char ∗classuid, const char ∗instanceuid)
- void AddStructureSetROI (int roinumber, const char ∗reframerefuid, const char ∗roiname, const char ∗ROIGenerationAlgorithm, const char ∗ROIDescription=0)
- void AddStructureSetROIObservation (int refnumber, int observationnumber, const char ∗rtroiinterpretedtype, const char ∗roiinterpreter, const char ∗roiobservationlabel=0)
- virtual void Clear ()
- virtual void DeepCopy (vtkRTStructSetProperties ∗p)
- const char ∗ GetContourReferencedFrameOfReferenceClassUID (vtkIdType pdnum, vtkIdType id)
- const char ∗ GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType pdnum, vtkIdType id)
- vtkIdType GetNumberOfContourReferencedFrameOfReferences ()

- vtkIdType GetNumberOfContourReferencedFrameOfReferences (vtkIdType pdnum)
- vtkIdType GetNumberOfReferencedFrameOfReferences ()
- vtkIdType GetNumberOfStructureSetROIs ()
- const char * GetReferencedFrameOfReferenceClassUID (vtkIdType id)
- const char * GetReferencedFrameOfReferenceInstanceUID (vtkIdType id)
- int GetStructureSetObservationNumber (vtkIdType id)
- const char * GetStructureSetROIDescription (vtkIdType id)
- const char * GetStructureSetROIGenerationAlgorithm (vtkIdType)
- const char * GetStructureSetROIName (vtkIdType)
- int GetStructureSetROINumber (vtkIdType id)
- const char * GetStructureSetROIObservationLabel (vtkIdType id)
- const char * GetStructureSetROIRefFrameRefUID (vtkIdType)
- const char * GetStructureSetRTROIInterpretedType (vtkIdType id)
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetStringMacro (ReferenceFrameOfReferenceUID)
- vtkGetStringMacro (ReferenceSeriesInstanceUID)
- vtkGetStringMacro (SeriesInstanceUID)
- vtkGetStringMacro (SOPInstanceUID)
- vtkGetStringMacro (StructureSetDate)
- vtkGetStringMacro (StructureSetLabel)
- vtkGetStringMacro (StructureSetName)
- vtkGetStringMacro (StructureSetTime)
- vtkGetStringMacro (StudyInstanceUID)
- vtkSetStringMacro (ReferenceFrameOfReferenceUID)
- vtkSetStringMacro (ReferenceSeriesInstanceUID)
- vtkSetStringMacro (SeriesInstanceUID)
- vtkSetStringMacro (SOPInstanceUID)
- vtkSetStringMacro (StructureSetDate)
- vtkSetStringMacro (StructureSetLabel)
- vtkSetStringMacro (StructureSetName)
- vtkSetStringMacro (StructureSetTime)
- vtkSetStringMacro (StudyInstanceUID)
- vtkTypeMacro (vtkRTStructSetProperties, vtkObject)

Static Public Member Functions

- static vtkRTStructSetProperties * New ()

Protected Member Functions

- vtkRTStructSetProperties ()
- ~vtkRTStructSetProperties ()

Protected Attributes

- vtkRTStructSetPropertiesInternals * Internals
- char * ReferenceFrameOfReferenceUID
- char * ReferenceSeriesInstanceUID
- char * SeriesInstanceUID
- char * SOPInstanceUID
- char * StructureSetDate
- char * StructureSetLabel
- char * StructureSetName
- char * StructureSetTime
- char * StudyInstanceUID

12.400.1 Detailed Description

Examples

GenerateRTSTRUCT.cxx.

12.400.2 Constructor & Destructor Documentation

12.400.2.1 vtkRTStructSetProperties()

```
vtkRTStructSetProperties::vtkRTStructSetProperties () [protected]
```

Referenced by DeepCopy(), New(), and vtkTypeMacro().

12.400.2.2 ~vtkRTStructSetProperties()

```
vtkRTStructSetProperties::~vtkRTStructSetProperties () [protected]
```

12.400.3 Member Function Documentation

12.400.3.1 AddContourReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (
            vtkIdType pdnum,
            const char * classuid,
            const char * instanceuid)
```

12.400.3.2 AddReferencedFrameOfReference()

```
void vtkRTStructSetProperties::AddReferencedFrameOfReference (
            const char * classuid,
            const char * instanceuid)
```

12.400.3.3 AddStructureSetROI()

```
void vtkRTStructSetProperties::AddStructureSetROI (
            int roinumber,
            const char * reframerefuid,
            const char * roiname,
            const char * ROIGenerationAlgorithm,
            const char * ROIDescription = 0)
```

12.400.3.4 AddStructureSetROIObservation()

```
void vtkRTStructSetProperties::AddStructureSetROIObservation (
            int refnumber,
            int observationnumber,
            const char * rtroiinterpretedtype,
            const char * roiinterpreter,
            const char * roiobservationlabel = 0)
```

12.400.3.5 Clear()

```
virtual void vtkRTStructSetProperties::Clear ()  [virtual]
```

12.400.3.6 DeepCopy()

```
virtual void vtkRTStructSetProperties::DeepCopy (
            vtkRTStructSetProperties * p)  [virtual]
```

References vtkRTStructSetProperties().

12.400.3.7 GetContourReferencedFrameOfReferenceClassUID()

```
const char * vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (
            vtkIdType pdnum,
            vtkIdType id)
```

12.400.3.8 GetContourReferencedFrameOfReferenceInstanceUID()

```
const char * vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (
            vtkIdType pdnum,
            vtkIdType id)
```

12.400.3.9 GetNumberOfContourReferencedFrameOfReferences() [1/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ()
```

12.400.3.10 GetNumberOfContourReferencedFrameOfReferences() [2/2]

```
vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (
            vtkIdType pdnum)
```

12.400.3.11 GetNumberOfReferencedFrameOfReferences()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ()
```

12.400.3.12 GetNumberOfStructureSetROIs()

```
vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ()
```

12.400.3.13 GetReferencedFrameOfReferenceClassUID()

```
const char * vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (
            vtkIdType id)
```

12.400.3.14 GetReferencedFrameOfReferenceInstanceUID()

```
const char * vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (
            vtkIdType id)
```

12.400.3.15 GetStructureSetObservationNumber()

```
int vtkRTStructSetProperties::GetStructureSetObservationNumber (
            vtkIdType id)
```

12.400.3.16 GetStructureSetROIDescription()

```
const char * vtkRTStructSetProperties::GetStructureSetROIDescription (
            vtkIdType id)
```

12.400.3.17 GetStructureSetROIGenerationAlgorithm()

```
const char * vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (
            vtkIdType )
```

12.400.3.18 GetStructureSetROIName()

```
const char * vtkRTStructSetProperties::GetStructureSetROIName (
            vtkIdType )
```

12.400.3.19 GetStructureSetROINumber()

```
int vtkRTStructSetProperties::GetStructureSetROINumber (
            vtkIdType id)
```

12.400.3.20 GetStructureSetROIObservationLabel()

```
const char * vtkRTStructSetProperties::GetStructureSetROIObservationLabel (
            vtkIdType id)
```

12.400.3.21 GetStructureSetROIRefFrameRefUID()

```
const char * vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (
            vtkIdType )
```

12.400.3.22 GetStructureSetRTROIInterpretedType()

```
const char * vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (
            vtkIdType id)
```

12.400.3.23 New()

```
vtkRTStructSetProperties * vtkRTStructSetProperties::New ()  [static]
```

Examples

GenerateRTSTRUCT.cxx.

References vtkRTStructSetProperties().

12.400.3.24 PrintSelf()

```
void vtkRTStructSetProperties::PrintSelf (
            ostream & os,
            vtkIndent indent)
```

12.400.3.25 vtkGetStringMacro() [1/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            ReferenceFrameOfReferenceUID )
```

References ReferenceFrameOfReferenceUID.

12.400.3.26 vtkGetStringMacro() [2/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            ReferenceSeriesInstanceUID )
```

References ReferenceSeriesInstanceUID.

12.400.3.27 vtkGetStringMacro() [3/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            SeriesInstanceUID )
```

References SeriesInstanceUID.

12.400.3.28 vtkGetStringMacro() [4/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            SOPInstanceUID )
```

References SOPInstanceUID.

12.400.3.29 vtkGetStringMacro() [5/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetDate )
```

References StructureSetDate.

12.400.3.30 vtkGetStringMacro() [6/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetLabel )
```

References StructureSetLabel.

12.400.3.31 vtkGetStringMacro() [7/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetName )
```

References StructureSetName.

12.400.3.32 vtkGetStringMacro() [8/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StructureSetTime )
```

References StructureSetTime.

12.400.3.33 vtkGetStringMacro() [9/9]

```
vtkRTStructSetProperties::vtkGetStringMacro (
            StudyInstanceUID )
```

References StudyInstanceUID.

12.400.3.34 vtkSetStringMacro() [1/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            ReferenceFrameOfReferenceUID )
```

References ReferenceFrameOfReferenceUID.

12.400.3.35 vtkSetStringMacro() [2/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            ReferenceSeriesInstanceUID )
```

References ReferenceSeriesInstanceUID.

12.400.3.36 vtkSetStringMacro() [3/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            SeriesInstanceUID )
```

References SeriesInstanceUID.

12.400.3.37 vtkSetStringMacro() [4/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            SOPInstanceUID )
```

References SOPInstanceUID.

12.400.3.38 vtkSetStringMacro() [5/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetDate )
```

References StructureSetDate.

12.400.3.39 vtkSetStringMacro() [6/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetLabel )
```

References StructureSetLabel.

12.400.3.40 vtkSetStringMacro() [7/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetName )
```

References StructureSetName.

12.400.3.41 vtkSetStringMacro() [8/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StructureSetTime )
```

References StructureSetTime.

12.400.3.42 vtkSetStringMacro() [9/9]

```
vtkRTStructSetProperties::vtkSetStringMacro (
            StudyInstanceUID )
```

References StudyInstanceUID.

12.400.3.43 vtkTypeMacro()

```
vtkRTStructSetProperties::vtkTypeMacro (
            vtkRTStructSetProperties ,
            vtkObject )
```

References vtkRTStructSetProperties().

12.400.4 Member Data Documentation

12.400.4.1 Internals

```
vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals  [protected]
```

12.400.4.2 ReferenceFrameOfReferenceUID

```
char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID  [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.400.4.3 ReferenceSeriesInstanceUID

```
char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID  [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.400.4.4 SeriesInstanceUID

```
char* vtkRTStructSetProperties::SeriesInstanceUID  [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.400.4.5 SOPInstanceUID

```
char* vtkRTStructSetProperties::SOPInstanceUID  [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.400.4.6 StructureSetDate

```
char* vtkRTStructSetProperties::StructureSetDate  [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.400.4.7 StructureSetLabel

```
char* vtkRTStructSetProperties::StructureSetLabel [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.400.4.8 StructureSetName

```
char* vtkRTStructSetProperties::StructureSetName [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.400.4.9 StructureSetTime

```
char* vtkRTStructSetProperties::StructureSetTime [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

12.400.4.10 StudyInstanceUID

```
char* vtkRTStructSetProperties::StudyInstanceUID [protected]
```

Referenced by vtkGetStringMacro(), and vtkSetStringMacro().

The documentation for this class was generated from the following file:

- vtkRTStructSetProperties.h

12.401 gdcm::Waveform Class Reference

Waveform class.

```
#include <gdcmWaveform.h>
```

Public Member Functions

- Waveform ()=default

12.401.1 Detailed Description

Waveform class.

12.401.2 Constructor & Destructor Documentation

12.401.2.1 Waveform()

```
gdcm::Waveform::Waveform () [default]
```

The documentation for this class was generated from the following file:

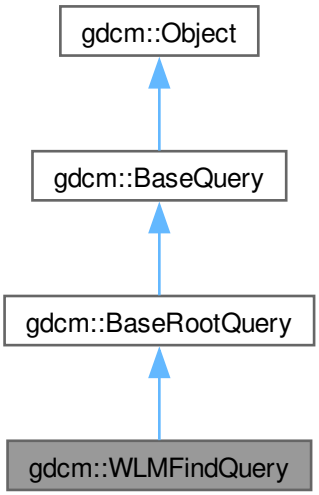
- gdcmWaveform.h

12.402 gdcm::WLMFindQuery Class Reference

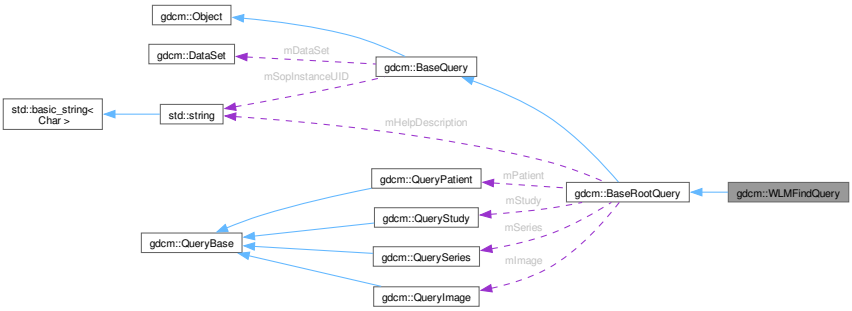
PatientRootQuery.

```
#include <gdcmWLMFindQuery.h>
```

Inheritance diagram for gdcm::WLMFindQuery:



Collaboration diagram for gdcm::WLMFindQuery:



Public Member Functions

- WLMFindQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const override
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel) override
- void InitializeDataSet (const EQueryLevel &inQueryLevel) override
- bool ValidateQuery (bool inStrict=true) const override

Public Member Functions inherited from gdcm::BaseRootQuery

- ~BaseRootQuery () override=default
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)

Public Member Functions inherited from gdcm::BaseQuery

- ~BaseQuery () override
- void AddQueryDataSet (const DataSet &ds)
- DataSet & GetQueryDataSet ()
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- std::string GetSOPInstanceUID () const
- void Print (std::ostream &os) const override
- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)
- void SetSOPInstanceUID (const std::string &iSopInstanceUID)
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

Public Member Functions inherited from gdcm::Object

- Object ()
- Object (const Object &)

  Special requirement for copy/cstor, assignment operator.
- virtual ~Object ()
- void operator= (const Object &)

Protected Member Functions

- DataSet GetValidDataSet () const

Protected Member Functions inherited from gdcm::BaseRootQuery

- BaseRootQuery ()

Protected Member Functions inherited from gdcm::BaseQuery

- BaseQuery ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)
- bool ValidDataSet (const DataSet &dataSetToValid, const DataSet &dataSetReference) const

Protected Member Functions inherited from gdcm::Object

- void Register ()
- void UnRegister ()

Friends

- class QueryFactory

Additional Inherited Members

Static Public Member Functions inherited from gdcm::BaseRootQuery

- static QueryBase * Construct (ERootType inRootType, EQueryLevel qlevel)
- static int GetQueryLevelFromString (const char *str)
- static const char * GetQueryLevelString (EQueryLevel ql)

Protected Attributes inherited from gdcm::BaseRootQuery

- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Protected Attributes inherited from gdcm::BaseQuery

- DataSet mDataSet
- std::string mSopInstanceUID

12.402.1 Detailed Description

PatientRootQuery.

contains: the class which will produce a dataset for c-find with patient root

12.402.2 Constructor & Destructor Documentation

12.402.2.1 WLMFindQuery()

```
gdcm::WLMFindQuery::WLMFindQuery ()
```

12.402.3 Member Function Documentation

12.402.3.1 GetAbstractSyntaxUID()

```
UIDs::TSName gdcm::WLMFindQuery::GetAbstractSyntaxUID () const  [override], [virtual]
```

Implements gdcm::BaseQuery.

12.402.3.2 GetTagListByLevel()

```
std::vector< Tag > gdcm::WLMFindQuery::GetTagListByLevel (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function will return all tags at a given query level, so that they maybe selected for searching. The boolean forFind is true if the query is a find query, or false for a move query.

Implements gdcm::BaseRootQuery.

12.402.3.3 GetValidDataSet()

```
DataSet gdcm::WLMFindQuery::GetValidDataSet () const  [protected]
```

12.402.3.4 InitializeDataSet()

```
void gdcm::WLMFindQuery::InitializeDataSet (
            const EQueryLevel & inQueryLevel)  [override], [virtual]
```

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmtk

Implements gdcm::BaseRootQuery.

12.402.3.5 ValidateQuery()

```
bool gdcm::WLMFindQuery::ValidateQuery (
            bool inStrict = true) const  [override], [virtual]
```

have to be able to ensure that 0x8,0x52 is set (which will be true if InitializeDataSet is called...) that the level is appropriate (ie, not setting PATIENT for a study query that the tags in the query match the right level (either required, unique, optional) by default, this function checks to see if the query is for finding, which is more permissive than for moving. For moving, only the unique tags are allowed. 10 Jan 2011: adding in the 'strict' mode. according to the standard (at least, how I've read it), only tags for a particular level should be allowed in a particular query (ie, just series level tags in a series level query). However, it seems that dcm4chee doesn't share that interpretation. So, if 'inStrict' is false, then tags from the current level and all higher levels are now considered valid. So, if you're doing a non-strict series-level query, tags from the patient and study level can be passed along as well.

Implements gdcm::BaseRootQuery.

12.402.4 Friends And Related Symbol Documentation

12.402.4.1 QueryFactory

```
friend class QueryFactory  [friend]
```

References QueryFactory.

Referenced by QueryFactory.

The documentation for this class was generated from the following file:

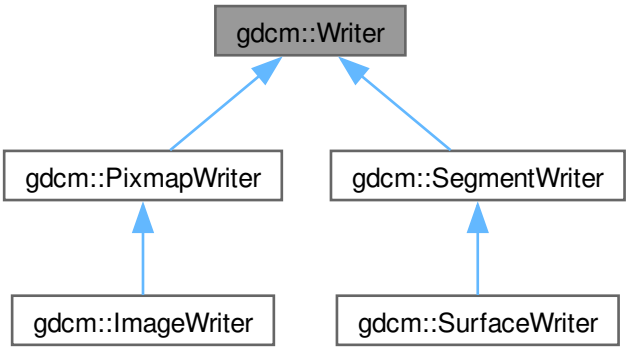
- gdcmWLMFindQuery.h

12.403 gdcm::Writer Class Reference

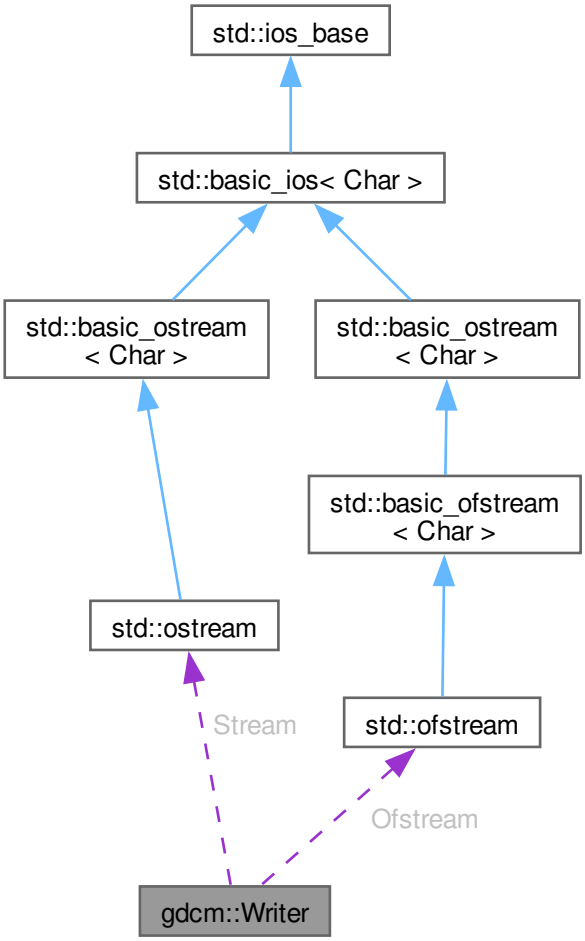
Writer ala DOM (Document Object Model).

```
#include <gdcmWriter.h>
```

Inheritance diagram for gdcm::Writer:



Collaboration diagram for gdcm::Writer:



Public Member Functions

- Writer ()
- virtual ~Writer ()
- void CheckFileMetaInformationOff ()
- void CheckFileMetaInformationOn ()
- File & GetFile ()
- void SetCheckFileMetaInformation (bool b)

  Undocumented function, do not use (= leave default).
- void SetFile (const File &f)

  Set/Get the DICOM file (DataSet + Header).
- void SetFileName (const char *filename_native)

Set the filename of DICOM file to write:

- void SetStream (std::ostream &output_stream)

  Set user ostream buffer.

- virtual bool Write ()

  Main function to tell the writer to write.

Protected Member Functions

- bool GetCheckFileMetaInformation () const
- std::ostream * GetStreamPtr () const
- void SetWriteDataSetOnly (bool b)

Protected Attributes

- std::ofstream * Ofstream
- std::ostream * Stream

Friends

- class StreamImageWriter

12.403.1 Detailed Description

Writer ala DOM (Document Object Model).

This class is a non-validating writer, it will only performs well- formedness check only.

Detailed description here To avoid GDCM being yet another broken DICOM lib we try to be user level and avoid writing illegal stuff (odd length, non-zero value for Item start/end length ...) Therefore you cannot (well unless you are really smart) write DICOM with even length tag. All the checks are consider basics:

- Correct Meta Information Header (see gdcm::FileMetaInformation)
- Zero value for Item Length (0xfffe, 0xe00d/0xe0dd)
- Even length for any elements
- Alphabetical order for elements (guaranteed by design of internals)
- 32bits VR will be rewritten with 00

Warning

gdcm::Writer cannot write a DataSet if no SOP Instance UID (0008,0018) is found, unless a DICOMDIR is being written out

See also

Reader DataSet File

Examples

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GenerateDICOMDIR.cs, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReformatFile.cs, StreamImageReaderTest.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.403.2 Constructor & Destructor Documentation

12.403.2.1 Writer()

```
gdcm::Writer::Writer ()
```

12.403.2.2 ~Writer()

```
virtual gdcm::Writer::~Writer ()  [virtual]
```

12.403.3 Member Function Documentation

12.403.3.1 CheckFileMetaInformationOff()

```
void gdcm::Writer::CheckFileMetaInformationOff ()  [inline]
```

Examples

CreateFakeRTDOSE.cxx, FixBrokenJ2K.cxx, and HelloWorld.cxx.

12.403.3.2 CheckFileMetaInformationOn()

```
void gdcm::Writer::CheckFileMetaInformationOn ()  [inline]
```

12.403.3.3 GetCheckFileMetaInformation()

```
bool gdcm::Writer::GetCheckFileMetaInformation () const  [inline], [protected]
```

12.403.3.4 GetFile()

```
File & gdcm::Writer::GetFile ()  [inline]
```

Examples

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, MpegVideoInfo.cs, QIDO-RS.cxx, StreamImageReaderTest.cxx, TemplateEmptyImage.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.403.3.5 GetStreamPtr()

```
std::ostream * gdcm::Writer::GetStreamPtr () const  [inline], [protected]
```

References Stream.

12.403.3.6 SetCheckFileMetaInformation()

```
void gdcm::Writer::SetCheckFileMetaInformation (
            bool b)  [inline]
```

Undocumented function, do not use (= leave default).

Examples

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and PatchFile.cxx.

12.403.3.7 SetFile()

```
void gdcm::Writer::SetFile (
            const File & f)  [inline]
```

Set/Get the DICOM file (DataSet + Header).

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, Cleaner.cs, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateFakeRTDOSE.cxx, DecompressImage.cs, DeriveSeries.cxx, DuplicatePCDE.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GenerateDICOMDIR.cs, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, NewSequence.cs, PatchFile.cxx, ReformatFile.cs, StandardizeFiles.cs, pmsct_rgb1.cxx, and rle2img.cxx.

12.403.3.8 SetFileName()

```
void gdcm::Writer::SetFileName (
            const char * filename_native)
```

Set the filename of DICOM file to write:

Examples

BasicAnonymizer.cs, BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, Cleaner.cs, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CompressImage.cxx, CompressLossyJPEG.cs, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DecompressImage.cs, DeriveSeries.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GenerateDICOMDIR.cs, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, MergeTwoFiles.cxx, MpegVideoInfo.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReformatFile.cs, StandardizeFiles.cs, TemplateEmptyImage.cxx, csa2img.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

12.403.3.9 SetStream()

```
void gdcm::Writer::SetStream (
            std::ostream & output_stream)  [inline]
```

Set user ostream buffer.

References Stream.

12.403.3.10 SetWriteDataSetOnly()

```
void gdcm::Writer::SetWriteDataSetOnly (
            bool b)  [inline], [protected]
```

12.403.3.11 Write()

```
virtual bool gdcm::Writer::Write ()  [virtual]
```

Main function to tell the writer to write.

Reimplemented in gdcm::ImageWriter, gdcm::PixmapWriter, gdcm::SegmentWriter, and gdcm::SurfaceWriter.

Examples

BasicAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, Cleaner.cs, ClinicalTrialAnnotate.cxx, ClinicalTrialIdentificationWorkflow.cs, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DeriveSeries.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, FixOrientation.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GenerateDICOMDIR.cs, HelloWorld.cxx, LargeVRDSExplicit.cxx, MakeTemplate.cxx, ManipulateFile.cs, NewSequence.cs, PatchFile.cxx, QIDO-RS.cxx, ReformatFile.cs, pmsct_rgb1.cxx, and rle2img.cxx.

12.403.4 Friends And Related Symbol Documentation

12.403.4.1 StreamImageWriter

```
friend class StreamImageWriter  [friend]
```

References StreamImageWriter.

Referenced by StreamImageWriter.

12.403.5 Member Data Documentation

12.403.5.1 Ofstream

```
std::ofstream* gdcm::Writer::Ofstream  [protected]
```

12.403.5.2 Stream

```
std::ostream* gdcm::Writer::Stream [protected]
```

Referenced by GetStreamPtr(), and SetStream().

The documentation for this class was generated from the following file:

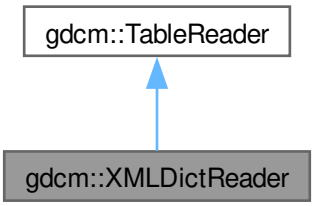
- gdcmWriter.h

12.404 gdcm::XMLDictReader Class Reference

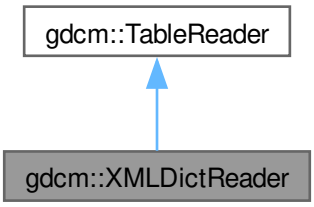
Class for representing a XMLDictReader.

```
#include <gdcmXMLDictReader.h>
```

Inheritance diagram for gdcm::XMLDictReader:



Collaboration diagram for gdcm::XMLDictReader:



Public Member Functions

- XMLDictReader ()
- ~XMLDictReader ()
- void CharacterDataHandler (const char *data, int length)
- void EndElement (const char *name)
- const Dict & GetDict ()
- void StartElement (const char *name, const char **atts)

Public Member Functions inherited from gdcm::TableReader

- TableReader (Defs &defs)
- virtual ~TableReader ()=default
- const Defs & GetDefs () const
- const char * GetFilename ()
- void HandleIOD (const char **atts)
- void HandleIODEntry (const char **atts)
- void HandleMacro (const char **atts)
- void HandleMacroEntry (const char **atts)
- void HandleMacroEntryDescription (const char **atts)
- void HandleModule (const char **atts)
- void HandleModuleEntry (const char **atts)
- void HandleModuleEntryDescription (const char **atts)
- void HandleModuleInclude (const char **atts)
- int Read ()
- void SetFilename (const char *filename)

Protected Member Functions

- void HandleDescription (const char **atts)
- void HandleEntry (const char **atts)

12.404.1 Detailed Description

Class for representing a XMLDictReader.

Note

bla Will read the DICOMV3.xml file

12.404.2 Constructor & Destructor Documentation

12.404.2.1 XMLDictReader()

```
gdcm::XMLDictReader::XMLDictReader ()
```

12.404.2.2 ~XMLDictReader()

```
gdcm::XMLDictReader::~XMLDictReader () [inline]
```

12.404.3 Member Function Documentation

12.404.3.1 CharacterDataHandler()

```
void gdcm::XMLDictReader::CharacterDataHandler (
            const char * data,
            int length) [virtual]
```

Reimplemented from gdcm::TableReader.

12.404.3.2 EndElement()

```
void gdcm::XMLDictReader::EndElement (
            const char * name) [virtual]
```

Reimplemented from gdcm::TableReader.

12.404.3.3 GetDict()

```
const Dict & gdcm::XMLDictReader::GetDict () [inline]
```

12.404.3.4 HandleDescription()

```
void gdcm::XMLDictReader::HandleDescription (
            const char ** atts) [protected]
```

12.404.3.5 HandleEntry()

```
void gdcm::XMLDictReader::HandleEntry (
            const char ** atts) [protected]
```

12.404.3.6 StartElement()

```
void gdcm::XMLDictReader::StartElement (
            const char * name,
            const char ** atts) [virtual]
```

Reimplemented from gdcm::TableReader.

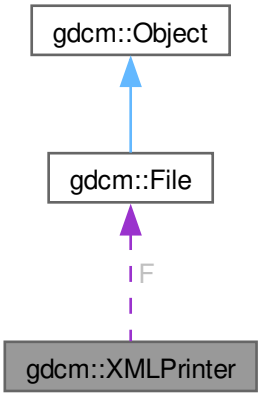
The documentation for this class was generated from the following file:

- gdcmXMLDictReader.h

12.405 gdcm::XMLPrinter Class Reference

```
#include <gdcmXMLPrinter.h>
```

Collaboration diagram for gdcm::XMLPrinter:



Public Types

- enum PrintStyles {
  OnlyUUID = 0 ,
  LOADBULKDATA = 1 }

Public Member Functions

- XMLPrinter ()
- virtual ~XMLPrinter ()
- PrintStyles GetPrintStyle () const
- virtual void HandleBulkData (const char *uuid, const TransferSyntax &ts, const char *bulkdata, size_t bulklen)
- void Print (std::ostream &os)
- void PrintDataSet (const DataSet &ds, const TransferSyntax &ts, std::ostream &os)
- void SetFile (File const &f)
- void SetStyle (PrintStyles ps)

Protected Member Functions

- VR PrintDataElement (std::ostream &os, const Dicts &dicts, const DataSet &ds, const DataElement &de, const TransferSyntax &ts)
- void PrintSQ (const SequenceOfItems *sqi, const TransferSyntax &ts, std::ostream &os)

Protected Attributes

- const File * F
- PrintStyles PrintStyle

12.405.1 Member Enumeration Documentation

12.405.1.1 PrintStyles

```
enum gdcm::XMLPrinter::PrintStyles
```

Enumerator

OnlyUUID	
LOADBULKDATA	

12.405.2 Constructor & Destructor Documentation

12.405.2.1 XMLPrinter()

```
gdcm::XMLPrinter::XMLPrinter ()
```

12.405.2.2 ~XMLPrinter()

```
virtual gdcm::XMLPrinter::~XMLPrinter ()  [virtual]
```

12.405.3 Member Function Documentation

12.405.3.1 GetPrintStyle()

```
PrintStyles gdcm::XMLPrinter::GetPrintStyle () const  [inline]
```

References PrintStyle.

12.405.3.2 HandleBulkData()

```
virtual void gdcm::XMLPrinter::HandleBulkData (
            const char * uuid,
            const TransferSyntax & ts,
            const char * bulkdata,
            size_t bulklen)  [virtual]
```

Virtual function mechanism to allow application programmer to override the default mechanism for BulkData handling. By default GDCM will simply discard the BulkData and only write the UUID

12.405.3.3 Print()

```
void gdcm::XMLPrinter::Print (
            std::ostream & os)
```

12.405.3.4 PrintDataElement()

```
VR gdcm::XMLPrinter::PrintDataElement (
            std::ostream & os,
            const Dicts & dicts,
            const DataSet & ds,
            const DataElement & de,
            const TransferSyntax & ts)  [protected]
```

12.405.3.5 PrintDataSet()

```
void gdcm::XMLPrinter::PrintDataSet (
            const DataSet & ds,
            const TransferSyntax & ts,
            std::ostream & os)
```

12.405.3.6 PrintSQ()

```
void gdcm::XMLPrinter::PrintSQ (
            const SequenceOfItems * sqi,
            const TransferSyntax & ts,
            std::ostream & os)  [protected]
```

12.405.3.7 SetFile()

```
void gdcm::XMLPrinter::SetFile (
            File const & f)  [inline]
```

References F.

12.405.3.8 SetStyle()

```
void gdcm::XMLPrinter::SetStyle (
            PrintStyles ps)  [inline]
```

References PrintStyle.

12.405.4 Member Data Documentation

12.405.4.1 F

```
const File* gdcm::XMLPrinter::F  [protected]
```

Referenced by SetFile().

12.405.4.2 PrintStyle

```
PrintStyles gdcm::XMLPrinter::PrintStyle  [protected]
```

Referenced by GetPrintStyle(), and SetStyle().

The documentation for this class was generated from the following file:

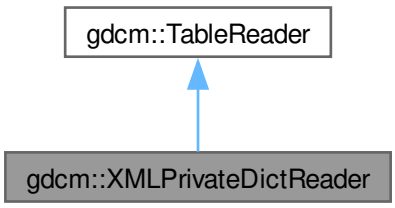
- gdcmXMLPrinter.h

12.406 gdcm::XMLPrivateDictReader Class Reference

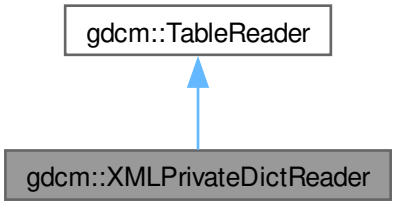
Class for representing a XMLPrivateDictReader.

```
#include <gdcmXMLPrivateDictReader.h>
```

Inheritance diagram for gdcm::XMLPrivateDictReader:



Collaboration diagram for gdcm::XMLPrivateDictReader:



Public Member Functions

- XMLPrivateDictReader ()
- ~XMLPrivateDictReader ()
- void CharacterDataHandler (const char *data, int length)
- void EndElement (const char *name)
- const PrivateDict & GetPrivateDict ()
- void StartElement (const char *name, const char **atts)

Public Member Functions inherited from gdcm::TableReader

- TableReader (Defs &defs)
- virtual ~TableReader ()=default
- const Defs & GetDefs () const
- const char * GetFilename ()
- void HandleIOD (const char **atts)
- void HandleIODEntry (const char **atts)
- void HandleMacro (const char **atts)
- void HandleMacroEntry (const char **atts)
- void HandleMacroEntryDescription (const char **atts)
- void HandleModule (const char **atts)
- void HandleModuleEntry (const char **atts)
- void HandleModuleEntryDescription (const char **atts)
- void HandleModuleInclude (const char **atts)
- int Read ()
- void SetFilename (const char *filename)

Protected Member Functions

- void HandleDescription (const char **atts)
- void HandleEntry (const char **atts)

12.406.1 Detailed Description

Class for representing a XMLPrivateDictReader.

Note

bla Will read the Private.xml file

12.406.2 Constructor & Destructor Documentation

12.406.2.1 XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::XMLPrivateDictReader ()
```

12.406.2.2 ~XMLPrivateDictReader()

```
gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ()  [inline]
```

12.406.3 Member Function Documentation

12.406.3.1 CharacterDataHandler()

```
void gdcm::XMLPrivateDictReader::CharacterDataHandler (
            const char * data,
            int length)  [virtual]
```

Reimplemented from gdcm::TableReader.

12.406.3.2 EndElement()

```
void gdcm::XMLPrivateDictReader::EndElement (
            const char * name)  [virtual]
```

Reimplemented from gdcm::TableReader.

12.406.3.3 GetPrivateDict()

```
const PrivateDict & gdcm::XMLPrivateDictReader::GetPrivateDict ()  [inline]
```

12.406.3.4 HandleDescription()

```
void gdcm::XMLPrivateDictReader::HandleDescription (
            const char ** atts)  [protected]
```

12.406.3.5 HandleEntry()

```
void gdcm::XMLPrivateDictReader::HandleEntry (
            const char ** atts)  [protected]
```

12.406.3.6 StartElement()

```
void gdcm::XMLPrivateDictReader::StartElement (
            const char * name,
            const char ** atts)  [virtual]
```

Reimplemented from gdcm::TableReader.

The documentation for this class was generated from the following file:

- gdcmXMLPrivateDictReader.h

Chapter 13

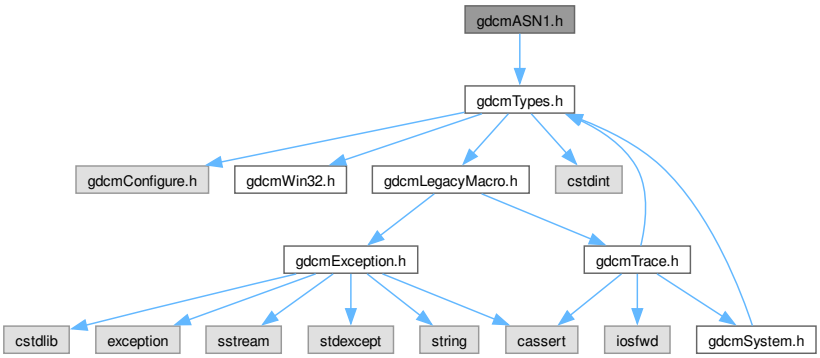
File Documentation

13.1 README.txt File Reference

13.2 TestsList.txt File Reference

13.3 gdcmASN1.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmASN1.h:



Classes

- class gdcm::ASN1

  Class for ASN1.

Namespaces

- namespace gdcm

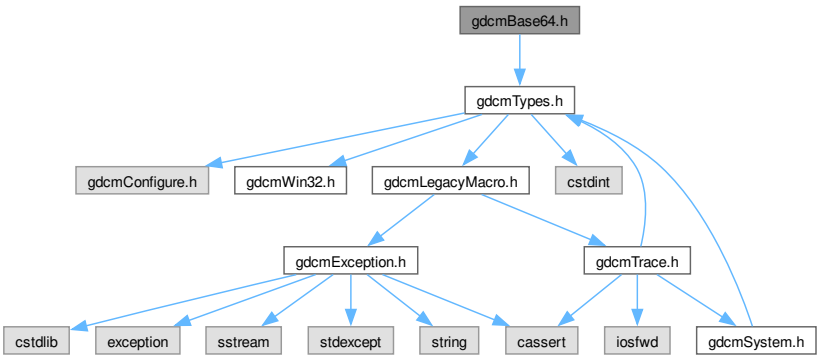
13.4 gdcmASN1.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMASN1_H
#define GDCMASN1_H

#include "gdcmTypes.h"


namespace gdcm
{
//-----------------------------------------------------------------------------
class ASN1Internals;
class GDCM_EXPORT ASN1
{
public :
  ASN1();
  ~ASN1();

  static bool ParseDumpFile(const char *filename);

  static bool ParseDump(const char *array, size_t length);

  ASN1(const ASN1&) = delete;
  void operator=(const ASN1&) = delete;
protected:
  int TestPBKDF2();

private:
  ASN1Internals *Internals;
};
} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMASN1_H
```

13.5 gdcmBase64.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmBase64.h:



Classes

- class gdcm::Base64

  Class for Base64.

Namespaces

- namespace gdcm

13.6 gdcmBase64.h

Go to the documentation of this file.

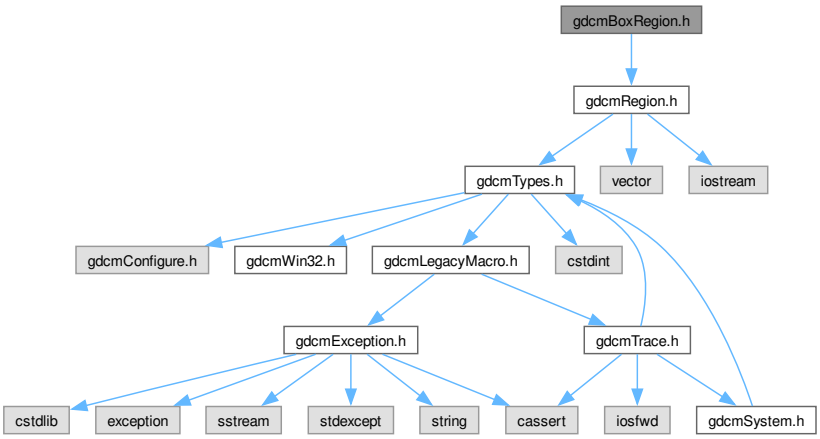
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMBASE64_H
#define GDCMBASE64_H

#include "gdcmTypes.h"

namespace gdcm
{
class GDCM_EXPORT Base64
```

```
{
public:

  static size_t GetEncodeLength(const char *src, size_t srclen );

  static size_t Encode( char *dst, size_t dlen, const char *src, size_t slen );

  static size_t GetDecodeLength( const char *src, size_t len );

  static size_t Decode( char *dst, size_t dlen, const char *src, size_t slen );

  Base64(const Base64&) = delete;
  void operator=(const Base64&) = delete;
};

} // end namespace gdcm

#endif // GDCMBASE64_H
```

13.7 gdcmBoxRegion.h File Reference

`#include "gdcmRegion.h"`

Include dependency graph for gdcmBoxRegion.h:



Classes

- class gdcm::BoxRegion

  Class for manipulation box region.

Namespaces

- namespace gdcm

13.8 gdcmBoxRegion.h

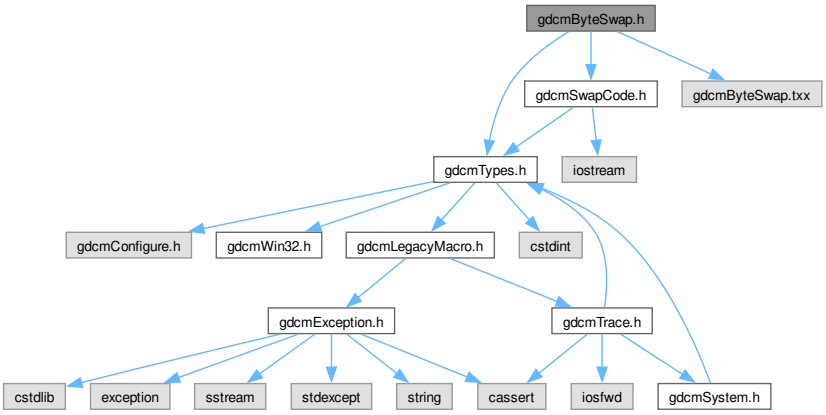
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMBOXREGION_H
#define GDCMBOXREGION_H

#include "gdcmRegion.h"

namespace gdcm
{
class BoxRegionInternals;
//-----------------------------------------------------------------------------
class GDCM_EXPORT BoxRegion : public Region
{
public :
  BoxRegion();
  ~BoxRegion() override;

  void SetDomain(unsigned int xmin, unsigned int xmax,
    unsigned int ymin, unsigned int ymax,
    unsigned int zmin, unsigned int zmax);

  unsigned int GetXMin() const;
  unsigned int GetXMax() const;
  unsigned int GetYMin() const;
  unsigned int GetYMax() const;
  unsigned int GetZMin() const;
  unsigned int GetZMax() const;

  // Satisfy pure virtual parent class
  Region *Clone() const override;
  bool Empty() const override;
  bool IsValid() const override;
  size_t Area() const override;
  BoxRegion ComputeBoundingBox() override;

  void Print(std::ostream &os = std::cout) const override;

  static BoxRegion BoundingBox(BoxRegion const & b1, BoxRegion const & b2 );

  BoxRegion(const BoxRegion&);
  void operator=(const BoxRegion&);
private:
  BoxRegionInternals *Internals;
};

} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMREGION_H
```

13.9 gdcmByteSwap.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
```

`#include "gdcmByteSwap.txx"`
Include dependency graph for gdcmByteSwap.h:



Classes

- class gdcm::ByteSwap< T >

  ByteSwap.

Namespaces

- namespace gdcm

13.10 gdcmByteSwap.h

Go to the documentation of this file.

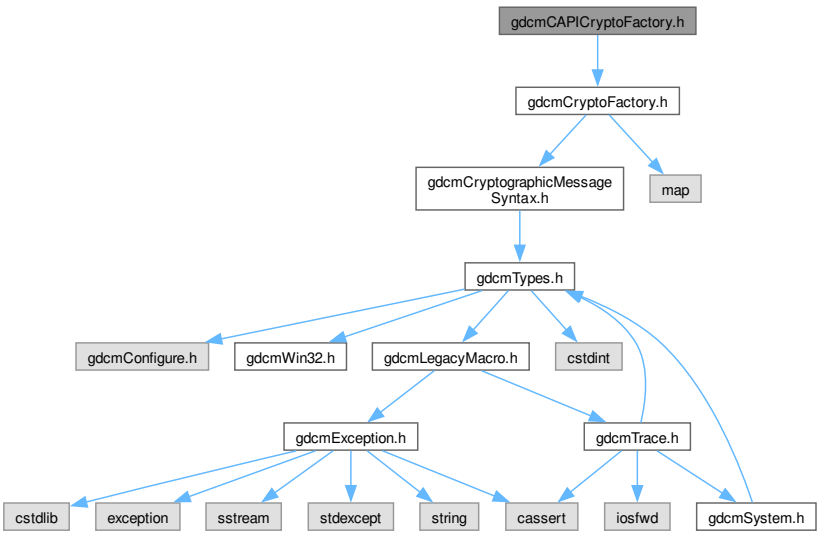
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMBYTESWAP_H
#define GDCMBYTESWAP_H

#include "gdcmTypes.h"
#include "gdcmSwapCode.h"

namespace gdcm
{

template<class T>
```

```
00030 class ByteSwap
00031 {
00032 public:
00034   static bool SystemIsBigEndian ();
00035   static bool SystemIsLittleEndian ();
00036
00037   static void Swap(T &p);
00038   static void SwapFromSwapCodeIntoSystem(T &p, SwapCode const &sc);
00039   static void SwapRange(T *p, unsigned int num);
00040   static void SwapRangeFromSwapCodeIntoSystem(T *p, SwapCode const &sc,
00041     std::streamoff num);
00042
00043 protected:
00044 //  ByteSwap() {}
00045 //  ~ByteSwap() {}
00046
00047 private:
00048
00049 };
00050
00055
00056 } // end namespace gdcm
00057
00058 #include "gdcmByteSwap.txx"
00059
00060 #endif //GDCMBYTESWAP_H
```

13.11 gdcmCAPICryptoFactory.h File Reference

`#include "gdcmCryptoFactory.h"`

Include dependency graph for gdcmCAPICryptoFactory.h:



Classes

- class gdcm::CAPICryptoFactory

Namespaces

- namespace gdcm

13.12 gdcmCAPICryptoFactory.h

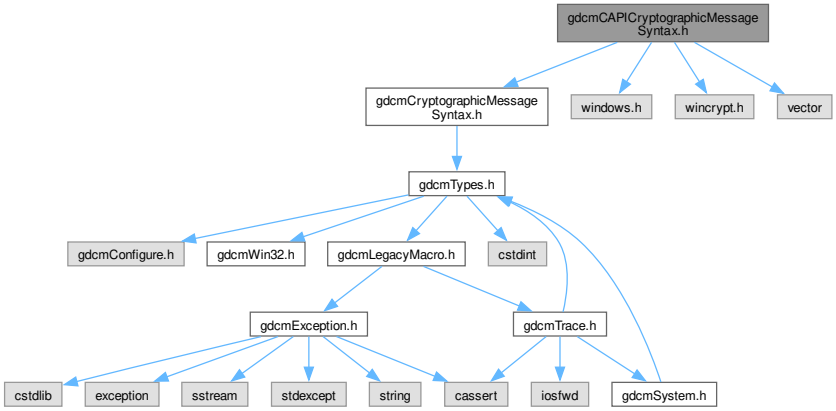
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCAPICRYPTOFACTORY_H
#define GDCMCAPICRYPTOFACTORY_H

#include "gdcmCryptoFactory.h"

namespace gdcm
{

class GDCM_EXPORT CAPICryptoFactory : public CryptoFactory
{
public:
  CAPICryptoFactory(CryptoLib id);
  CryptographicMessageSyntax* CreateCMSProvider();

private:
  CAPICryptoFactory() {}
};

} // end namespace gdcm

#endif //GDCMCAPICRYPTOFACTORY_H
```

13.13 gdcmCAPICryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include <windows.h>
#include <wincrypt.h>
#include <vector>
```

Include dependency graph for gdcmCAPICryptographicMessageSyntax.h:



Classes

- class gdcm::CAPICryptographicMessageSyntax

Namespaces

- namespace gdcm

13.14 gdcmCAPICryptographicMessageSyntax.h

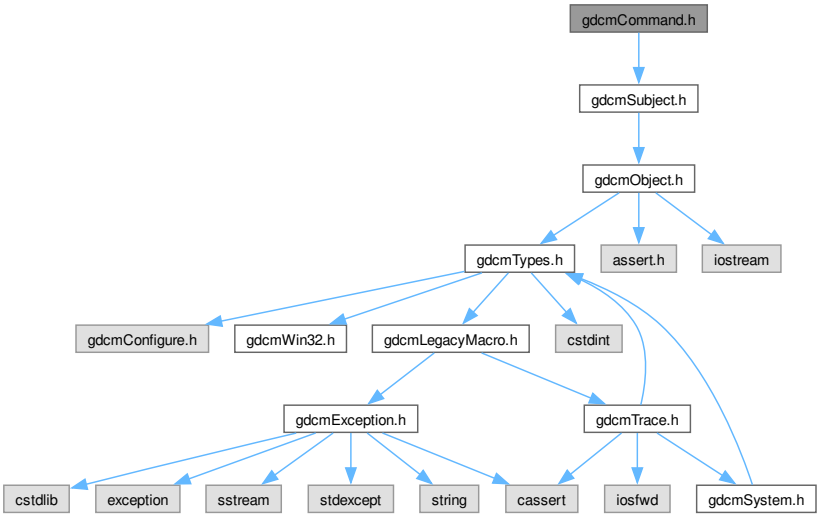
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
#define GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H

#include "gdcmCryptographicMessageSyntax.h"
#include <windows.h>
#include <wincrypt.h>
#include <vector>

namespace gdcm
{

class GDCM_EXPORT CAPICryptographicMessageSyntax : public CryptographicMessageSyntax
{
public:
```

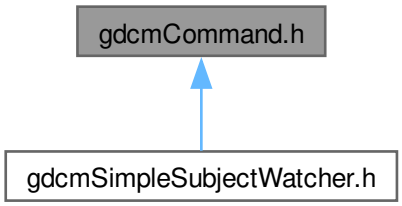
```
  CAPICryptographicMessageSyntax();
  ~CAPICryptographicMessageSyntax();

  // X.509
  bool ParseCertificateFile( const char *filename );
  bool ParseKeyFile( const char *filename );

  // PBE
  bool SetPassword(const char * pass, size_t passLen);

  void SetCipherType(CipherTypes type);

  CipherTypes GetCipherType() const;

  bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
  bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;

  bool GetInitialized() const
  {
    return initialized;
  }

private:
  bool Initialize();
  static ALG_ID GetAlgIdByObjId(const char * pszObjId);
  const char *GetCipherObjId() const;
  static void ReverseBytes(unsigned char* data, DWORD len);
  static bool LoadFile(const char * filename, unsigned char* & buffer, DWORD & bufLen);

private:
  bool initialized;
  HCRYPTPROV hProv;
  std::vector<PCCERT_CONTEXT> certifList;
  HCRYPTKEY hRsaPrivK;
  CipherTypes cipherType;
};

} // end namespace gdcm

#endif //GDCMCAPICRYPTOGRAPHICMESSAGESYNTAX_H
```

13.15 gdcmCommand.h File Reference

`#include "gdcmSubject.h"`
Include dependency graph for gdcmCommand.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Command

  Command superclass for callback/observer methods.
- class gdcm::MemberCommand< T >

  Command subclass that calls a pointer to a member function.
- class gdcm::SimpleMemberCommand< T >

  Command subclass that calls a pointer to a member function.

Namespaces

- namespace gdcm

13.16 gdcmCommand.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCOMMAND_H
#define GDCMCOMMAND_H

#include "gdcmSubject.h"

namespace gdcm
{
class Event;

class GDCM_EXPORT Command : public Subject
{
public :
  Command(const Command&) = delete;
  void operator=(const Command&) = delete;

  virtual void Execute(Subject *caller, const Event & event ) = 0;

  virtual void Execute(const Subject *caller, const Event & event ) = 0;

protected:
  Command();
  ~Command() override;
};

template <class T>
class MemberCommand : public Command
{
public:

  typedef  void (T::*TMemberFunctionPointer)(Subject*, const Event &);
  typedef  void (T::*TConstMemberFunctionPointer)(const Subject*,
                                                  const Event &);

  typedef MemberCommand       Self;
  //typedef SmartPointer<Self>  Pointer;

  MemberCommand(const Self&) = delete;
  void operator=(const Self&) = delete;

  static SmartPointer<MemberCommand> New()
    {
    return new MemberCommand;
    }

  //gdcmTypeMacro(MemberCommand,Command);

  void SetCallbackFunction(T* object,
                           TMemberFunctionPointer memberFunction)
    {
    m_This = object;
    m_MemberFunction = memberFunction;
    }
  void SetCallbackFunction(T* object,
```

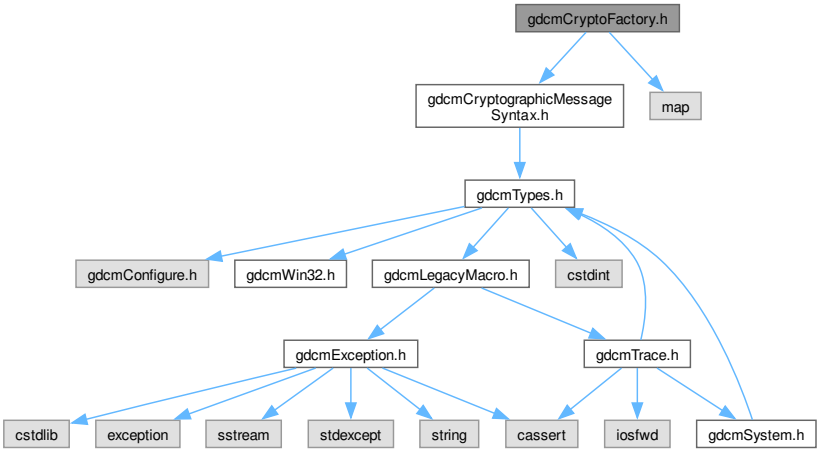
```
                            TConstMemberFunctionPointer memberFunction)
    {
    m_This = object;
    m_ConstMemberFunction = memberFunction;
    }

  void Execute(Subject *caller, const Event & event ) override
    {
    if( m_MemberFunction )
      {
      ((*m_This).*(m_MemberFunction))(caller, event);
      }
    }

  void Execute( const Subject *caller, const Event & event ) override
    {
    if( m_ConstMemberFunction )
      {
      ((*m_This).*(m_ConstMemberFunction))(caller, event);
      }
    }

protected:

  T* m_This;
  TMemberFunctionPointer m_MemberFunction;
  TConstMemberFunctionPointer m_ConstMemberFunction;
  MemberCommand():m_MemberFunction(nullptr),m_ConstMemberFunction(nullptr) {}
  ~MemberCommand() override= default;

};

template <typename T>
class SimpleMemberCommand : public Command
{
public:

  typedef  void (T::*TMemberFunctionPointer)();

  typedef SimpleMemberCommand   Self;
  //typedef SmartPointer<Self>    Pointer;

  SimpleMemberCommand(const Self&) = delete;
  void operator=(const Self&) = delete;

  //gdcmTypeMacro(SimpleMemberCommand,Command);

  static SmartPointer<SimpleMemberCommand> New()
    {
    return new SimpleMemberCommand;
    }

  void SetCallbackFunction(T* object,
                            TMemberFunctionPointer memberFunction)
    {
    m_This = object;
    m_MemberFunction = memberFunction;
    }

  void Execute(Subject *,const Event & ) override
    {
    if( m_MemberFunction )
      {
      ((*m_This).*(m_MemberFunction))();
      }
    }
  void Execute(const Subject *,const Event & ) override
    {
    if( m_MemberFunction )
      {
      ((*m_This).*(m_MemberFunction))();
      }
    }

protected:
  T* m_This;
  TMemberFunctionPointer m_MemberFunction;
  SimpleMemberCommand():m_This(nullptr),m_MemberFunction(nullptr) {}
  ~SimpleMemberCommand() override = default;
};
```

```
} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMCOMMAND_H
```

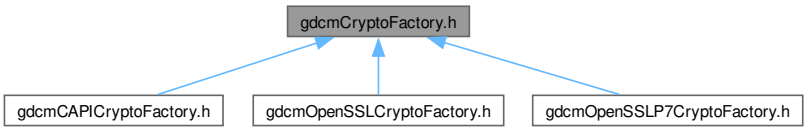
13.17 gdcmCryptoFactory.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include <map>
```

Include dependency graph for gdcmCryptoFactory.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CryptoFactory

  Class to do handle the crypto factory.

Namespaces

- namespace gdcm

13.18 gdcmCryptoFactory.h

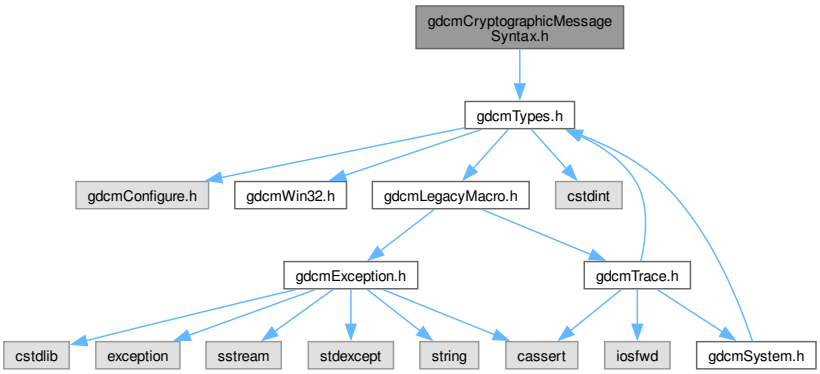
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCRYPTOFACTORY_H
#define GDCMCRYPTOFACTORY_H

#include "gdcmCryptographicMessageSyntax.h"
#include <map>

namespace gdcm
{

class GDCM_EXPORT CryptoFactory
{
public:
  enum CryptoLib {DEFAULT = 0, OPENSSL = 1, CAPI = 2, OPENSSLP7 = 3};

  virtual CryptographicMessageSyntax* CreateCMSProvider() = 0;
  static CryptoFactory* GetFactoryInstance(CryptoLib id = DEFAULT);

protected:
  CryptoFactory(CryptoLib id)
  {
    AddLib(id, this);
  }

private:
  static std::map<CryptoLib, CryptoFactory*>& getInstanceMap()
  {
    static std::map<CryptoLib, CryptoFactory*> libs;
    return libs;
  }

  static void AddLib(CryptoLib id, CryptoFactory* f)
  {
    if (getInstanceMap().insert(std::pair<CryptoLib, CryptoFactory*>(id, f)).second == false)
      {
      gdcmErrorMacro( "Library already registered under id " « (int)id );
      }
  }

protected:
  CryptoFactory()= default;
  ~CryptoFactory()= default;
};

} // end namespace gdcm

#endif // GDCMCRYPTOFACTORY_H
```

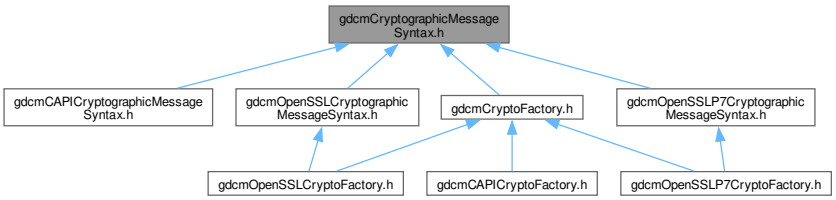
13.19 gdcmCryptographicMessageSyntax.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CryptographicMessageSyntax

Namespaces

- namespace gdcm

13.20 gdcmCryptographicMessageSyntax.h

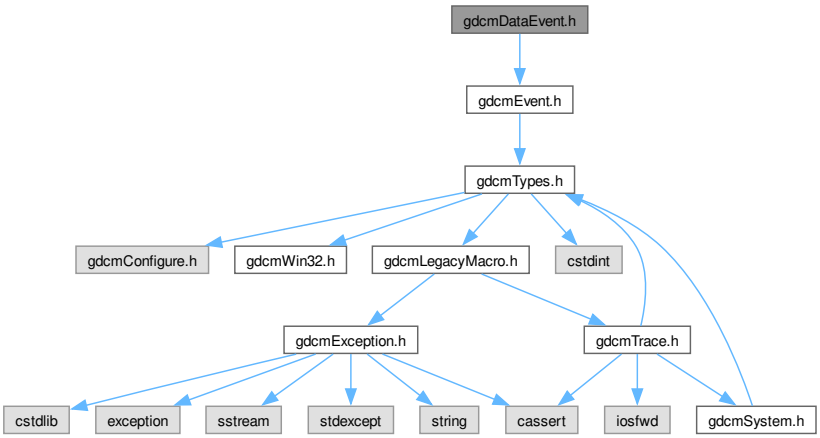
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
#define GDCMCRYPTOGRAPHICMESSAGESYNTAX_H

#include "gdcmTypes.h"

namespace gdcm
{

class GDCM_EXPORT CryptographicMessageSyntax
{
public:
  CryptographicMessageSyntax() = default;

  virtual ~CryptographicMessageSyntax() = default;
  CryptographicMessageSyntax(const CryptographicMessageSyntax&) = delete;
  void operator=(const CryptographicMessageSyntax&) = delete;

  typedef enum {
    DES3_CIPHER,   // Triple DES
    AES128_CIPHER, // CBC AES
    AES192_CIPHER, // '   '
    AES256_CIPHER  // '   '
  } CipherTypes;

    // X.509
  virtual bool ParseCertificateFile( const char *filename ) = 0;
  virtual bool ParseKeyFile( const char *filename ) = 0;

  // PBE
  virtual bool SetPassword(const char * pass, size_t passLen) = 0;

  virtual bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const = 0;
  virtual bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const = 0;

  virtual void SetCipherType(CipherTypes type) = 0;

  virtual CipherTypes GetCipherType() const = 0;
};

} // end namespace gdcm

#endif //GDCMCRYPTOGRAPHICMESSAGESYNTAX_H
```

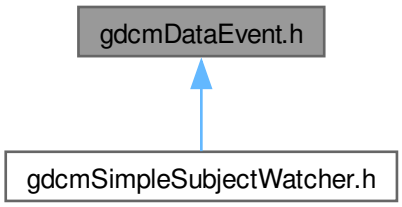
13.21 gdcmDataEvent.h File Reference

```
#include "gdcmEvent.h"
```

Include dependency graph for gdcmDataEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::DataEvent

  DataEvent.

Namespaces

- namespace gdcm

13.22 gdcmDataEvent.h

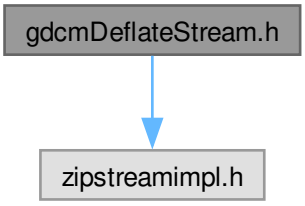
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDATAEVENT_H
#define GDCMDATAEVENT_H

#include "gdcmEvent.h"

namespace gdcm
{

class DataEvent : public AnyEvent
{
public:
  typedef DataEvent Self;
  typedef AnyEvent Superclass;
  DataEvent(const char *bytes = nullptr, size_t len = 0):Bytes(bytes),Length(len) {}
  ~DataEvent() override = default;
  DataEvent(const Self&s) : AnyEvent(s){};
  void operator=(const Self&) = delete;

  const char * GetEventName() const override { return "DataEvent"; }
  bool CheckEvent(const ::gdcm::Event* e) const override
  { return (dynamic_cast<const Self*>(e) == nullptr ? false : true) ; }
  ::gdcm::Event* MakeObject() const override
    { return new Self; }

  void SetData(const char *bytes, size_t len) {
    Bytes = bytes;
    Length = len;
  }
  size_t GetDataLength() const { return Length; }
  const char *GetData() const { return Bytes; }

  //std::string GetValueAsString() const { return; }

private:
  const char *Bytes;
  size_t Length;
};


} // end namespace gdcm

#endif //GDCMDATAEVENT_H
```

13.23 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

Include dependency graph for gdcmDeflateStream.h:



13.24 gdcmDeflateStream.h

Go to the documentation of this file.

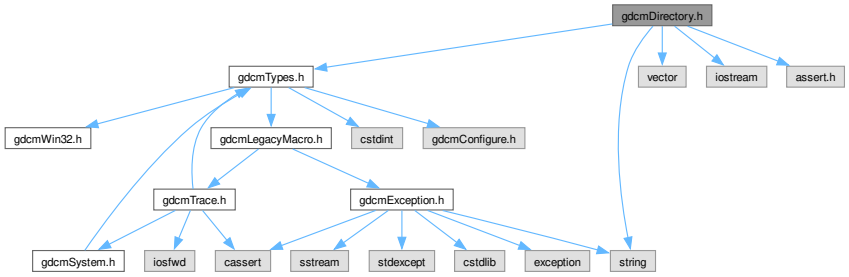
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDEFLATESTREAM_H
#define GDCMDEFLATESTREAM_H

#include "zipstreamimpl.h"

#endif //GDCMDEFLATESTREAM_H
```

13.25 gdcmDirectory.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
#include <iostream>
```

```
#include <assert.h>
```

Include dependency graph for gdcmDirectory.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Directory

  Class for manipulation directories.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Directory &d)

13.26 gdcmDirectory.h

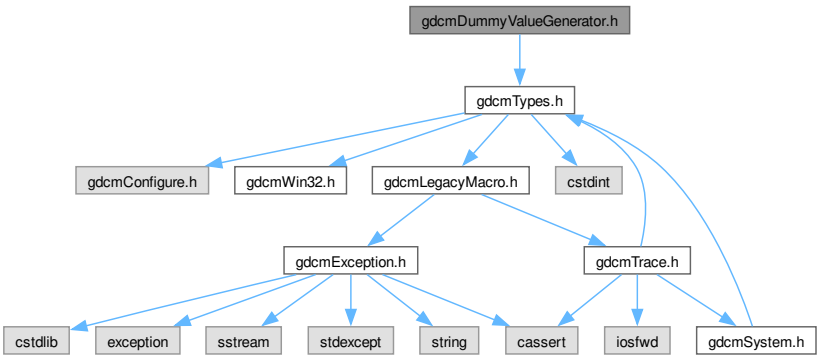
Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
```

```
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDIRECTORY_H
#define GDCMDIRECTORY_H

#include "gdcmTypes.h"

#include <string>
#include <vector>
#include <iostream>
#include <assert.h>

namespace gdcm
{
//-----------------------------------------------------------------------------
class GDCM_EXPORT Directory
{
  friend std::ostream& operator«(std::ostream &_os, const Directory &d);
public :
  Directory() = default;
  ~Directory() = default;
  typedef std::string FilenameType;
  typedef std::vector<FilenameType> FilenamesType;

  void Print(std::ostream &os = std::cout) const;

  FilenameType const &GetToplevel() const { return Toplevel; }

  FilenamesType const &GetFilenames() const {
    assert( !(Toplevel.empty()) && "Need to call Explore first" );
    return Filenames; }

  FilenamesType const &GetDirectories() const { return Directories; }

  unsigned int Load(FilenameType const &name, bool recursive = false);

  // \todo later: GLOB
  // The glob() function searches for all the pathnames matching pattern according to
  // the rules used by the shell (see glob(7)).  No tilde expansion or parameter
  // substitution is done; if you want these, use wordexp(3).
  // int Glob(...);

protected:
  unsigned int Explore(FilenameType const &name, bool recursive);

private :
  FilenamesType Filenames;
  FilenamesType Directories;

  FilenameType Toplevel;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Directory &d)
{
  d.Print( os );
  return os;
}

} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMDIRECTORY_H
```

13.27 gdcmDummyValueGenerator.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmDummyValueGenerator.h:



Classes

- class gdcm::DummyValueGenerator

  Class for generating dummy value.

Namespaces

- namespace gdcm

13.28 gdcmDummyValueGenerator.h

Go to the documentation of this file.

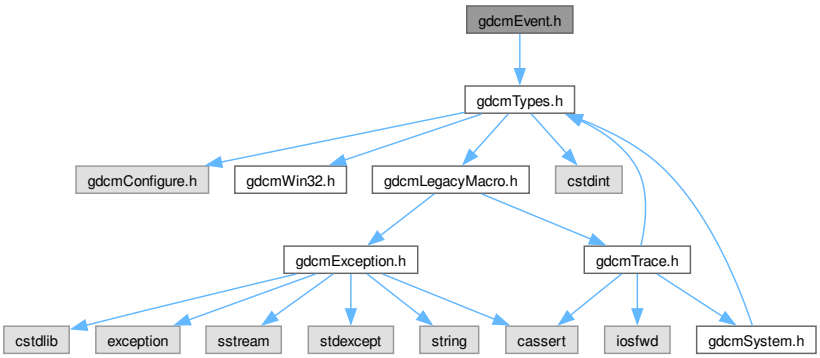
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDUMMYVALUEGENERATOR_H
#define GDCMDUMMYVALUEGENERATOR_H

#include "gdcmTypes.h"

namespace gdcm
{

```

```
00026 class GDCM_EXPORT DummyValueGenerator
00027 {
00028 public:
00029
00035   static const char* Generate(const char *input);
00036
00037 private:
00038 };
00039
00040
00041 } // end namespace gdcm
00042
00043 #endif //GDCMDUMMYVALUEGENERATOR_H
```

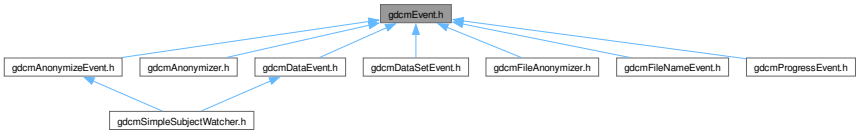
13.29 gdcmEvent.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::AbortEvent
- class gdcm::AnyEvent
- class gdcm::EndEvent
- class gdcm::Event

  superclass for callback/observer methods

- class gdcm::ExitEvent
- class gdcm::InitializeEvent
- class gdcm::IterationEvent
- class gdcm::ModifiedEvent
- class gdcm::NoEvent
- class gdcm::StartEvent
- class gdcm::UserEvent

Namespaces

- namespace gdcm

Macros

- #define gdcmEventMacro(classname, super)

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Event &e)

  Generic inserter operator for Event and its subclasses.

13.29.1 Macro Definition Documentation

13.29.1.1 gdcmEventMacro

```
#define gdcmEventMacro(
            classname,
            super)
```

Value:

```
 \
class  classname : public super { \
  public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
    virtual ~classname() override = default; \
    virtual const char * GetEventName() const override { return #classname; } \
    virtual bool CheckEvent(const ::gdcm::Event* e) const override \
      { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const override \
      { return new Self; } \
    classname(const Self&) : super(s){} \
  private: \
    void operator=(const Self&); \
}
```

13.30 gdcmEvent.h

Go to the documentation of this file.

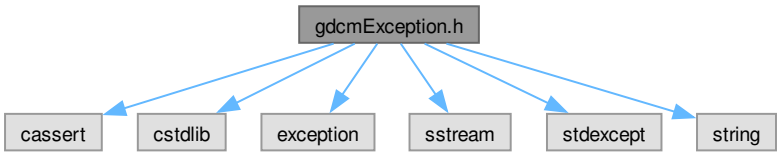
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMEVENT_H
#define GDCMEVENT_H

#include "gdcmTypes.h"

namespace gdcm
{
//-----------------------------------------------------------------------------
class GDCM_EXPORT Event
{
public :
  Event();
  virtual ~Event();
  Event(const Event&);
  void operator=(const Event&) = delete;

  virtual Event* MakeObject() const = 0;

  virtual void Print(std::ostream& os) const;

  virtual const char * GetEventName() const = 0;

  virtual bool CheckEvent(const Event*) const = 0;

};

inline std::ostream& operator«(std::ostream& os, const Event &e)
{
  e.Print(os);
  return os;
}

/*
 *  Macro for creating new Events
 */
#define gdcmEventMacro( classname , super ) \
  \
 class  classname : public super { \
   public: \
     typedef classname Self; \
     typedef super Superclass; \
     classname() {} \
     virtual ~classname() override = default; \
     virtual const char * GetEventName() const override { return #classname; } \
     virtual bool CheckEvent(const ::gdcm::Event* e) const override \
       { return dynamic_cast<const Self*>(e) ? true : false; } \
     virtual ::gdcm::Event* MakeObject() const override \
       { return new Self; } \
     classname(const Self&s) : super(s){} \
   private: \
     void operator=(const Self&); \
 }

gdcmEventMacro( NoEvent            , Event );
gdcmEventMacro( AnyEvent           , Event );
gdcmEventMacro( StartEvent         , AnyEvent );
gdcmEventMacro( EndEvent           , AnyEvent );
//gdcmEventMacro( ProgressEvent      , AnyEvent );
gdcmEventMacro( ExitEvent          , AnyEvent );
gdcmEventMacro( AbortEvent         , AnyEvent );
gdcmEventMacro( ModifiedEvent      , AnyEvent );
gdcmEventMacro( InitializeEvent    , AnyEvent );
```

```
gdcmEventMacro( IterationEvent      , AnyEvent );
//gdcmEventMacro( AnonymizeEvent      , AnyEvent );
gdcmEventMacro( UserEvent           , AnyEvent );


} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMEVENT_H
```

13.31 gdcmException.h File Reference

```
#include <cassert>
#include <cstdlib>
#include <exception>
#include <sstream>
#include <stdexcept>
#include <string>
```

Include dependency graph for gdcmException.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Exception

  Exception.

Namespaces

- namespace gdcm

13.32 gdcmException.h

Go to the documentation of this file.

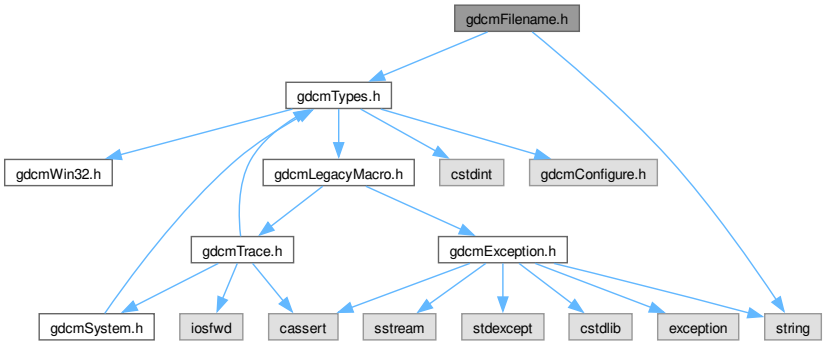
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMEXCEPTION_H
#define GDCMEXCEPTION_H

#include <cassert>
#include <cstdlib> // NULL
#include <exception>
#include <sstream> // ostringstream
#include <stdexcept> // logic_error
#include <string>

// Disable clang warning "dynamic exception specifications are deprecated".
// We need to be C++03 and C++11 compatible, and if we remove the 'throw()'
// specifier we'll get an error in C++03 by not matching the superclass.
#if defined(__clang__) && defined(__has_warning)
# if __has_warning("-Wdeprecated")
#  pragma clang diagnostic push
#  pragma clang diagnostic ignored "-Wdeprecated"
# endif
#endif

namespace gdcm
{

class Exception : public std::exception
{
  typedef std::logic_error StringHolder;

  static StringHolder CreateWhat(const char* const desc,
                                 const char* const file,
                                 const unsigned int lineNumber,
                                 const char* const func)
  {
    assert(desc != nullptr);
    assert(file != nullptr);
    assert(func != nullptr);
    std::ostringstream oswhat;
    oswhat « file « ":" « lineNumber « " (" « func « "):\n";
    oswhat « desc;
    return StringHolder( oswhat.str() );
  }

public:
  explicit Exception(const char *desc = "None",
            const char *file = __FILE__,
            unsigned int lineNumber = __LINE__,
            // FIXME:  __PRETTY_FUNCTION__ is the non-mangled version for __GNUC__ compiler
            const char *func = "" /*__FUNCTION__*/)
  :
  What( CreateWhat(desc, file, lineNumber, func) ),
  Description(desc)
  {
  }

  ~Exception() throw() override {}

  const char* what() const throw() override
    {
    return What.what();
    }

  const char * GetDescription() const { return Description.what(); }

```

```
private:
  StringHolder  What;
  StringHolder  Description;
};

} // end namespace gdcm

// Undo warning suppression.
#if defined(__clang__) && defined(__has_warning)
# if __has_warning("-Wdeprecated")
#  pragma clang diagnostic pop
# endif
#endif

#endif
```

13.33 gdcmFilename.h File Reference

```
#include "gdcmTypes.h"
#include <string>
```

Include dependency graph for gdcmFilename.h:



Classes

- class gdcm::Filename

  Class to manipulate file name's.

Namespaces

- namespace gdcm

13.34 gdcmFilename.h

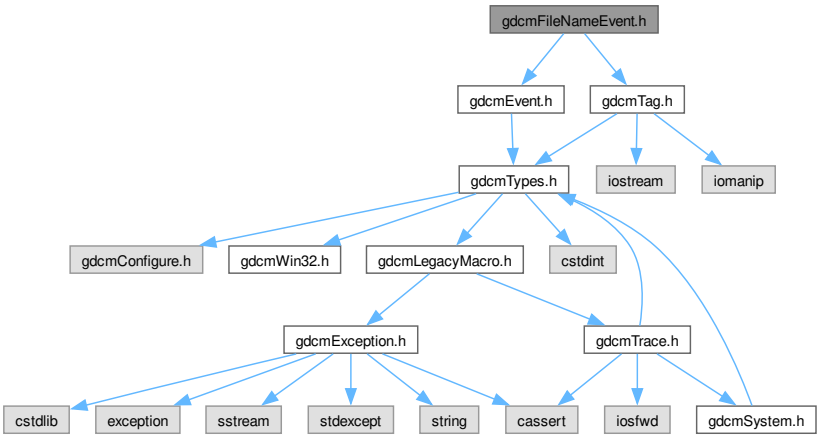
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILENAME_H
#define GDCMFILENAME_H

#include "gdcmTypes.h"

#include <string>

namespace gdcm
{
class GDCM_EXPORT Filename
{
public:
  Filename(const char* filename = ""):FileName(filename ? filename : ""),Path(),Conversion() {}

  const char *GetFileName() const { return FileName.c_str(); }
  const char *GetPath();
  const char *GetName();
  const char *GetExtension();
  const char *ToUnixSlashes();
  const char *ToWindowsSlashes();

  static const char *Join(const char *path, const char *filename);

  bool IsEmpty() const { return FileName.empty(); }

  operator const char * () const { return GetFileName(); }

  // FIXME: I don't like this function
  // It hides the realpath call (maybe useful)
  // and it forces file to exist on the disk whereas Filename
  // should be independent from file existence.
  bool IsIdentical(Filename const &fn) const;

  bool EndWith(const char ending[]) const;

private:
  std::string FileName;
  std::string Path;
  std::string Conversion;
};

} // end namespace gdcm

#endif //GDCMFILENAME_H
```

13.35 gdcmFileNameEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmFileNameEvent.h:



Classes

- class gdcm::FileNameEvent

  FileNameEvent.

Namespaces

- namespace gdcm

13.36 gdcmFileNameEvent.h

Go to the documentation of this file.

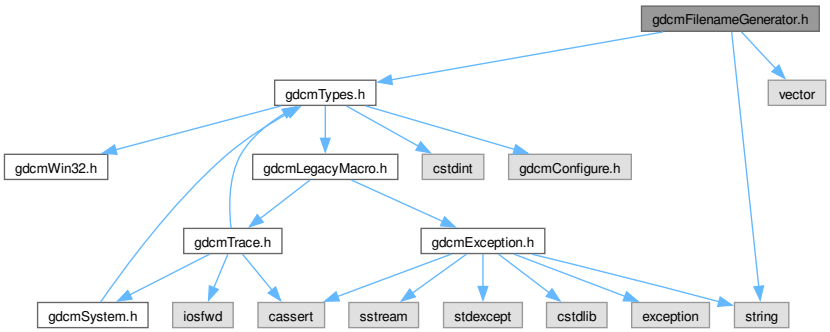
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILENAMEEVENT_H
#define GDCMFILENAMEEVENT_H

#include "gdcmEvent.h"
#include "gdcmTag.h"

namespace gdcm
{

class FileNameEvent : public AnyEvent
```

```
{
public:
  typedef FileNameEvent Self;
  typedef AnyEvent Superclass;
  FileNameEvent(const char *s = ""):m_FileName(s) {}
  ~FileNameEvent() override = default;

  FileNameEvent(const Self&s) : AnyEvent(s){}
  void operator=(const Self&) = delete;


  const char * GetEventName() const override { return "FileNameEvent"; }
  bool CheckEvent(const ::gdcm::Event* e) const override
    { return dynamic_cast<const Self*>(e) ? true : false; }
  ::gdcm::Event* MakeObject() const override
    { return new Self; }

  void SetFileName(const char *f) { m_FileName = f; }
  const char *GetFileName() const { return m_FileName.c_str(); }
private:
  std::string m_FileName;
};


} // end namespace gdcm

#endif //GDCMFILENAMEEVENT_H
```

13.37 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
```

Include dependency graph for gdcmFilenameGenerator.h:



Classes

- class gdcm::FilenameGenerator

  FilenameGenerator.

Namespaces

- namespace gdcm

13.38 gdcmFilenameGenerator.h

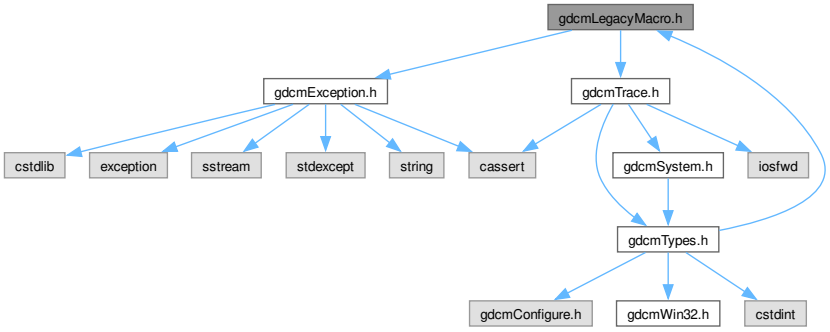
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILENAMEGENERATOR_H
#define GDCMFILENAMEGENERATOR_H

#include "gdcmTypes.h"
#include <string>
#include <vector>


namespace gdcm
{

class GDCM_EXPORT FilenameGenerator
{
public:
  FilenameGenerator():Pattern(),Prefix(),Filenames() {}
  ~FilenameGenerator() = default;
  // FIXME: already defines in gdcm::Directory
  typedef std::string FilenameType;
  typedef std::vector<FilenameType> FilenamesType;
  typedef FilenamesType::size_type  SizeType;

  void SetPattern(const char *pattern) { Pattern = pattern; }
  const char *GetPattern() const { return Pattern.c_str(); }

  void SetPrefix(const char *prefix) { Prefix = prefix; }
  const char *GetPrefix() const { return Prefix.c_str(); }

  bool Generate();

  void SetNumberOfFilenames(SizeType nfiles);
  SizeType GetNumberOfFilenames() const;

  const char * GetFilename(SizeType n) const;

  FilenamesType const & GetFilenames() const { assert( !Pattern.empty() ); return Filenames; }

private:
  FilenameType Pattern;
  FilenameType Prefix;
  FilenamesType Filenames;
};

} // end namespace gdcm

#endif //GDCMFILENAMEGENERATOR_H
```

13.39 gdcmLegacyMacro.h File Reference

```
#include "gdcmException.h"
#include "gdcmTrace.h"
```

Include dependency graph for gdcmLegacyMacro.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define GDCM_LEGACY(method)
- #define GDCM_LEGACY_BODY(method, version)
- #define GDCM_LEGACY_REPLACED_BODY(method, version, replace)
- #define GDCM_NOOP_STATEMENT static_assert(true, "")

13.39.1 Macro Definition Documentation

13.39.1.1 GDCM_LEGACY

```
#define GDCM_LEGACY(
            method)
```

Value:

```
method;
```

13.39.1.2 GDCM_LEGACY_BODY

```
#define GDCM_LEGACY_BODY(
            method,
            version)
```

Value:

```
gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")
```

13.39.1.3 GDCM_LEGACY_REPLACED_BODY

```
#define GDCM_LEGACY_REPLACED_BODY(
              method,
              version,
              replace)
```

Value:

```
gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.  Use "
    #replace " instead.")
```

13.39.1.4 GDCM_NOOP_STATEMENT

```
#define GDCM_NOOP_STATEMENT static_assert(true, "")
```

The `static_assert(true, "")` idiom is commonly employed for C++11 or greater to ensure that it is compile-time only check that can not be part of the binary file. This allows a macro to be used anywhere that a statement is expected, and to enforce consistent use of ; after a macro. The static_assert is a constexpr that can be used in places where raw statements (i.e. 'do{} while(0)') are not allowed (i.e. after class member function definitions).

13.40 gdcmLegacyMacro.h

Go to the documentation of this file.

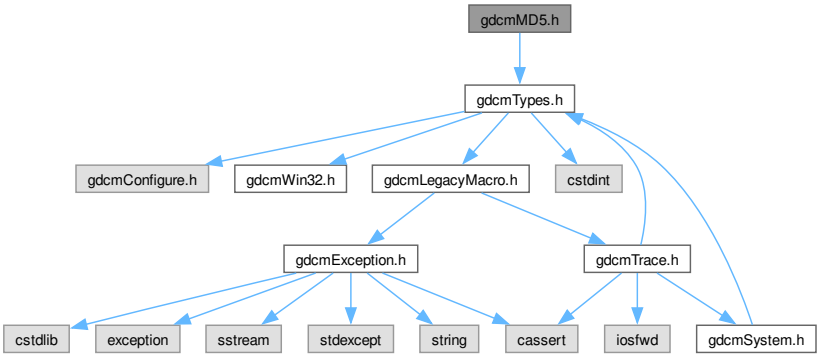
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMLEGACYMACRO_H
#define GDCMLEGACYMACRO_H

#if !defined(GDCMTYPES_H) && !defined(SWIG)
#error you need to include gdcmTypes.h instead
#endif

#include "gdcmException.h"

//-----------------------------------------------------------------------------
// Setup legacy code policy.

// Define GDCM_LEGACY macro to mark legacy methods where they are
// declared in their class.  Example usage:
//
//   // @deprecated Replaced by MyOtherMethod() as of GDCM 2.0.
//   GDCM_LEGACY(void MyMethod());
#if defined(GDCM_LEGACY_REMOVE)
# define GDCM_LEGACY(method)
#elif defined(GDCM_LEGACY_SILENT) || defined(SWIG)
  // Provide legacy methods with no warnings.
# define GDCM_LEGACY(method) method;
#else
  // Setup compile-time warnings for uses of deprecated methods if
  // possible on this compiler.
# if defined(__GNUC__) && !defined(__INTEL_COMPILER) && (__GNUC__ > 3 || (__GNUC__ == 3 && __GNUC_MINOR__
    >= 1))
```

```
#  define GDCM_LEGACY(method) method __attribute__((deprecated));
# elif defined(_MSC_VER) && _MSC_VER >= 1300
#  define GDCM_LEGACY(method) __declspec(deprecated) method;
# else
#  define GDCM_LEGACY(method) method;
# endif
#endif

#  define GDCM_NOOP_STATEMENT static_assert(true, "")

// Macros to create runtime deprecation warning messages in function
// bodies.  Example usage:
//
//   #if !defined(GDCM_LEGACY_REMOVE)
//   void gdcm::MyClass::MyOldMethod()
//   {
//     GDCM_LEGACY_BODY(gdcm::MyClass::MyOldMethod, "GDCM 2.0");
//   }
//   #endif
//
//   #if !defined(GDCM_LEGACY_REMOVE)
//   void gdcm::MyClass::MyMethod()
//   {
//     GDCM_LEGACY_REPLACED_BODY(gdcm::MyClass::MyMethod, "GDCM 2.0",
//                               gdcm::MyClass::MyOtherMethod);
//   }
//   #endif
#if defined(GDCM_LEGACY_REMOVE) || defined(GDCM_LEGACY_SILENT)
# define GDCM_LEGACY_BODY(method, version)
# define GDCM_LEGACY_REPLACED_BODY(method, version, replace)
#else
# define GDCM_LEGACY_BODY(method, version) \
  gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")
# define GDCM_LEGACY_REPLACED_BODY(method, version, replace) \
  gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.  Use " #replace " instead.")
#endif

#include "gdcmTrace.h"

#endif // GDCMLEGACYMACRO_H
```

13.41 gdcmMD5.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmMD5.h:



Classes

- class gdcm::MD5

  Class for MD5.

Namespaces

- namespace gdcm

13.42 gdcmMD5.h

Go to the documentation of this file.

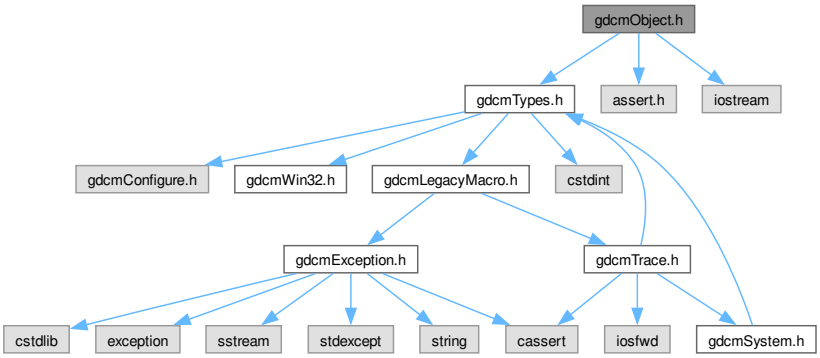
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMMD5_H
00015 #define GDCMMD5_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021 //-----------------------------------------------------------------------------
00032 class GDCM_EXPORT MD5
00033 {
00034 public :
00035   // Compute md5 from memory pointed by `pointer` of size `buf_len`
00036   static bool Compute(const char *buffer, size_t buf_len, char digest_str[33]);
00037
00039   static bool ComputeFile(const char *filename, char digest_str[33]);
00040 };
00041 } // end namespace gdcm
00042 //-----------------------------------------------------------------------------
00043 #endif //GDCMMD5_H
```

13.43 gdcmObject.h File Reference

```
#include "gdcmTypes.h"
#include <assert.h>
```

```
#include <iostream>
```

Include dependency graph for gdcmObject.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Object

  Object.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)

13.44 gdcmObject.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
```

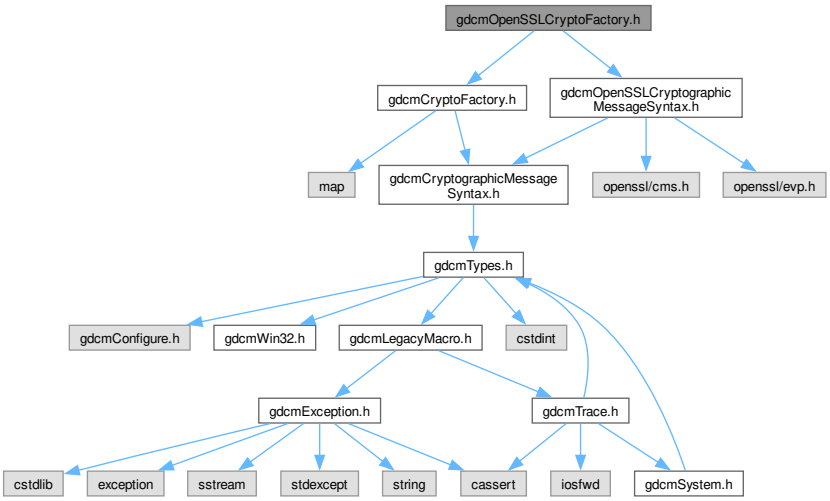
```
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMOBJECT_H
#define GDCMOBJECT_H

#include "gdcmTypes.h"

#include <assert.h>
#include <iostream> // grrrr

//namespace std { class ostream; }
namespace gdcm
{

template<class ObjectType> class SmartPointer;

class GDCM_EXPORT Object
{
  template <class ObjectType> friend class SmartPointer;
  friend std::ostream& operator«(std::ostream &os, const Object &obj);

public:
  Object():ReferenceCount(0) {}

  // Implementation note:
  // If I move ~Object in the protected section I can prevent people
  // from writing:
  // SmartPointer<Object> p = new Object;
  // delete p; // due to SmartPointer::operator ObjectType * () const
  // but on the other hand one could not define an Object on the stack
  // Object obj;
  // Furthermore it would not prevent anyone from doing:
  // class MyObject : public Object {};
  // SmartPointer<MyObject> o = new MyObject;
  // delete o; // grrrrrr
  virtual ~Object() {
    // If your debugger reach here it means you are doing something silly
    // like using SmartPointer on object allocated on the stack (vs heap)
    assert(ReferenceCount == 0);
    }

  // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.24
  // Set the ref count to 0
  // Do NOT copy the reference count !
  Object(const Object&):ReferenceCount(0){}
  void operator=(const Object&){}

  //static Object* New() { return new Object; }

protected:
  // For the purpose of the invasive SmartPointer implementation
  void Register() {
    ReferenceCount++;
    assert( ReferenceCount > 0 );
  }
  void UnRegister() {
    assert( ReferenceCount > 0 );
    ReferenceCount--;
    if(!ReferenceCount)
      {
      delete this;
      }
    }

public:
  // For dealing with printing of object and polymorphism
  virtual void Print(std::ostream &) const {}

private:
  long ReferenceCount;
};

//-----------------------------------------------------------------------------
// function do not carry vtable. Thus define in the base class the operator
// and use the member function ->Print() to call the appropriate function
// NOTE: All subclass of Object needs to implement the Print function
```

```
inline std::ostream& operator«(std::ostream &os, const Object &obj)
{
  obj.Print(os);
  return os;
}

} // end namespace gdcm

#endif //GDCMOBJECT_H
```

13.45 gdcmOpenSSLCryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLCryptoFactory.h:



Classes

- class gdcm::OpenSSLCryptoFactory

Namespaces

- namespace gdcm

13.46 gdcmOpenSSLCryptoFactory.h

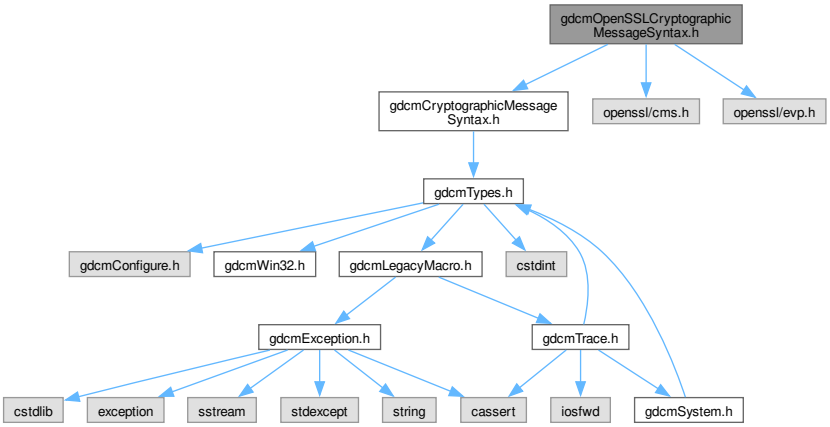
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMOPENSSLCRYPTOFACTORY_H
#define GDCMOPENSSLCRYPTOFACTORY_H

#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"

namespace gdcm
{

class GDCM_EXPORT OpenSSLCryptoFactory : public CryptoFactory
{
public:
  OpenSSLCryptoFactory(CryptoLib id) : CryptoFactory(id)
  {
    gdcmDebugMacro( "OpenSSL Factory registered." );
  }

public:
  CryptographicMessageSyntax* CreateCMSProvider()
  {
    InitOpenSSL();
    return new OpenSSLCryptographicMessageSyntax();
  }

protected:
  void InitOpenSSL();

private:
  OpenSSLCryptoFactory(){}
};

} // end namespace gdcm

#endif //GDCMOPENSSLCRYPTOFACTORY_H
```

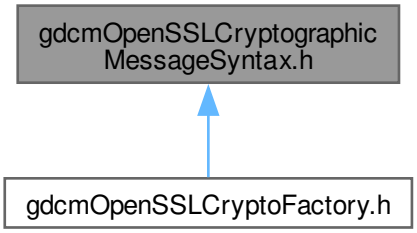
13.47 gdcmOpenSSLCryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include <openssl/cms.h>
#include <openssl/evp.h>
```

Include dependency graph for gdcmOpenSSLCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::OpenSSLCryptographicMessageSyntax

Namespaces

- namespace gdcm

13.48 gdcmOpenSSLCryptographicMessageSyntax.h

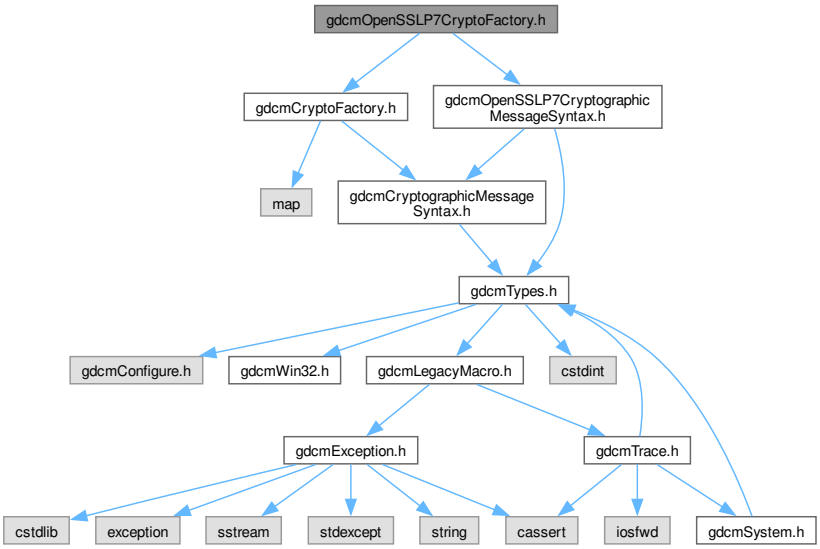
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
#define GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H

#include "gdcmCryptographicMessageSyntax.h"
#include <openssl/cms.h>
#include <openssl/evp.h>

namespace gdcm
{

class GDCM_EXPORT OpenSSLCryptographicMessageSyntax : public CryptographicMessageSyntax
{
public:
  OpenSSLCryptographicMessageSyntax();
  ~OpenSSLCryptographicMessageSyntax();

  // X.509
  bool ParseCertificateFile( const char *filename );
  bool ParseKeyFile( const char *filename );

  // PBE
  bool SetPassword(const char * pass, size_t passLen);

  void SetCipherType(CipherTypes type);
  CipherTypes GetCipherType() const;
  bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;
  bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;

private:
//#ifdef GDCM_HAVE_CMS_RECIPIENT_PASSWORD
//  ::stack_st_X509 *recips;
//#else
  STACK_OF(X509) *recips;
//#endif
  ::EVP_PKEY *pkey;
  const EVP_CIPHER *internalCipherType;
  char * password;
  size_t passwordLength;
  CipherTypes cipherType;

private:
  OpenSSLCryptographicMessageSyntax(const OpenSSLCryptographicMessageSyntax&);  // Not implemented.
  void operator=(const OpenSSLCryptographicMessageSyntax&);  // Not implemented.
  const EVP_CIPHER *CreateCipher( CryptographicMessageSyntax::CipherTypes ciphertype);

};

} // end namespace gdcm

#endif //GDCMOPENSSLCRYPTOGRAPHICMESSAGESYNTAX_H
```

13.49 gdcmOpenSSLP7CryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLP7CryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLP7CryptoFactory.h:



Classes

- class gdcm::OpenSSLP7CryptoFactory

Namespaces

- namespace gdcm

13.50 gdcmOpenSSLP7CryptoFactory.h

Go to the documentation of this file.

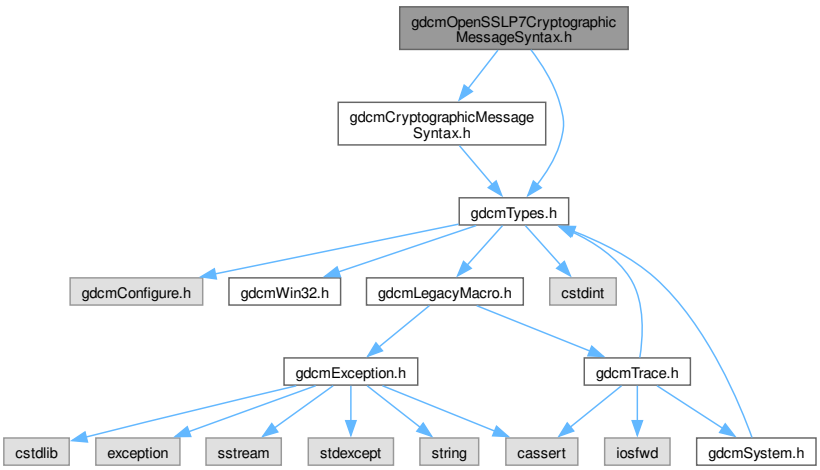
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMOPENSSLP7CRYPTOFACTORY_H
#define GDCMOPENSSLP7CRYPTOFACTORY_H

#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLP7CryptographicMessageSyntax.h"

namespace gdcm
```

```
{
class GDCM_EXPORT OpenSSLP7CryptoFactory : public CryptoFactory
{
public:
  OpenSSLP7CryptoFactory(CryptoLib id) : CryptoFactory(id)
  {
    gdcmDebugMacro( "OpenSSL (PKCS7) Factory registered." );
  }

public:
  CryptographicMessageSyntax* CreateCMSProvider()
  {
    return new OpenSSLP7CryptographicMessageSyntax();
  }

private:
  OpenSSLP7CryptoFactory(){}
};
}

#endif //GDCMOPENSSLP7CRYPTOFACTORY_H
```

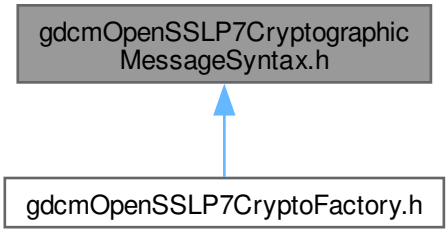
13.51 gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOpenSSLP7CryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::OpenSSLP7CryptographicMessageSyntax

Namespaces

- namespace gdcm

13.52 gdcmOpenSSLP7CryptographicMessageSyntax.h

Go to the documentation of this file.

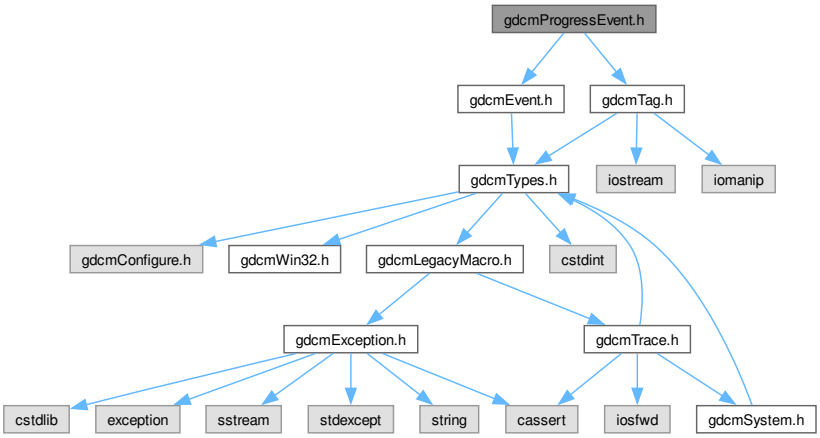
```
00001 /*=========================================================================

00003   Program: GDCM (Grassroots DICOM). A DICOM library

00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.

00013 =========================================================================*/
00014 #ifndef GDCMOPENSSLP7CRYPTOGRAPHICMESSAGESYNTAX_H
00015 #define GDCMOPENSSLP7CRYPTOGRAPHICMESSAGESYNTAX_H

00017 #include "gdcmCryptographicMessageSyntax.h"
00018 #include "gdcmTypes.h"

00020 namespace gdcm
00021 {
00022 class CryptographicMessageSyntaxInternals;
00023 //-----------------------------------------------------------------------------

00034 class GDCM_EXPORT OpenSSLP7CryptographicMessageSyntax : public CryptographicMessageSyntax
00035 {
00036 public:
00037   OpenSSLP7CryptographicMessageSyntax();
00038   ~OpenSSLP7CryptographicMessageSyntax();

```

```
    // X.509
  bool ParseCertificateFile( const char *filename );
  bool ParseKeyFile( const char *filename );

  // PBE
  bool SetPassword(const char * /*pass*/, size_t /*passLen*/)
  {
    gdcmWarningMacro( "Openssl using PKCS7 does not support Password Based Encryption." );
    return false;
  }

  void SetCipherType(CipherTypes type);
  CipherTypes GetCipherType() const;

  bool Encrypt(char *output, size_t &outlen, const char *array, size_t len) const;

  bool Decrypt(char *output, size_t &outlen, const char *array, size_t len) const;

private:
  CryptographicMessageSyntaxInternals *Internals;
private:
  OpenSSLP7CryptographicMessageSyntax(const OpenSSLP7CryptographicMessageSyntax&);  // Not implemented.
  void operator=(const OpenSSLP7CryptographicMessageSyntax&);  // Not implemented.
};
} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMOPENSSLP7CRYPTOGRAPHICMESSAGESYNTAX_H
```

13.53 gdcmProgressEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmProgressEvent.h:



Classes

- class gdcm::ProgressEvent

  ProgressEvent.

Namespaces

- namespace gdcm

13.54 gdcmProgressEvent.h

Go to the documentation of this file.

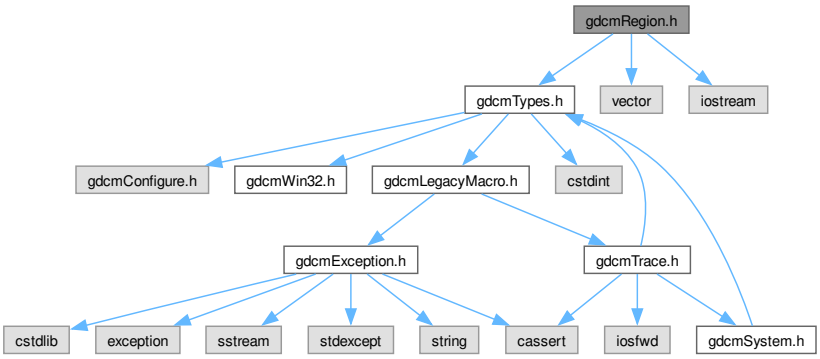
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPROGRESSEVENT_H
#define GDCMPROGRESSEVENT_H

#include "gdcmEvent.h"
#include "gdcmTag.h"

namespace gdcm
{

class ProgressEvent : public AnyEvent
{
public:
  typedef ProgressEvent Self;
  typedef AnyEvent Superclass;
  ProgressEvent(double p = 0):m_Progress(p) {}
  ~ProgressEvent() override = default;

  ProgressEvent(const Self&s) : AnyEvent(s){};
  void operator=(const Self&) = delete;

  const char * GetEventName() const override { return "ProgressEvent"; }
  bool CheckEvent(const ::gdcm::Event* e) const override
    { return dynamic_cast<const Self*>(e) ? true : false; }
  ::gdcm::Event* MakeObject() const override
    { return new Self; }

  void SetProgress(double p) { m_Progress = p; }
  double GetProgress() const { return m_Progress; }
private:
  double m_Progress;
};


} // end namespace gdcm

#endif //GDCMPROGRESSEVENT_H
```

13.55 gdcmRegion.h File Reference

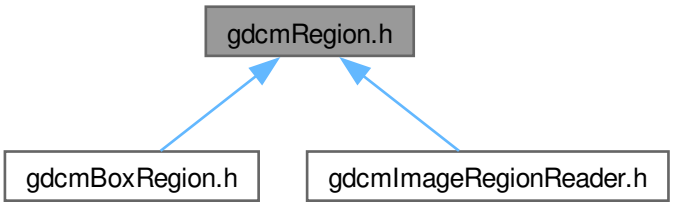
```
#include "gdcmTypes.h"
#include <vector>
```

```
#include <iostream>
```

Include dependency graph for gdcmRegion.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Region

  Class for manipulation region.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)

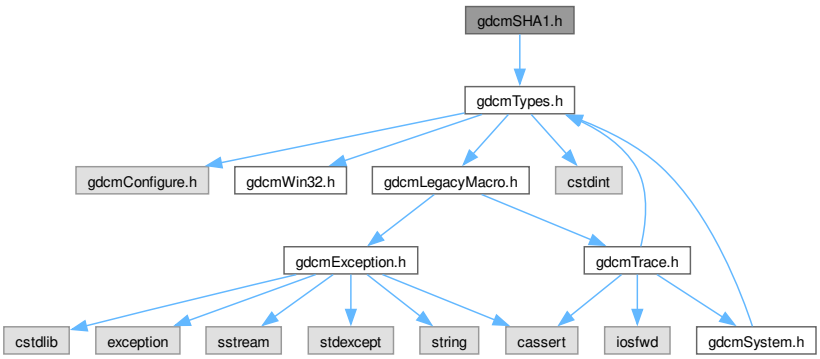
13.56 gdcmRegion.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMREGION_H
#define GDCMREGION_H

#include "gdcmTypes.h"
#include <vector>
#include <iostream>

namespace gdcm
{
class BoxRegion;
//-----------------------------------------------------------------------------
class GDCM_EXPORT Region
{
public :
  Region();
  virtual ~Region();

  virtual void Print(std::ostream &os = std::cout) const;

  virtual bool Empty() const = 0;

  virtual bool IsValid() const = 0;

  virtual size_t Area() const = 0;

  // implementation detail of heterogeneous container in C++
  virtual Region *Clone() const = 0;

  virtual BoxRegion ComputeBoundingBox() = 0;
private:
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Region&r)
{
  r.Print( os );
  return os;
}

} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMREGION_H
```

13.57 gdcmSHA1.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmSHA1.h:



Classes

- class gdcm::SHA1

  Class for SHA1.

Namespaces

- namespace gdcm

13.58 gdcmSHA1.h

Go to the documentation of this file.

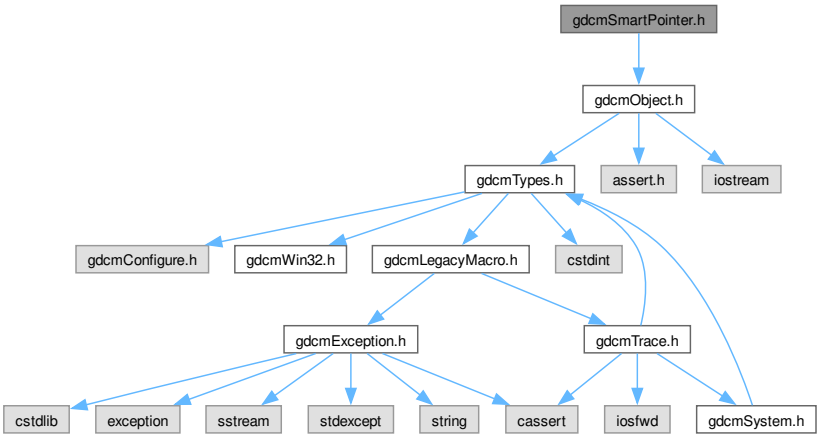
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSHA1_H
#define GDCMSHA1_H

#include "gdcmTypes.h"

namespace gdcm
{
//-----------------------------------------------------------------------------
```

```
class SHA1Internals;
class GDCM_EXPORT SHA1
{
public :
  SHA1();
  ~SHA1();
  SHA1(const SHA1&) = delete;
  void operator=(const SHA1&) = delete;

  static bool Compute(const char *buffer, unsigned long buf_len, char digest_str[20*2+1]);

  static bool ComputeFile(const char *filename, char digest_str[20*2+1]);

private:
  SHA1Internals *Internals;
};
} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMSHA1_H
```

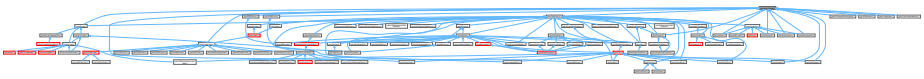
13.59 gdcmSmartPointer.h File Reference

`#include "gdcmObject.h"`

Include dependency graph for gdcmSmartPointer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::SmartPointer< ObjectType >

  Class for Smart Pointer.

Namespaces

- namespace gdcm

13.60 gdcmSmartPointer.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSMARTPOINTER_H
#define GDCMSMARTPOINTER_H

#include "gdcmObject.h"

namespace gdcm
{
template<class ObjectType>
class SmartPointer
{
public:
  SmartPointer():Pointer(nullptr) {}
  SmartPointer(const SmartPointer<ObjectType>& p):Pointer(p.Pointer)
    { Register(); }
  SmartPointer(ObjectType* p):Pointer(p)
    { Register(); }
  SmartPointer(ObjectType const & p)
    {
    Pointer = const_cast<ObjectType*>(&p);
    Register();
    }
  ~SmartPointer() {
    UnRegister();
    Pointer = nullptr;
  }

  ObjectType *operator -> () const
    { return Pointer; }

  ObjectType& operator * () const
    {
    assert( Pointer );
    return *Pointer;
    }

  operator ObjectType * () const
    { return Pointer; }

  SmartPointer &operator = (SmartPointer const &r)
    { return operator = (r.Pointer); }

  SmartPointer &operator = (ObjectType *r)
    {
    // http://www.parashift.com/c++-faq-lite/freestore-mgmt.html#faq-16.22
    // DO NOT CHANGE THE ORDER OF THESE STATEMENTS!
    // (This order properly handles self-assignment)
    // (This order also properly handles recursion, e.g., if a ObjectType contains SmartPointer<ObjectType>s)
    if( Pointer != r )
      {
      ObjectType* old = Pointer;
      Pointer = r;
      Register();
```

```
      if ( old ) { old->UnRegister(); }
      }
    return *this;
    }

  SmartPointer &operator = (ObjectType const &r)
    {
    ObjectType* tmp = const_cast<ObjectType*>(&r);
    return operator = (tmp);
    }

  ObjectType *GetPointer() const
    { return Pointer; }

private:
  void Register()
    {
    if(Pointer) Pointer->Register();
    }

  void UnRegister()
    {
    if(Pointer) Pointer->UnRegister();
    }

  ObjectType* Pointer;
};

} // end namespace gdcm

#endif //GDCMSMARTPOINTER_H
```

13.61 gdcmStaticAssert.h File Reference

This graph shows which files directly or indirectly include this file:

Classes

- struct gdcm::static_assert_test< x >
- struct gdcm::STATIC_ASSERTION_FAILURE< true >

Namespaces

- namespace gdcm

Macros

- #define GDCM_DO_JOIN(X, Y)
- #define GDCM_DO_JOIN2(X, Y)
- #define GDCM_JOIN(X, Y)
- #define GDCM_STATIC_ASSERT(B)

  *The GDCM_JOIN + **LINE** is needed to create a uniq identifier.*

13.61.1 Macro Definition Documentation

13.61.1.1 GDCM_DO_JOIN

```
#define GDCM_DO_JOIN(
              X,
              Y)
```

Value:

```
GDCM_DO_JOIN2(X,Y)
```

13.61.1.2 GDCM_DO_JOIN2

```
#define GDCM_DO_JOIN2(
              X,
              Y)
```

Value:

```
X##Y
```

13.61.1.3 GDCM_JOIN

```
#define GDCM_JOIN(
              X,
              Y)
```

Value:

```
GDCM_DO_JOIN( X, Y )
```

13.61.1.4 GDCM_STATIC_ASSERT

```
#define GDCM_STATIC_ASSERT(
              B)
```

Value:

```
typedef ::gdcm::static_assert_test<\
  sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>\
    GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
```

The GDCM_JOIN + **LINE** is needed to create a uniq identifier.

13.62 gdcmStaticAssert.h

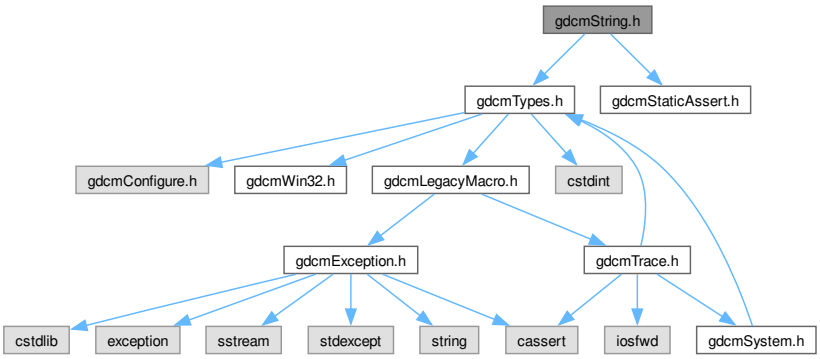
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSTATICASSERT_H
#define GDCMSTATICASSERT_H


// the following was shamelessly borowed from BOOST static assert:
namespace gdcm
{
  template <bool x>
  struct STATIC_ASSERTION_FAILURE;

  template <>
  struct STATIC_ASSERTION_FAILURE<true> { enum { value = 1 }; };

  template <int x>
  struct static_assert_test {};
}

#define GDCM_JOIN( X, Y ) GDCM_DO_JOIN( X, Y )
#define GDCM_DO_JOIN( X, Y ) GDCM_DO_JOIN2(X,Y)
#define GDCM_DO_JOIN2( X, Y ) X##Y

#define GDCM_STATIC_ASSERT( B ) \
  typedef ::gdcm::static_assert_test<\
    sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>\
      GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)


/* Example of use:
 *
 * template <class T>
 * struct must_not_be_instantiated
 * {
 * // this will be triggered if this type is instantiated
 * GDCM_STATIC_ASSERT(sizeof(T) == 0);
 * };
 *
 */
#endif // GDCMSTATICASSERT_H
```

13.63 gdcmString.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmStaticAssert.h"
```

Include dependency graph for gdcmString.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::String< TDelimiter, TMaxLength, TPadChar >

  String.

Namespaces

- namespace gdcm

Functions

- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
  std::istream & gdcm::operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)

13.64 gdcmString.h

Go to the documentation of this file.

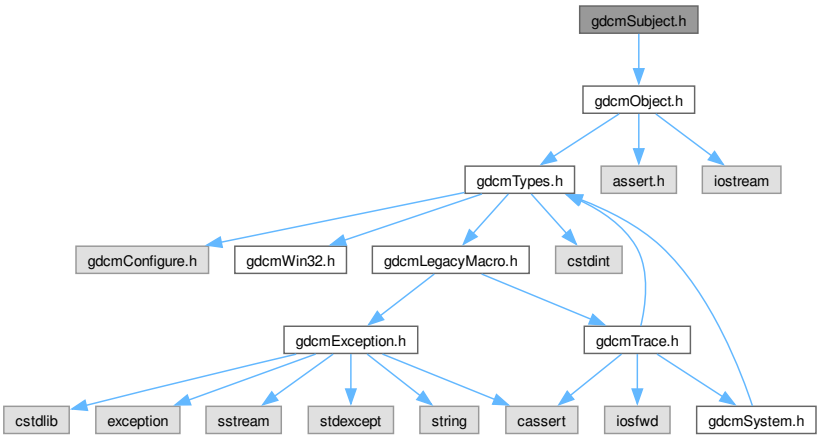
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSTRING_H
#define GDCMSTRING_H

#include "gdcmTypes.h"
#include "gdcmStaticAssert.h"

namespace gdcm
{

template <char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>
class /*GDCM_EXPORT*/ String : public std::string /* PLEASE do not export me */
{
  // UI wants \0 for pad character, while ASCII ones wants space char... do not allow anything else
  GDCM_STATIC_ASSERT( TPadChar == ' ' || TPadChar == 0 );

public:
  // typedef are not inherited:
  typedef std::string::value_type             value_type;
  typedef std::string::pointer                pointer;
  typedef std::string::reference              reference;
  typedef std::string::const_reference        const_reference;
  typedef std::string::size_type              size_type;
  typedef std::string::difference_type        difference_type;
  typedef std::string::iterator               iterator;
  typedef std::string::const_iterator         const_iterator;
  typedef std::string::reverse_iterator       reverse_iterator;
  typedef std::string::const_reverse_iterator const_reverse_iterator;

  String(): std::string() {}
  String(const value_type* s): std::string(s)
  {
  if( size() % 2 )
    {
    push_back( TPadChar );
    }
  }
  String(const value_type* s, size_type n): std::string(s, n)
  {
  // We are being passed a const char* pointer, so s[n] == 0 (guaranteed!)
  if( n % 2 )
    {
    push_back( TPadChar );
    }
  }
  String(const std::string& s, size_type pos=0, size_type n=npos):
    std::string(s, pos, n)
  {
  // FIXME: some users might already have padded the string 's' with a trailing \0...
  if( size() % 2 )
    {
    push_back( TPadChar );
    }
  }

  operator const char *() const { return this->c_str(); }

  bool IsValid() const {
    // Check Length:
    size_type l = size();
    if( l > TMaxLength ) return false;
    return true;
  }
```

```
  gdcm::String<TDelimiter, TMaxLength, TPadChar> Truncate() const {
    if( IsValid() ) return *this;
    std::string str = *this; // copy
    str.resize( TMaxLength );
    return str;
  }

  std::string Trim() const {
    std::string str = *this; // copy
    std::string::size_type pos1 = str.find_first_not_of(' ');
    std::string::size_type pos2 = str.find_last_not_of(' ');
    str = str.substr( (pos1 == std::string::npos) ? 0 : pos1,
      (pos2 == std::string::npos) ? (str.size() - 1) : (pos2 - pos1 + 1));
    return str;
  }

  static std::string Trim(const char *input) {
    if( !input ) return "";
    std::string str = input;
    std::string::size_type pos1 = str.find_first_not_of(' ');
    std::string::size_type pos2 = str.find_last_not_of(' ');
    str = str.substr( (pos1 == std::string::npos) ? 0 : pos1,
      (pos2 == std::string::npos) ? (str.size() - 1) : (pos2 - pos1 + 1));
    return str;
  }
};
template <char TDelimiter, unsigned int TMaxLength, char TPadChar>
inline std::istream& operator»(std::istream &is, String<TDelimiter,TMaxLength,TPadChar> &ms)
{
  if(is)
    {
    std::getline(is, ms, TDelimiter);
    // no such thing as std::get where the delim char would be left, so I need to manually add it back...
    // hopefully this is the right thing to do (no overhead)
    if( !is.eof() ) is.putback( TDelimiter );
    }
  return is;
}
//template <char TDelimiter = EOF, unsigned int TMaxLength = 64, char TPadChar = ' '>
//String String::Trim() const
//{
//  String s;
//  return s;
//}

} // end namespace gdcm

#endif //GDCMSTRING_H
```

13.65 gdcmSubject.h File Reference

`#include "gdcmObject.h"`

Include dependency graph for gdcmSubject.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Subject

  Subject.

Namespaces

- namespace gdcm

13.66 gdcmSubject.h

Go to the documentation of this file.

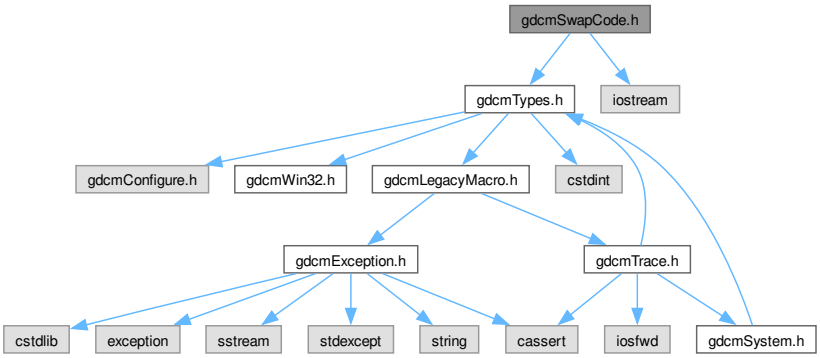
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSUBJECT_H
#define GDCMSUBJECT_H

#include "gdcmObject.h"

namespace gdcm
{
class Event;
class Command;
class SubjectInternals;
class GDCM_EXPORT Subject : public Object
{
public:
  Subject();
  ~Subject() override;

  unsigned long AddObserver(const Event & event, Command *);
  unsigned long AddObserver(const Event & event, Command *) const;

  Command* GetCommand(unsigned long tag);

  void InvokeEvent( const Event & );

  void InvokeEvent( const Event & ) const;

  void RemoveObserver(unsigned long tag);

  void RemoveAllObservers();

  bool HasObserver( const Event & event ) const;

protected:

private:
  SubjectInternals *Internals;
private:
};

} // end namespace gdcm

#endif //GDCMSUBJECT_H
```

13.67 gdcmSwapCode.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmSwapCode.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::SwapCode

  SwapCode representation.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)

13.68 gdcmSwapCode.h

Go to the documentation of this file.

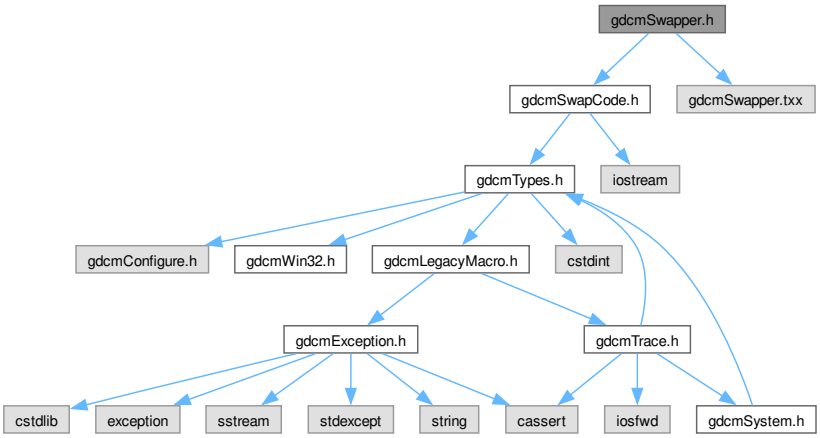
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSWAPCODE_H
#define GDCMSWAPCODE_H

#include "gdcmTypes.h"
#include <iostream>

namespace gdcm
{

class GDCM_EXPORT SwapCode
{
public:
  typedef enum {
    Unknown         = 0,
    LittleEndian    = 1234,
    BigEndian       = 4321,
    BadLittleEndian = 3412,
    BadBigEndian    = 2143
  } SwapCodeType;

  operator SwapCode::SwapCodeType() const { return SwapCodeValue; }
  SwapCode(SwapCodeType sc = Unknown):SwapCodeValue(sc) { }
  static const char* GetSwapCodeString(SwapCode const & sc);

  friend std::ostream& operator«(std::ostream& os, const SwapCode& sc);
protected:
  static int GetIndex(SwapCode const & sc);

private:
  SwapCodeType SwapCodeValue;
};
//---------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& os, const SwapCode& sc)
{
  os « SwapCode::GetSwapCodeString(sc);
  return os;
}

} // end namespace gdcm

#endif //GDCMSWAPCODE_H
```

13.69 gdcmSwapper.h File Reference

```
#include "gdcmSwapCode.h"
#include "gdcmSwapper.txx"
```

Include dependency graph for gdcmSwapper.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::SwapperDoOp
- class gdcm::SwapperNoOp

Namespaces

- namespace gdcm

13.70 gdcmSwapper.h

Go to the documentation of this file.

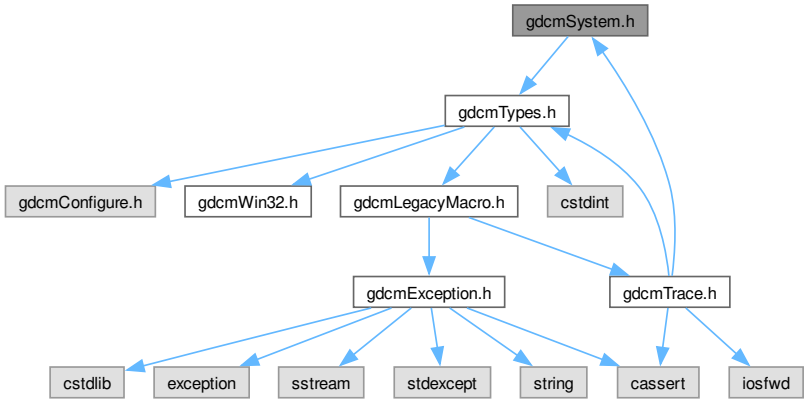
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

```

```
=========================================================================*/
#ifndef GDCMSWAPPER_H
#define GDCMSWAPPER_H

#include "gdcmSwapCode.h"

namespace gdcm
{


#ifdef GDCM_WORDS_BIGENDIAN
class SwapperDoOp
{
public:
  template <typename T> static T Swap(T val) {return val;}
  template <typename T> static void SwapArray(T *, size_t ) {}
};

class SwapperNoOp
{
public:
  template <typename T> static T Swap(T val);
  template <typename T>
  static void SwapArray(T *array, size_t n)
  {
    // TODO: need to unroll loop:
    for(size_t i = 0; i < n; ++i)
    {
      array[i] = Swap<T>(array[i]);
    }
  }
};
#else
class SwapperNoOp
{
public:
  template <typename T> static T Swap(T val) {return val;}
  template <typename T> static void SwapArray(T *, size_t ) {}
};

class SwapperDoOp
{
public:
  template <typename T> static T Swap(T val);
  template <typename T>
  static void SwapArray(T *array, size_t n)
  {
    // TODO: need to unroll loop:
    for(size_t i = 0; i < n; ++i)
    {
      array[i] = Swap<T>(array[i]);
    }
  }
};
#endif


} // end namespace gdcm

#include "gdcmSwapper.txx"

#endif //GDCMSWAPPER_H
```

13.71 gdcmSystem.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmSystem.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::System

  Class to do system operation.

Namespaces

- namespace gdcm

13.72 gdcmSystem.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

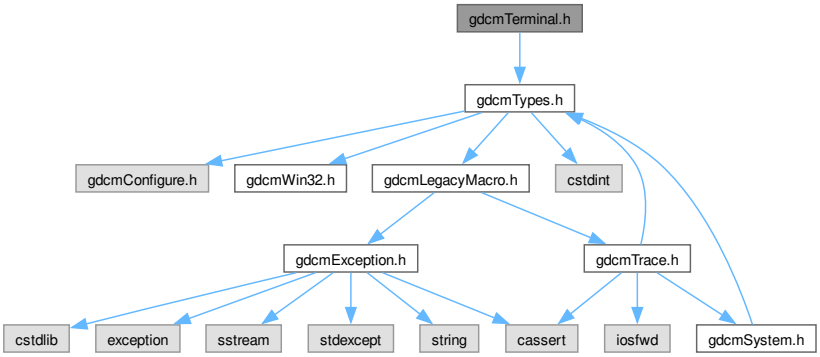
```
     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSYSTEM_H
#define GDCMSYSTEM_H

#include "gdcmTypes.h"

namespace gdcm
{

class GDCM_EXPORT System
{
public:
  static bool MakeDirectory(const char *path);
  static bool FileExists(const char* filename);
  static bool FileIsDirectory(const char* name);
  static bool FileIsSymlink(const char* name);
  static bool RemoveFile(const char* source);
  static bool DeleteDirectory(const char *source);

  static std::wstring ConvertToUNC(const char *utf8path);

  static const char *GetLastSystemError();

  static size_t FileSize(const char* filename);

  static time_t FileTime(const char* filename);

  static const char *GetCurrentProcessFileName();

  static const char *GetCurrentModuleFileName();

  static const char *GetCurrentResourcesDirectory();

  // TODO some system calls
  // Chdir
  // copy a file

  static bool GetHostName(char hostname[255]);

  // In the following the size '22' is explicitly listed. You need to pass in
  // at least 22bytes of array. If the string is an output it will be
  // automatically padded ( array[21] == 0 ) for you.
  // Those functions: GetCurrentDateTime / FormatDateTime / ParseDateTime do
  // not return the &YYZZ part of the DT structure as defined in DICOM PS 3.5 -
  // 2008 In this case it is simple to split the date[22] into a DA and TM
  // structure

  static bool GetCurrentDateTime(char date[22]);

  static bool FormatDateTime(char date[22], time_t t, long milliseconds = 0);

  static bool ParseDateTime(time_t &timep, const char date[22]);

  static bool ParseDateTime(time_t &timep, long &milliseconds, const char date[22]);

  static const char *GetTimezoneOffsetFromUTC();

  static size_t EncodeBytes(char *out, const unsigned char *data, int size);

  static int StrCaseCmp(const char *s1, const char *s2);
  static int StrNCaseCmp(const char *s1, const char *s2, size_t n);

  static const char * GetCWD();

  static char *StrTokR(char *ptr, const char *sep, char **end);

  static char *StrSep(char **stringp, const char *delim);

  static const char *GetLocaleCharset();

/*
  static void SetArgv0(const char *);
  static const char* GetArgv0();
*/

protected:
  static bool GetPermissions(const char* file, unsigned short& mode);
```

```
00144   static bool SetPermissions(const char* file, unsigned short mode);
00145
00146 private:
00147 };
00148
00149 } // end namespace gdcm
00150
00151 #endif //GDCMSYSTEM_H
```

13.73 gdcmTerminal.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmTerminal.h:



Namespaces

- namespace gdcm
- namespace gdcm::terminal

  Class for Terminal.

Enumerations

- enum gdcm::terminal::Attribute {
  gdcm::terminal::reset = 0 ,
  gdcm::terminal::bright = 1 ,
  gdcm::terminal::dim = 2 ,
  gdcm::terminal::underline = 3 ,
  gdcm::terminal::blink = 5 ,
  gdcm::terminal::reverse = 7 ,
  gdcm::terminal::hidden = 8 }

- enum gdcm::terminal::Color {
  gdcm::terminal::black = 0 ,
  gdcm::terminal::red ,
  gdcm::terminal::green ,
  gdcm::terminal::yellow ,
  gdcm::terminal::blue ,
  gdcm::terminal::magenta ,
  gdcm::terminal::cyan ,
  gdcm::terminal::white }
- enum gdcm::terminal::Mode {
  gdcm::terminal::CONSOLE = 0 ,
  gdcm::terminal::VT100 }

Functions

- GDCM_EXPORT std::string gdcm::terminal::setattribute (Attribute att)
- GDCM_EXPORT std::string gdcm::terminal::setbgcolor (Color c)
- GDCM_EXPORT std::string gdcm::terminal::setfgcolor (Color c)
- GDCM_EXPORT void gdcm::terminal::setmode (Mode m)

13.74 gdcmTerminal.h

Go to the documentation of this file.

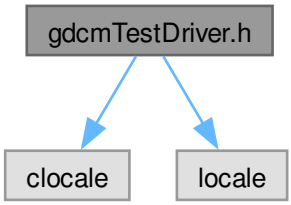
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTERMINAL_H
#define GDCMTERMINAL_H

#include "gdcmTypes.h"


namespace gdcm
{
//-----------------------------------------------------------------------------

namespace terminal
{
  typedef enum
    {
    CONSOLE = 0,
    VT100
    } Mode;
  typedef enum
    {
    black = 0,
    red,
    green,
    yellow, // brown ??
    blue,
    magenta,
    cyan,
    white
```

```
    } Color;
  typedef enum
    {
    reset     = 0,
    bright    = 1, // bold
    dim       = 2,
    underline = 3,
    blink     = 5,
    reverse   = 7,
    hidden    = 8
    } Attribute;
  GDCM_EXPORT std::string setattribute( Attribute att );
  GDCM_EXPORT std::string setfgcolor( Color c );
  GDCM_EXPORT std::string setbgcolor( Color c );
  GDCM_EXPORT void setmode( Mode m);
}

} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMTERMINAL_H
```

13.75 gdcmTestDriver.h File Reference

```
#include <clocale>
#include <locale>
```

Include dependency graph for gdcmTestDriver.h:



13.76 gdcmTestDriver.h

Go to the documentation of this file.

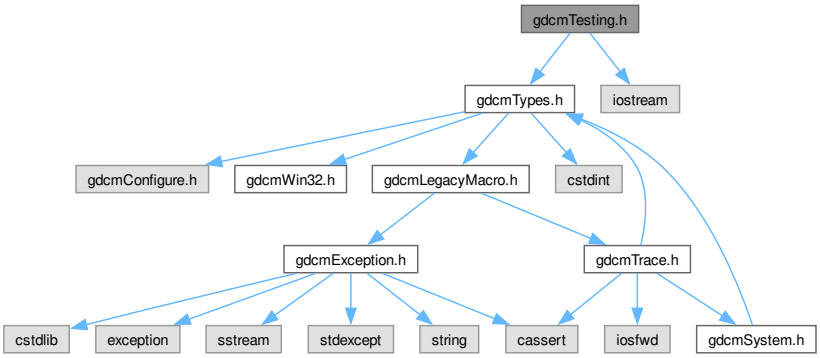
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This header is included by all the C++ test drivers in GDCM.
#ifndef GDCMTESTDRIVER_H
```

```
#define GDCMTESTDRIVER_H

// CREATE_TEST_SOURCELIST supports the flag EXTRA_INCLUDE but only one per call.
// So there is no way to specify we want to include two files... instead
// gather the #include in a single file and include that one...
#include <clocale> // C setlocale()
#include <locale> // C++ locale

#endif // GDCMTESTDRIVER_H
```

13.77 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



Classes

- class gdcm::Testing

  class for testing

Namespaces

- namespace gdcm

13.78 gdcmTesting.h

Go to the documentation of this file.

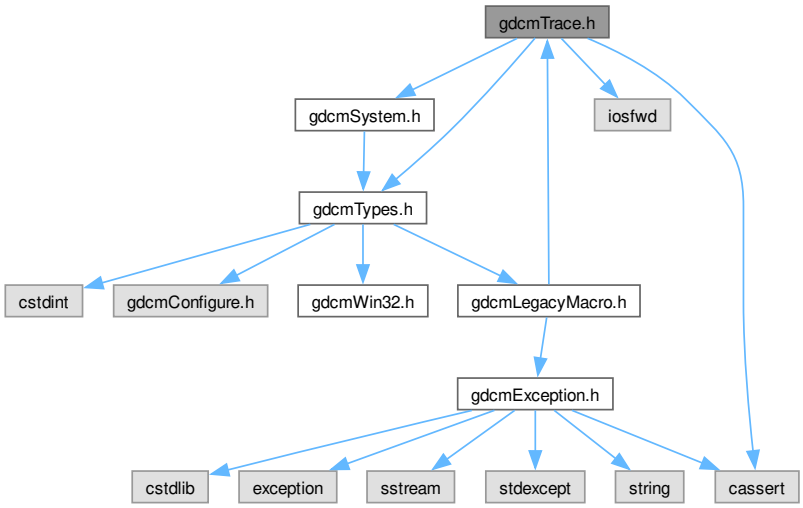
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
```

```
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTESTING_H
#define GDCMTESTING_H

#include "gdcmTypes.h"

#include <iostream>

namespace gdcm
{
//-----------------------------------------------------------------------------
class GDCM_EXPORT Testing
{
public :
  Testing() = default;
  ~Testing() = default;

  static bool ComputeMD5(const char *buffer, size_t buf_len,
    char digest_str[33]);
  static bool ComputeFileMD5(const char *filename, char digest_str[33]);

  void Print(std::ostream &os = std::cout);

  static const char * const * GetFileNames();
  static unsigned int GetNumberOfFileNames();
  static const char * GetFileName(unsigned int file);

  typedef const char* const (*MediaStorageDataFilesType)[2];
  static MediaStorageDataFilesType GetMediaStorageDataFiles();
  static unsigned int GetNumberOfMediaStorageDataFiles();
  static const char * const * GetMediaStorageDataFile(unsigned int file);
  static const char * GetMediaStorageFromFile(const char *filepath);

  typedef const char* const (*MD5DataImagesType)[2];
  static MD5DataImagesType GetMD5DataImages();
  static unsigned int GetNumberOfMD5DataImages();
  static const char * const * GetMD5DataImage(unsigned int file);
  static const char * GetMD5FromFile(const char *filepath);

  static const char * GetMD5FromBrokenFile(const char *filepath);

  static std::streamoff GetStreamOffsetFromFile(const char *filepath);

  static std::streamoff GetSelectedTagsOffsetFromFile(const char *filepath);

  static std::streamoff GetSelectedPrivateGroupOffsetFromFile(const char *filepath);

  static int GetLossyFlagFromFile(const char *filepath);

  static const char * GetDataRoot();

  static const char * GetDataExtraRoot();

  static const char * GetPixelSpacingDataRoot();

  static const char * GetTempDirectory(const char * subdir = nullptr);

  static const wchar_t *GetTempDirectoryW(const wchar_t * subdir = nullptr);

  static const char * GetTempFilename(const char *filename, const char * subdir = nullptr);

  static const wchar_t* GetTempFilenameW(const wchar_t *filename, const wchar_t* subdir = nullptr);

  static const char *GetSourceDirectory();
};
} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMTESTING_H
```

13.79 gdcmTrace.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmSystem.h"
#include <iosfwd>
#include <cassert>
```

Include dependency graph for gdcmTrace.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Trace

  Trace.

Namespaces

- namespace gdcm

Macros

- #define GDCM_FUNCTION "<unknown>"
- #define gdcmAssertAlwaysMacro(arg)

  AssertAlways.
- #define gdcmAssertMacro(arg)

  Assert.
- #define gdcmDebugMacro(msg)

  Debug.
- #define gdcmErrorMacro(msg)

  Error this is pretty bad, more than just warning It could mean lost of data, something not handle...
- #define gdcmWarningMacro(msg)

  Warning.

13.79.1 Macro Definition Documentation

13.79.1.1 GDCM_FUNCTION

```
#define GDCM_FUNCTION "<unknown>"
```

13.79.1.2 gdcmAssertAlwaysMacro

```
#define gdcmAssertAlwaysMacro(
            arg)
```

Value:

```
gdcmAssertMacro(arg)
```

AssertAlways.

Parameters

arg	argument to test An easy solution to pass also a message is to do: gdcmAssertMacro("my message" && 2 < 3)

Referenced by gdcm::DataElement::GetValue(), gdcm::DataElement::GetValue(), gdcm::BasicOffsetTable::Read(), gdcm::SequenceOfFragments::ReadValue(), gdcm::DataSet::Replace(), gdcm::DataSet::ReplaceEmpty(), and gdcm::VR::Write().

13.79.1.3 gdcmAssertMacro

```
#define gdcmAssertMacro(
            arg)
```

Value:

```
{                                                                   \
   if( !(arg) )                                                     \
   {                                                                \
   std::ostringstream osmacro;                                      \
   osmacro « "Assert: In " __FILE__ ", line " « __LINE__          \
           « ", function " « GDCM_FUNCTION                         \
           « "\n\n";                                                \
   std::ostream &_os = gdcm::Trace::GetErrorStream();               \
   _os « osmacro.str() « std::endl;                                \
   assert ( arg );                                                  \
   }                                                                \
}                                                                   \
GDCM_NOOP_STATEMENT
```

Assert.

Parameters

arg	argument to test An easy solution to pass also a message is to do: gdcmAssertMacro("my message" && 2 < 3)

Referenced by gdcm::PixelFormat::SetSamplesPerPixel().

13.79.1.4 gdcmDebugMacro

```
#define gdcmDebugMacro(
            msg)
```

Value:

```
{                                                                   \
   if( gdcm::Trace::GetDebugFlag() )                                \
   {                                                                \
   std::ostringstream osmacro;                                      \
   osmacro « "Debug: In " __FILE__ ", line " « __LINE__           \
           « ", function " « GDCM_FUNCTION « '\n'                 \
           « "Last system error was: "                              \
           « gdcm::System::GetLastSystemError() « '\n' « msg;      \
   std::ostream &_os = gdcm::Trace::GetDebugStream();               \
   _os « osmacro.str() « "\n\n" « std::endl;                       \
   }                                                                \
}                                                                   \
GDCM_NOOP_STATEMENT
```

Debug.

Parameters

msg	message part

Referenced by gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory(), gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory(), gdcm::Item::Read(), gdcm::SequenceOfItems::Read(), gdcm::VR::Read(), gdcm::SequenceOfFragments::ReadPreValue(), and gdcm::SequenceOfFragments::ReadValue().

13.79.1.5 gdcmErrorMacro

```
#define gdcmErrorMacro(
            msg)
```

Value:

```
{                                                                   \
   if( gdcm::Trace::GetErrorFlag() )                                \
   {                                                                \
   std::ostringstream osmacro;                                      \
   osmacro « "Error: In " __FILE__ ", line " « __LINE__            \
           « ", function " « GDCM_FUNCTION « '\n'                 \
           « msg « "\n\n";                                          \
   std::ostream &_os = gdcm::Trace::GetErrorStream();               \
   _os « osmacro.str() « std::endl;                                \
   }                                                                \
}                                                                   \
GDCM_NOOP_STATEMENT
```

Error this is pretty bad, more than just warning It could mean lost of data, something not handle...

Parameters

msg	second message part

Referenced by gdcm::CommandDataSet::Insert(), gdcm::DataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::Item::Read(), and gdcm::Fragment::ReadBacktrack().

13.79.1.6 gdcmWarningMacro

```
#define gdcmWarningMacro(
            msg)
```

Value:

```
{                                                                   \
   if( gdcm::Trace::GetWarningFlag() )                              \
   {                                                                \
   std::ostringstream osmacro;                                      \
   osmacro « "Warning: In " __FILE__ ", line " « __LINE__          \
           « ", function " « GDCM_FUNCTION « "\n"                 \
           « msg « "\n\n";                                          \
   std::ostream &_os = gdcm::Trace::GetWarningStream();             \
   _os « osmacro.str() « std::endl;                                \
   }                                                                \
}                                                                   \
GDCM_NOOP_STATEMENT
```

Warning.

Parameters

msg	message part

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Item::Read(), gdcm::SequenceOfItems::Read(), gdcm::Fragment::ReadBacktr
gdcm::Fragment::ReadValue(), gdcm::SequenceOfFragments::ReadValue(), gdcm::OpenSSLP7CryptographicMessageSyntax::SetPassw
and gdcm::Item::Write().

13.80 gdcmTrace.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTRACE_H
#define GDCMTRACE_H

#include "gdcmTypes.h"
#include "gdcmSystem.h"

#include <iosfwd>
#include <cassert>

namespace gdcm
{

class GDCM_EXPORT Trace
{
public :
  Trace();
  ~Trace();

  static void SetStream(std::ostream &os);
  static std::ostream &GetStream();

  static void SetDebugStream(std::ostream &os);
  static std::ostream &GetDebugStream();

  static void SetWarningStream(std::ostream &os);
  static std::ostream &GetWarningStream();

  static void SetErrorStream(std::ostream &os);
  static std::ostream &GetErrorStream();

  static void SetStreamToFile( const char *filename );

  static void SetDebug(bool debug);
  static void DebugOn();
  static void DebugOff();
  static bool GetDebugFlag();

  static void SetWarning(bool debug);
  static void WarningOn();
  static void WarningOff();
  static bool GetWarningFlag();

  static void SetError(bool debug);
  static void ErrorOn();
  static void ErrorOff();
  static bool GetErrorFlag();

protected:
private:
};

// Here we define function this is the only way to be able to pass
// stuff with indirection like:
// gdcmDebug( "my message:" « i « '\t' );
// You cannot use function unless you use vnsprintf ...

// __FUNCTION is not always defined by preprocessor
// In c++ we should use __PRETTY_FUNCTION__ instead...
#ifdef GDCM_CXX_HAS_FUNCTION
// Handle particular case for GNU C++ which also defines __PRETTY_FUNCTION__
// which is a lot nice in C++
#ifdef __BORLANDC__
```

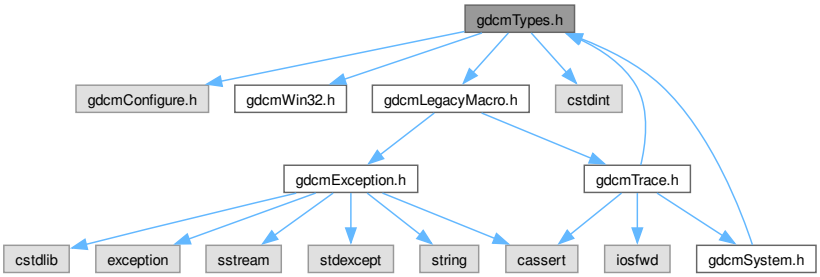
```
#  define __FUNCTION__ __FUNC__
#endif
#ifdef __GNUC__
#  define GDCM_FUNCTION __PRETTY_FUNCTION__
#else
#  define GDCM_FUNCTION __FUNCTION__
#endif //__GNUC__
#else
#  define GDCM_FUNCTION "<unknown>"
#endif //GDCM_CXX_HAS_FUNCTION

#if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
#define gdcmDebugMacro(msg) GDCM_NOOP_STATEMENT
#else
#define gdcmDebugMacro(msg)                                          \
{                                                                    \
   if( gdcm::Trace::GetDebugFlag() )                                 \
   {                                                                 \
   std::ostringstream osmacro;                                       \
   osmacro « "Debug: In " __FILE__ ", line " « __LINE__          \
           « ", function " « GDCM_FUNCTION « '\n'                \
           « "Last system error was: "                             \
           « gdcm::System::GetLastSystemError() « '\n' « msg;  \
   std::ostream &_os = gdcm::Trace::GetDebugStream();                \
   _os « osmacro.str() « "\n\n" « std::endl;                     \
   }                                                                 \
}                                                                    \
GDCM_NOOP_STATEMENT
#endif //NDEBUG

#if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
#define gdcmWarningMacro(msg) GDCM_NOOP_STATEMENT
#else
#define gdcmWarningMacro(msg)                                        \
{                                                                    \
   if( gdcm::Trace::GetWarningFlag() )                               \
   {                                                                 \
   std::ostringstream osmacro;                                       \
   osmacro « "Warning: In " __FILE__ ", line " « __LINE__        \
           « ", function " « GDCM_FUNCTION « "\n"               \
           « msg « "\n\n";                                        \
   std::ostream &_os = gdcm::Trace::GetWarningStream();              \
   _os « osmacro.str() « std::endl;                              \
   }                                                                 \
}                                                                    \
GDCM_NOOP_STATEMENT
#endif //NDEBUG

#if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
#define gdcmErrorMacro(msg) GDCM_NOOP_STATEMENT
#else
#define gdcmErrorMacro(msg)                                          \
{                                                                    \
   if( gdcm::Trace::GetErrorFlag() )                                 \
   {                                                                 \
   std::ostringstream osmacro;                                       \
   osmacro « "Error: In " __FILE__ ", line " « __LINE__          \
           « ", function " « GDCM_FUNCTION « '\n'                \
           « msg « "\n\n";                                         \
   std::ostream &_os = gdcm::Trace::GetErrorStream();                \
   _os « osmacro.str() « std::endl;                              \
   }                                                                 \
}                                                                    \
GDCM_NOOP_STATEMENT
#endif //NDEBUG

#if defined(NDEBUG) && !defined(GDCM_ALWAYS_TRACE_MACRO)
#define gdcmAssertMacro(arg) GDCM_NOOP_STATEMENT
#else
#define gdcmAssertMacro(arg)                                         \
{                                                                    \
   if( !(arg) )                                                      \
   {                                                                 \
   std::ostringstream osmacro;                                       \
   osmacro « "Assert: In " __FILE__ ", line " « __LINE__         \
           « ", function " « GDCM_FUNCTION                       \
           « "\n\n";                                               \
   std::ostream &_os = gdcm::Trace::GetErrorStream();                \
   _os « osmacro.str() « std::endl;                              \
   assert ( arg );                                                   \
   }                                                                 \
```

```
}                                                                    \
GDCM_NOOP_STATEMENT
#endif //NDEBUG

#if defined(NDEBUG)
// User asked for release compilation, but still need to report
// if grave issue.
#define gdcmAssertAlwaysMacro(arg) \
{                                                                    \
   if( !(arg) )                                                      \
   {                                                                 \
   std::ostringstream osmacro;                                       \
   osmacro « "Assert: In " __FILE__ ", line " « __LINE__        \
           « ", function " « GDCM_FUNCTION                       \
           « "\n\n";                                              \
   throw osmacro.str();                                              \
   }                                                                 \
}                                                                    \
GDCM_NOOP_STATEMENT
#else
// Simply reproduce gdcmAssertMacro behavior:
#define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(arg)
#endif //NDEBUG

} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMTRACE_H
```

13.81 gdcmTypes.h File Reference

```
#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
#include <cstdint>
```

Include dependency graph for gdcmTypes.h:



13.82 gdcmTypes.h

Go to the documentation of this file.

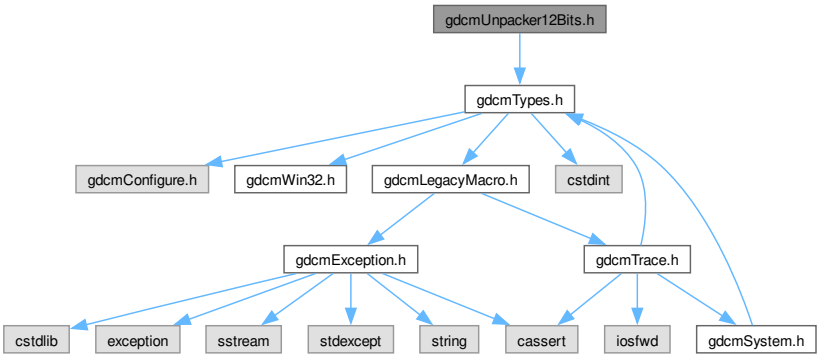
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
```

```
 See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTYPES_H
#define GDCMTYPES_H

#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"

//-----------------------------------------------------------------------------
#include <cstdint>

//-----------------------------------------------------------------------------
#endif //GDCMTYPES_H
```

13.83 gdcmUnpacker12Bits.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmUnpacker12Bits.h:



Classes

- class gdcm::Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.

Namespaces

- namespace gdcm

13.84 gdcmUnpacker12Bits.h

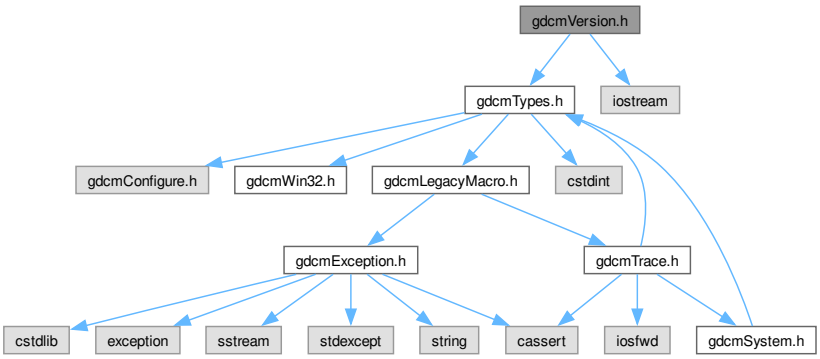
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMUNPACKER12BITS_H
#define GDCMUNPACKER12BITS_H

#include "gdcmTypes.h"

namespace gdcm
{
class GDCM_EXPORT Unpacker12Bits
{
public:
  static bool Pack(char *out, const char *in, size_t n);

  static bool Unpack(char *out, const char *in, size_t n);
};

} // end namespace gdcm

#endif //GDCMUNPACKER12BITS_H
```

13.85 gdcmVersion.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmVersion.h:



Classes

- class gdcm::Version

  major/minor and build version

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)

13.86 gdcmVersion.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMVERSION_H
#define GDCMVERSION_H

#include "gdcmTypes.h"
#include <iostream>

namespace gdcm
{
//-----------------------------------------------------------------------------
class GDCM_EXPORT Version
{
  friend std::ostream& operator«(std::ostream &_os, const Version &v);
public :
  static const char *GetVersion();
  static int GetMajorVersion();
  static int GetMinorVersion();
  static int GetBuildVersion();

  void Print(std::ostream &os = std::cout) const;

//protected:
  Version() = default;
  ~Version() = default;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Version &v)
{
  v.Print( os );
  return os;
}

} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMVERSION_H
```

13.87 gdcmWin32.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define GDCM_EXPORT

13.87.1 Macro Definition Documentation

13.87.1.1 GDCM_EXPORT

```
#define GDCM_EXPORT
```

Referenced by gdcm::terminal::setattribute(), gdcm::terminal::setbgcolor(), gdcm::terminal::setfgcolor(), and gdcm::terminal::setmode().

13.88 gdcmWin32.h

Go to the documentation of this file.

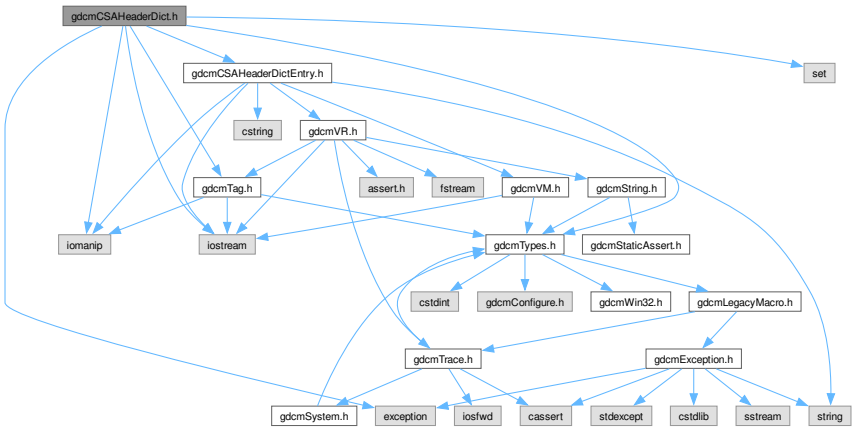
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMWIN32_H
#define GDCMWIN32_H

#if !defined(GDCMTYPES_H)
#error you need to include gdcmTypes.h instead
#endif
//-----------------------------------------------------------------------------
// http://gcc.gnu.org/wiki/Visibility
#if defined(_WIN32) && defined(GDCM_BUILD_SHARED_LIBS)
  #if (defined(gdcmCommon_EXPORTS) || defined(gdcmDICT_EXPORTS) || defined(gdcmDSED_EXPORTS) ||
      defined(gdcmIOD_EXPORTS) || defined(gdcmMSFF_EXPORTS) || defined(gdcmMEXD_EXPORTS)||
      defined(_gdcmswig_EXPORTS)) || defined(vtkgdcm_EXPORTS)
    #define GDCM_EXPORT __declspec( dllexport )
  #else
    #define GDCM_EXPORT __declspec( dllimport )
  #endif
#else
  #if __GNUC__ >= 4 && defined(GDCM_BUILD_SHARED_LIBS)
    #define GDCM_EXPORT __attribute__ ((visibility ("default")))
    #define GDCM_LOCAL  __attribute__ ((visibility ("hidden")))
  #else
    #define GDCM_EXPORT
  #endif
#endif

#if defined(GDCM_OVERRIDE_BROKEN_IMPLEMENTATION) && !defined(GDCM_FORCE_EXPORT)
#undef GDCM_EXPORT
#define GDCM_EXPORT
#endif

// In VTK 4.2 vtkWrapPython does not like anything other than VTK_*EXPORT
// [ 86%] Generating vtkGDCMImageReaderPython.cxx
// syntax error
```

```
// *** SYNTAX ERROR found in parsing the header file
/usr/local/src/gdcm2/tags/gdcm-2-0-11/Utilities/VTK/vtkGDCMImageReader.h before line 128***
// make[2]: *** [Utilities/VTK/vtkGDCMImageReaderPython.cxx] Error 1
// make[1]: *** [Utilities/VTK/CMakeFiles/vtkgdcmPythonD.dir/all] Error 2
// make: *** [all] Error 2

#if defined(VTK_MAJOR_VERSION) && ( VTK_MAJOR_VERSION == 4 )
#undef VTK_EXPORT
#define VTK_EXPORT GDCM_EXPORT
#endif

//-----------------------------------------------------------------------------
//This is needed when compiling in debug mode
#ifdef _MSC_VER
// to allow construct such as: std::numeric_limits<int>::max() we need the following:
// warning C4003: not enough actual parameters for macro 'max'
#ifndef NOMINMAX
  #define NOMINMAX
#endif
# pragma warning ( default : 4263 ) /* no override, call convention differs */
// 'identifier' : class 'type' needs to have dll-interface to be used by
// clients of class 'type2'
#pragma warning ( disable : 4251 )
// non dll-interface class 'type' used as base for dll-interface class 'type2'
#pragma warning ( disable : 4275 )
// 'identifier' : identifier was truncated to 'number' characters in the
// debug information
#pragma warning ( disable : 4786 )
//'identifier' : decorated name length exceeded, name was truncated
#pragma warning ( disable : 4503 )
#endif //_MSC_VER

//-----------------------------------------------------------------------------
#endif //GDCMWIN32_H
```

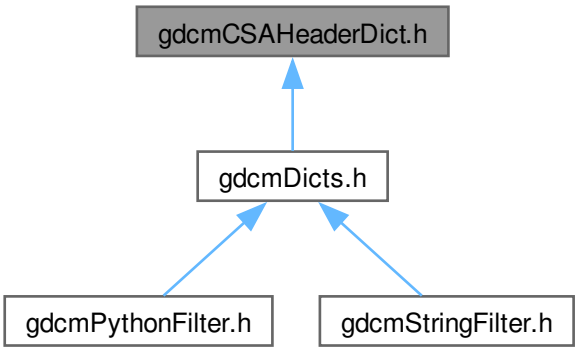
13.89 gdcmCSAHeaderDict.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmCSAHeaderDictEntry.h"
#include <iostream>
#include <iomanip>
#include <set>
#include <exception>
```

Include dependency graph for gdcmCSAHeaderDict.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CSAHeaderDict

  Class to represent a map of CSAHeaderDictEntry.
- class gdcm::CSAHeaderDictException

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)

13.90 gdcmCSAHeaderDict.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCSAHEADERDICT_H
```

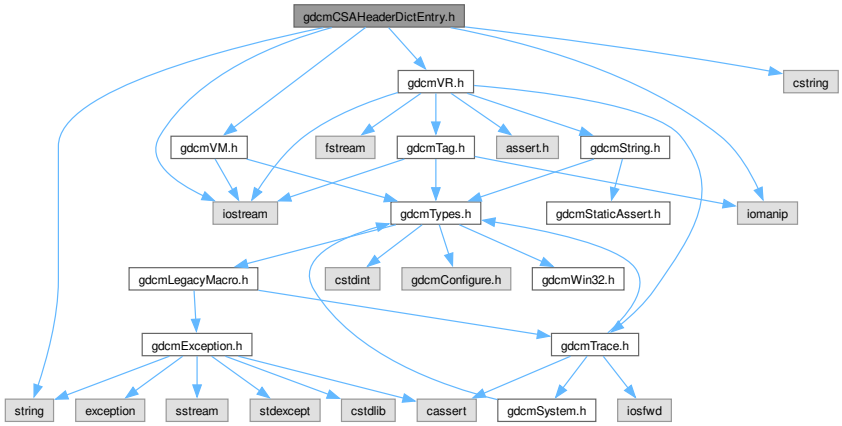
```
#define GDCMCSAHEADERDICT_H

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmCSAHeaderDictEntry.h"

#include <iostream>
#include <iomanip>
#include <set>
#include <exception>

namespace gdcm
{

class GDCM_EXPORT CSAHeaderDictException : public std::exception {};

class GDCM_EXPORT CSAHeaderDict
{
public:
  typedef std::set<CSAHeaderDictEntry> MapCSAHeaderDictEntry;
  typedef MapCSAHeaderDictEntry::iterator Iterator;
  typedef MapCSAHeaderDictEntry::const_iterator ConstIterator;
  //static CSAHeaderDictEntry GroupLengthCSAHeaderDictEntry; // = CSAHeaderDictEntry("Group Length",VR::UL,VM::VM1);

  CSAHeaderDict():CSAHeaderDictInternal() {
    assert( CSAHeaderDictInternal.empty() );
  }
  CSAHeaderDict &operator=(const CSAHeaderDict &_val) = delete;
  CSAHeaderDict(const CSAHeaderDict &_val) = delete;

  friend std::ostream& operator«(std::ostream& _os, const CSAHeaderDict &_val);

  ConstIterator Begin() const { return CSAHeaderDictInternal.begin(); }
  ConstIterator End() const { return CSAHeaderDictInternal.end(); }

  bool IsEmpty() const { return CSAHeaderDictInternal.empty(); }
  void AddCSAHeaderDictEntry(const CSAHeaderDictEntry &de)
    {
#ifndef NDEBUG
    MapCSAHeaderDictEntry::size_type s = CSAHeaderDictInternal.size();
#endif
    CSAHeaderDictInternal.insert( de );
    assert( s < CSAHeaderDictInternal.size() );
    }

  const CSAHeaderDictEntry &GetCSAHeaderDictEntry(const char *name) const
    {
    MapCSAHeaderDictEntry::const_iterator it = CSAHeaderDictInternal.find( name );
    if( it != CSAHeaderDictInternal.end() )
      {
      return *it;
      }
    throw CSAHeaderDictException();
    }

protected:
  friend class Dicts;
  void LoadDefault();

private:

  MapCSAHeaderDictEntry CSAHeaderDictInternal;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& os, const CSAHeaderDict &val)
{
  CSAHeaderDict::MapCSAHeaderDictEntry::const_iterator it = val.CSAHeaderDictInternal.begin();
  for(;it != val.CSAHeaderDictInternal.end(); ++it)
    {
    const CSAHeaderDictEntry &de = *it;
    os « de « '\n';
    }


  return os;
}


} // end namespace gdcm

```

```
00098 #endif //GDCMCSAHEADERDICT_H
```

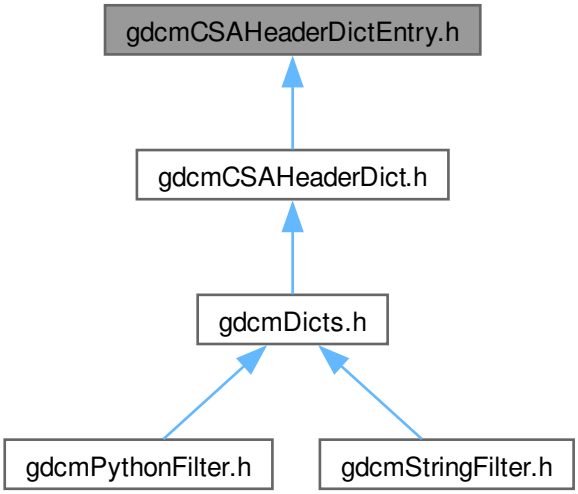
13.91 gdcmCSAHeaderDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
#include <cstring>
```

Include dependency graph for gdcmCSAHeaderDictEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CSAHeaderDictEntry

  Class to represent an Entry in the Dict.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)

13.92 gdcmCSAHeaderDictEntry.h

Go to the documentation of this file.

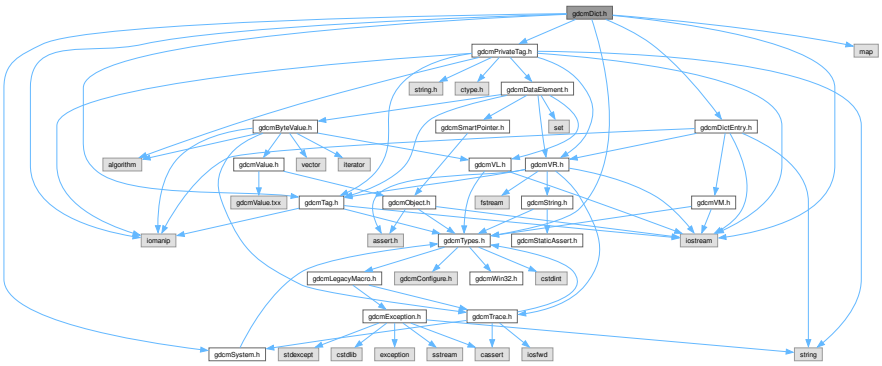
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

```
     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCSAHEADERDICTENTRY_H
#define GDCMCSAHEADERDICTENTRY_H

#include "gdcmVR.h"
#include "gdcmVM.h"

#include <string>
#include <iostream>
#include <iomanip>

#include <cstring>

namespace gdcm
{
class GDCM_EXPORT CSAHeaderDictEntry
{
public:
  CSAHeaderDictEntry(const char *name = "", VR const &vr = VR::INVALID, VM const &vm = VM::VM0, const char *desc = ""):Name(name),ValueRepresentation(vr),ValueMultiplicity(vm),Description(desc) {
  }

  friend std::ostream& operator«(std::ostream& _os, const CSAHeaderDictEntry &_val);

  const VR &GetVR() const { return ValueRepresentation; }
  void SetVR(const VR & vr) { ValueRepresentation = vr; }

  const VM &GetVM() const { return ValueMultiplicity; }
  void SetVM(VM const & vm) { ValueMultiplicity = vm; }

  const char *GetName() const { return Name.c_str(); }
  void SetName(const char* name) { Name = name; }

  const char *GetDescription() const { return Description.c_str(); }
  void SetDescription(const char* desc) { Description = desc; }

  bool operator<(const CSAHeaderDictEntry &entry) const
    {
    return strcmp(GetName(),entry.GetName()) < 0;
    }

private:
  std::string Name;
  VR ValueRepresentation;
  VM ValueMultiplicity;
  std::string Description;
  std::string Type; // TODO
};


//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& os, const CSAHeaderDictEntry &val)
{
  if( val.Name.empty() )
    {
    os « "[No name]";
    }
  else
    {
    os « val.Name;
    }
  os « "\t" « val.ValueRepresentation « "\t" « val.ValueMultiplicity;
  if( !val.Description.empty() )
    {
    os « "\t" « val.Description;
    }
  return os;
}

} // end namespace gdcm

#endif //GDCMCSAHEADERDICTENTRY_H
```

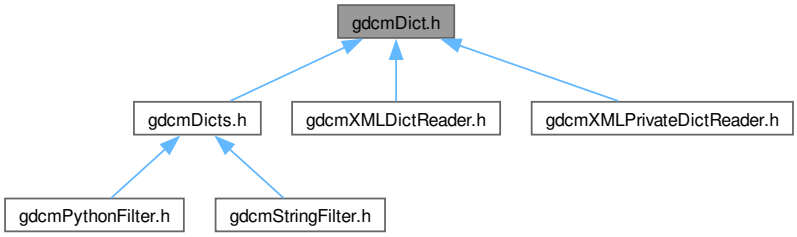
13.93 gdcmDict.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmDictEntry.h"
#include "gdcmSystem.h"
#include <iostream>
#include <iomanip>
#include <map>
```

Include dependency graph for gdcmDict.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Dict

  Class to represent a map of DictEntry.
- class gdcm::PrivateDict

  Private Dict.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)
- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)

13.94 gdcmDict.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDICT_H
#define GDCMDICT_H

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmDictEntry.h"
#include "gdcmSystem.h"

#include <iostream>
#include <iomanip>
#include <map>

/*
 * FIXME / TODO
 * I need to seriously rewrite this mess. a class template should work for both a public
 * and a private dict
 */

namespace gdcm
{
// Data Element Tag
class GDCM_EXPORT Dict
{
public:
  typedef std::map<Tag, DictEntry> MapDictEntry;
  typedef MapDictEntry::iterator Iterator;
  typedef MapDictEntry::const_iterator ConstIterator;
  //static DictEntry GroupLengthDictEntry; // = DictEntry("Group Length",VR::UL,VM::VM1);

  Dict():DictInternal() {
    assert( DictInternal.empty() );
  }
  Dict &operator=(const Dict &_val) = delete;
  Dict(const Dict &_val) = delete;


  friend std::ostream& operator«(std::ostream& _os, const Dict &_val);

  ConstIterator Begin() const { return DictInternal.begin(); }
  ConstIterator End() const { return DictInternal.end(); }

  bool IsEmpty() const { return DictInternal.empty(); }
  void AddDictEntry(const Tag &tag, const DictEntry &de)
```

```
    {
#ifndef NDEBUG
    MapDictEntry::size_type s = DictInternal.size();
#endif
    DictInternal.insert(
      MapDictEntry::value_type(tag, de));
    assert( s < DictInternal.size() );
    }

  const DictEntry &GetDictEntry(const Tag &tag) const
    {
    MapDictEntry::const_iterator it =
      DictInternal.find(tag);
    if (it == DictInternal.end())
      {
#ifdef UNKNOWNPUBLICTAG
      // test.acr
      if( tag != Tag(0x28,0x15)
        && tag != Tag(0x28,0x16)
        && tag != Tag(0x28,0x199)
        // gdcmData/TheralysGDCM1.dcm
        && tag != Tag(0x20,0x1)
        // gdcmData/0019004_Baseline_IMG1.dcm
        && tag != Tag(0x8348,0x339)
        && tag != Tag(0xb5e8,0x338)
        // gdcmData/dicomdir_Acusson_WithPrivate_WithSR
        && tag != Tag(0x40,0xa125)
      )
        {
        assert( 0 && "Impossible" );
        }
#endif
      it = DictInternal.find( Tag(0xffff,0xffff) );
      return it->second;
      }
    assert( DictInternal.count(tag) == 1 );
    return it->second;
    }

  const char *GetKeywordFromTag(Tag const & tag) const
    {
    MapDictEntry::const_iterator it =
      DictInternal.find(tag);
    if (it == DictInternal.end())
      {
      return nullptr;
      }
    assert( DictInternal.count(tag) == 1 );
    return it->second.GetKeyword();
    }

  const DictEntry &GetDictEntryByKeyword(const char *keyword, Tag & tag) const
    {
    MapDictEntry::const_iterator it =
      DictInternal.begin();
    if( keyword )
      {
      for(; it != DictInternal.end(); ++it)
        {
        if( strcmp( keyword, it->second.GetKeyword() ) == 0 )
          {
          // Found a match !
          tag = it->first;
          break;
          }
        }
      }
    else
      {
      it = DictInternal.end();
      }
    if (it == DictInternal.end())
      {
      tag = Tag(0xffff,0xffff);
      it = DictInternal.find( tag );
      return it->second;
      }
    assert( DictInternal.count(tag) == 1 );
    return it->second;
    }

```

```
  const DictEntry &GetDictEntryByName(const char *name, Tag & tag) const
    {
    MapDictEntry::const_iterator it =
      DictInternal.begin();
    if( name )
      {
      for(; it != DictInternal.end(); ++it)
        {
        if( strcmp( name, it->second.GetName() ) == 0 )
          {
          // Found a match !
          tag = it->first;
          break;
          }
        }
      }
    else
      {
      it = DictInternal.end();
      }
    if (it == DictInternal.end())
      {
      tag = Tag(0xffff,0xffff);
      it = DictInternal.find( tag );
      return it->second;
      }
    assert( DictInternal.count(tag) == 1 );
    return it->second;
    }

protected:
  friend class Dicts;
  void LoadDefault();

private:
  MapDictEntry DictInternal;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& os, const Dict &val)
{
  Dict::MapDictEntry::const_iterator it = val.DictInternal.begin();
  for(;it != val.DictInternal.end(); ++it)
    {
    const Tag &t = it->first;
    const DictEntry &de = it->second;
    os « t « " " « de « '\n';
    }

  return os;
}

// TODO
// For private dict, element < 0x10 should automatically defined:
// Name = "Private Creator"
// ValueRepresentation = LO
// ValueMultiplicity = 1
// Owner = ""

class GDCM_EXPORT PrivateDict
{
  typedef std::map<PrivateTag, DictEntry> MapDictEntry;
  friend std::ostream& operator«(std::ostream& os, const PrivateDict &val);
public:
  PrivateDict() = default;
  ~PrivateDict() = default;
  void AddDictEntry(const PrivateTag &tag, const DictEntry &de)
    {
#ifndef NDEBUG
    MapDictEntry::size_type s = DictInternal.size();
#endif
    DictInternal.insert(
      MapDictEntry::value_type(tag, de));
// The following code should only be used when manually constructing a Private.xml file by hand
// it will get rid of VR::UN duplicate (ie. if a VR != VR::Un can be found)
#if defined(NDEBUG) && 0
    if( s == DictInternal.size() )
      {
      MapDictEntry::iterator it =
        DictInternal.find(tag);
      assert( it != DictInternal.end() );
      DictEntry &duplicate = it->second;
```

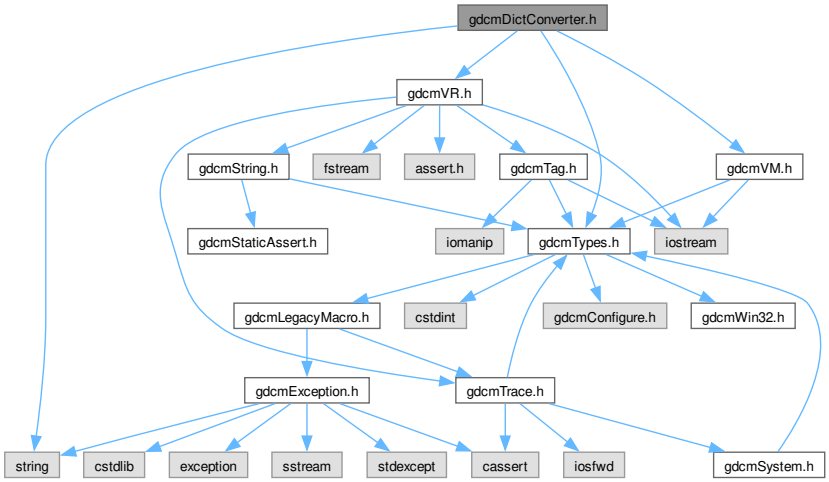
```
      assert( de.GetVR() == VR::UN || duplicate.GetVR() == VR::UN );
      assert( de.GetVR() != duplicate.GetVR() );
      if( duplicate.GetVR() == VR::UN )
        {
        assert( de.GetVR() != VR::UN );
        duplicate.SetVR( de.GetVR() );
        duplicate.SetVM( de.GetVM() );
        assert( GetDictEntry(tag).GetVR() != VR::UN );
        assert( GetDictEntry(tag).GetVR() == de.GetVR() );
        assert( GetDictEntry(tag).GetVM() == de.GetVM() );
        }
      return;
      }
#endif
    assert( s < DictInternal.size() /*&& std::cout « tag « "," « de « std::endl*/ );
    }
  bool RemoveDictEntry(const PrivateTag &tag)
    {
    MapDictEntry::size_type s =
      DictInternal.erase(tag);
    assert( s == 1 || s == 0 );
    return s == 1;
    }
  bool FindDictEntry(const PrivateTag &tag) const
    {
    MapDictEntry::const_iterator it =
      DictInternal.find(tag);
    if (it == DictInternal.end())
      {
      return false;
      }
    return true;
    }
  const DictEntry &GetDictEntry(const PrivateTag &tag) const
    {
    // if 0x10 -> return Private Creator
    MapDictEntry::const_iterator it =
      DictInternal.find(tag);
    if (it == DictInternal.end())
      {
      //assert( 0 && "Impossible" );
      it = DictInternal.find( PrivateTag(0xffff,0xffff,"GDCM Private Sentinel" ) );
      assert (it != DictInternal.end());
      return it->second;
      }
    assert( DictInternal.count(tag) == 1 );
    return it->second;
    }


  void PrintXML() const
    {
    MapDictEntry::const_iterator it = DictInternal.begin();
    std::cout « "<dict edition=\"2008\">\n";
    for(;it != DictInternal.end(); ++it)
      {
      const PrivateTag &t = it->first;
      const DictEntry &de = it->second;
      std::cout « "  <entry group=\"" « std::hex « std::setw(4)
        « std::setfill('0') « t.GetGroup() « "\"" «
        " element=\"xx" « std::setw(2) « std::setfill('0')« t.GetElement() « "\"" « " vr=\""
        « de.GetVR() « "\" vm=\"" « de.GetVM() « "\" owner=\""
        « t.GetOwner();
      const char *name = de.GetName();
      if( *name == 0 )
        {
        std::cout « "\"/>\n";
        }
      else
        {
        std::cout « "\" name=\"" « de.GetName() « "\"/>\n";
        }
      }
    std::cout « "</dict>\n";
    }

  bool IsEmpty() const { return DictInternal.empty(); }
protected:
  friend class Dicts;
  void LoadDefault();

```

```
private:
  PrivateDict &operator=(const PrivateDict &_val) = delete;
  PrivateDict(const PrivateDict &_val) = delete;

  MapDictEntry DictInternal;
};
//----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& os, const PrivateDict &val)
{
  PrivateDict::MapDictEntry::const_iterator it = val.DictInternal.begin();
  for(;it != val.DictInternal.end(); ++it)
    {
    const PrivateTag &t = it->first;
    const DictEntry &de = it->second;
    os « t « " " « de « '\n';
    }

  return os;
}

} // end namespace gdcm

#endif //GDCMDICT_H
```

13.95 gdcmDictConverter.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
```

Include dependency graph for gdcmDictConverter.h:



Classes

- class gdcm::DictConverter

  Class to convert a .dic file into something else:

Namespaces

- namespace gdcm

13.96 gdcmDictConverter.h

Go to the documentation of this file.

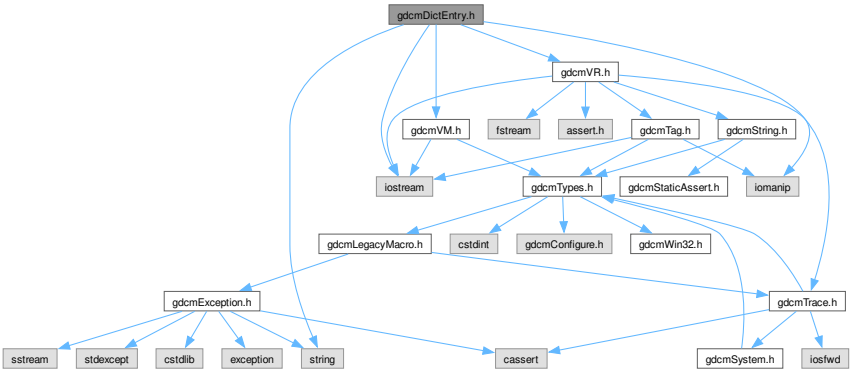
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014
00015 #ifndef GDCMDICTCONVERTER_H
00016 #define GDCMDICTCONVERTER_H
00017
00018 #include "gdcmTypes.h"
00019 #include "gdcmVR.h"
00020 #include "gdcmVM.h"
00021
00022 #include <string>
00023
00024 namespace gdcm
00025 {
00026
00027 class DictConverterInternal;
00036 class GDCM_EXPORT DictConverter
00037 {
00038 public:
00039   DictConverter();
00040   ~DictConverter();
00041   void SetInputFileName(const char* filename);
00042   const std::string &GetInputFilename() const;
00043   void SetOutputFileName(const char* filename);
00044   const std::string &GetOutputFilename() const;
00045
00046   int GetOutputType() const {
00047     return OutputType;
00048   }
00049   void SetOutputType(int type) {
00050     OutputType = type;
00051   }
00052   const std::string &GetDictName() const;
00053   void SetDictName(const char *name);
00054
00055   void Convert();
00056
00057   // Leaving them public for now. Not really user oriented but may be
00058   // useful
00059   static bool ReadVR(const char *raw, VR::VRType &type);
00060   static bool ReadVM(const char *raw, VM::VMType &type);
00061   static bool Readuint16(const char *raw, uint16_t &ov);
00062
00063   enum OutputTypes {
00064     DICT_DEFAULT = 0,
00065     DICT_DEBUG,
00066     DICT_XML
00067   };
00068
00069 protected:
00070   void WriteHeader();
00071   void WriteFooter();
00072   bool ConvertToXML(const char *raw, std::string &cxx);
00073   bool ConvertToCXX(const char *raw, std::string &cxx);
00074   void AddGroupLength();
```

```
00075
00076 private:
00077   DictConverterInternal *Internal;
00078
00079   int OutputType;
00080 };
00081
00082 } // end namespace gdcm
00083
00084 #endif //GDCMDICTCONVERTER_H
```

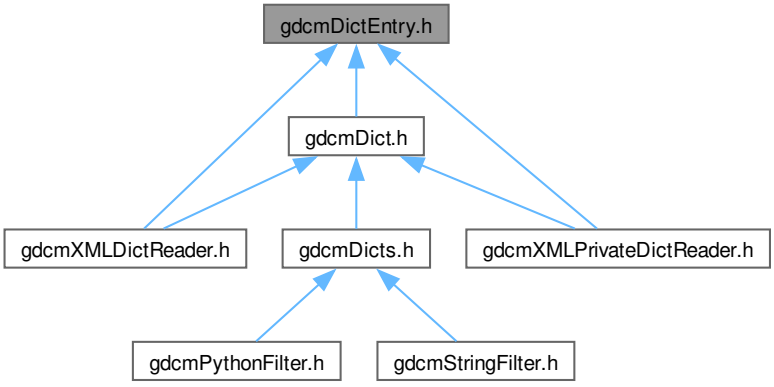
13.97 gdcmDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
```

Include dependency graph for gdcmDictEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::DictEntry

  Class to represent an Entry in the Dict.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)

13.98 gdcmDictEntry.h

Go to the documentation of this file.

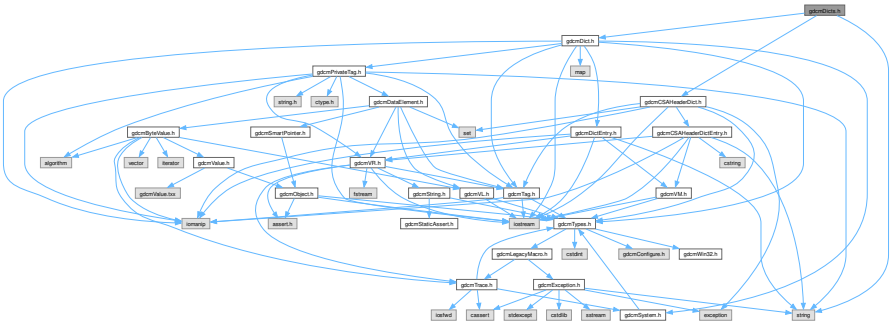
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDICTENTRY_H
#define GDCMDICTENTRY_H

#include "gdcmVR.h"
#include "gdcmVM.h"

#include <string>
#include <iostream>
#include <iomanip>

namespace gdcm
{
class GDCM_EXPORT DictEntry
{
public:
  DictEntry(const char *name = "", const char *keyword = "", VR const &vr = VR::INVALID, VM const &vm = VM::VM0, bool ret = false):
    Name(name),
    Keyword(keyword),
    ValueRepresentation(vr),
    ValueMultiplicity(vm),
    Retired(ret),
    GroupXX(false),
    ElementXX(false)
  {
  }

  friend std::ostream& operator«(std::ostream& _os, const DictEntry &_val);

  const VR &GetVR() const { return ValueRepresentation; }
  void SetVR(const VR & vr) { ValueRepresentation = vr; }
//  bool IsValid() const { return ValueRepresentation != VR::VR_END; }
//    !Name.empty() /*&& ValueRepresentation && ValueMultiplicity*/; }

  const VM &GetVM() const { return ValueMultiplicity; }
  void SetVM(VM const & vm) { ValueMultiplicity = vm; }
```

```
00061
00063   const char *GetName() const { return Name.c_str(); }
00064   void SetName(const char* name) { Name = name; }
00065
00067   const char *GetKeyword() const { return Keyword.c_str(); }
00068   void SetKeyword(const char* keyword) { Keyword = keyword; }
00069
00071   bool GetRetired() const { return Retired; }
00072   void SetRetired(bool retired) { Retired = retired; }
00073
00074   // <entry group="50xx" element="0005" vr="US" vm="1" retired="true" version="3">
00076   void SetGroupXX(bool v) { GroupXX = v; }
00077
00078   // <entry group="0020" element="31xx" vr="CS" vm="1-n" retired="true" version="2">
00080   void SetElementXX(bool v) { ElementXX = v; }
00081
00084   bool IsUnique() const { return ElementXX == false && GroupXX == false; }
00085
00086 private:
00087   //
00088   friend class Dict;
00089   static bool CheckKeywordAgainstName(const char *name, const char *keyword);
00090
00091 private:
00092   std::string Name;
00093   std::string Keyword;
00094   VR ValueRepresentation;
00095   VM ValueMultiplicity;
00096   bool Retired : 1;
00097   bool GroupXX : 1;
00098   bool ElementXX : 1;
00099 };
00100
00101 #if 0
00102 class GDCM_EXPORT PrivateDictEntry : public DictEntry
00103 {
00104 public:
00105   PrivateDictEntry(const char *name = "", VR::VRType const &vr = VR::INVALID, VM::VMType const &vm =
      VM::VM0 , bool ret = false, const char *owner = ""):DictEntry(name,vr,vm,ret),Owner(owner) {}
00106   PrivateDictEntry(const char *name, const char *vr, const char *vm):DictEntry(name,vr,vm) {}
00107
00108   const char *GetOwner() const { return Owner.c_str(); }
00109   void SetOwner(const char *owner) { Owner = owner; }
00110
00111 private:
00112   // SIEMENS MED, GEMS_PETD_01 ...
00113   std::string Owner;
00114 };
00115 #endif
00116
00117 //-----------------------------------------------------------------------------
00118 inline std::ostream& operator«(std::ostream& os, const DictEntry &val)
00119 {
00120   if( val.Name.empty() )
00121     {
00122     os « "[No name]";
00123     }
00124   else
00125     {
00126     os « val.Name;
00127     }
00128   if( val.Keyword.empty() )
00129     {
00130     os « "[No keyword]";
00131     }
00132   else
00133     {
00134     os « val.Keyword;
00135     }
00136   os « "\t" « val.ValueRepresentation « "\t" « val.ValueMultiplicity;
00137   if( val.Retired )
00138     {
00139     os « "\t(RET)";
00140     }
00141   return os;
00142 }
00143
00144 } // end namespace gdcm
00145
00146 #endif //GDCMDICTENTRY_H
```

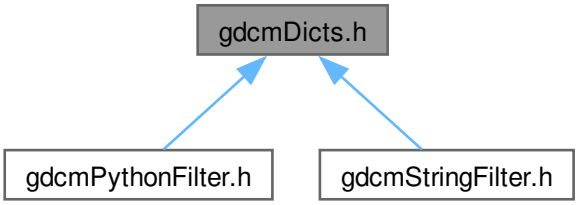
13.99 gdcmDicts.h File Reference

```
#include "gdcmDict.h"
#include "gdcmCSAHeaderDict.h"
#include <string>
```

Include dependency graph for gdcmDicts.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Dicts

  Class to manipulate the sum of knowledge (all the dict user load).

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)

13.100 gdcmDicts.h

Go to the documentation of this file.

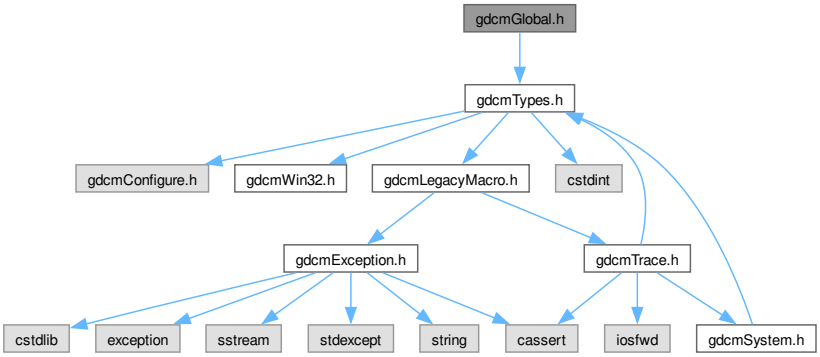
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDICTS_H
#define GDCMDICTS_H

#include "gdcmDict.h"
#include "gdcmCSAHeaderDict.h"

#include <string>

namespace gdcm
{
class GDCM_EXPORT Dicts
{
  friend std::ostream& operator«(std::ostream &_os, const Dicts &d);
public:
  Dicts();
  ~Dicts();
  Dicts &operator=(const Dicts &_val) = delete;
  Dicts(const Dicts &_val) = delete;

  // DataSet::GetPrivateCreator
  const DictEntry &GetDictEntry(const Tag& tag, const char *owner = nullptr) const;

  const DictEntry &GetDictEntry(const PrivateTag& tag) const;

  //enum PublicTypes {
  //  DICOMV3_DICT,
  //  ACRNEMA_DICT,
  //  NIH_DICT
  //};
  const Dict &GetPublicDict() const;

  const PrivateDict &GetPrivateDict() const;
  PrivateDict &GetPrivateDict();

  const CSAHeaderDict &GetCSAHeaderDict() const;

  bool IsEmpty() const { return GetPublicDict().IsEmpty(); }

protected:
  typedef enum {
    PHILIPS,
    GEMS,
    SIEMENS
  //  ...
  } ConstructorType;
  static const char *GetConstructorString(ConstructorType type);

  friend class Global;
  void LoadDefaults();

private:
  // Public dict:
  Dict PublicDict;

  // Private Dicts:
  PrivateDict ShadowDict;

  CSAHeaderDict CSADict;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Dicts &d)
{
```

```
  (void)d;
  return os;
}


} // end namespace gdcm

#endif //GDCMDICTS_H
```

13.101 gdcmGlobal.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmGlobal.h:



Classes

- class gdcm::Global

  Global.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)

Variables

- static Global gdcm::GlobalInstance

13.102 gdcmGlobal.h

Go to the documentation of this file.

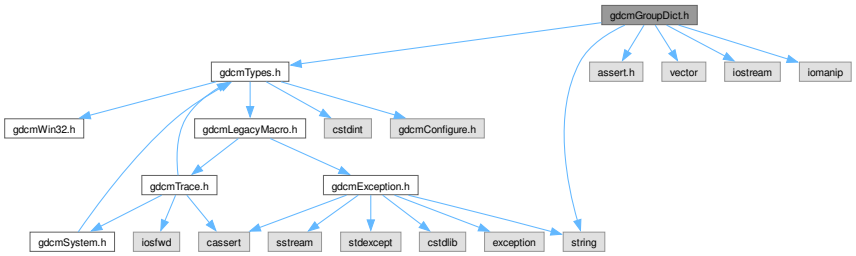
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// Implementation detail was shamelessly borowed from the VTK excellent
// implementation of debug leak manager singleton:
/*=========================================================================

  Program:   Visualization Toolkit
  Module:    $RCSfile: vtkDebugLeaks.cxx,v $

  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
  All rights reserved.
  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMGLOBAL_H
#define GDCMGLOBAL_H

#include "gdcmTypes.h"

namespace gdcm
{
class GlobalInternal;
class Dicts;
class Defs;
class GDCM_EXPORT Global // why expose the symbol I think I only need to expose the instance...
{
  friend std::ostream& operator«(std::ostream &_os, const Global &g);
public:
  Global();
  ~Global();
  Global &operator=(const Global &_val) = delete;
  Global(const Global &_val) = delete;

  Dicts const &GetDicts() const;
  Dicts &GetDicts();

  Defs const &GetDefs() const;

  static Global& GetInstance();

  bool LoadResourcesFiles();

  bool Append(const char *path);

  bool Prepend(const char *path);

protected:
  const char *Locate(const char *resfile) const;

private:
  // PIMPL:
  // but we could have also directly exposed a Dicts *Internals;
  static GlobalInternal *Internals;
};
//----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Global &g)
{
  (void)g;
  return os;
}
```

```
// This instance will show up in any translation unit that uses
// Global or that has a singleton.  It will make sure
// Global is initialized before it is used and is the last
// static object destroyed.
static Global GlobalInstance;

} // end namespace gdcm

#endif //GDCMGLOBAL_H
```

13.103 gdcmGroupDict.h File Reference

```
#include "gdcmTypes.h"
#include <assert.h>
#include <vector>
#include <string>
#include <iostream>
#include <iomanip>
```

Include dependency graph for gdcmGroupDict.h:



Classes

- class gdcm::GroupDict

  Class to represent the mapping from group number to its abbreviation and name.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)

13.104 gdcmGroupDict.h

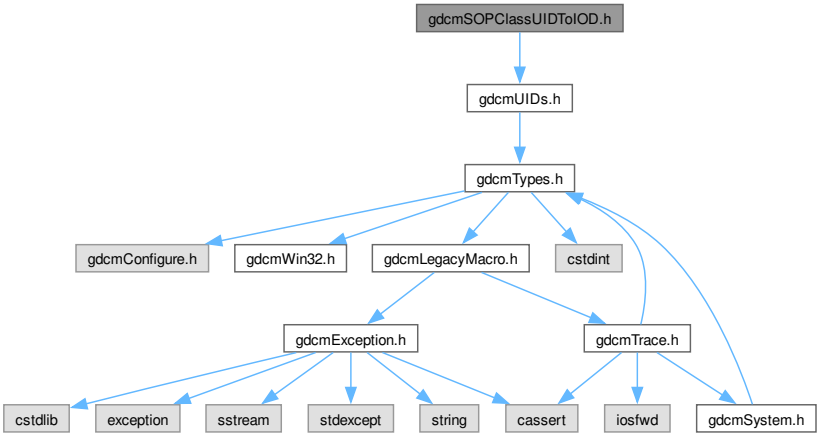
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMGROUPDICT_H
#define GDCMGROUPDICT_H

#include "gdcmTypes.h"

#include <assert.h>
#include <vector>
#include <string>
#include <iostream>
#include <iomanip>

namespace gdcm
{
class GDCM_EXPORT GroupDict
{
public:
  typedef std::vector<std::string> GroupStringVector;
  GroupDict() { FillDefaultGroupName(); }
  ~GroupDict() = default;

  friend std::ostream& operator«(std::ostream& _os, const GroupDict &_val);

  size_t Size() const
    {
    assert( Names.size() == Abbreviations.size() );
    return Names.size(); }

  std::string const &GetAbbreviation(uint16_t num) const;

  std::string const &GetName(uint16_t num) const;

protected:
  void Add(std::string const &abbreviation, std::string const &name);
  void Insert(uint16_t num, std::string const &abbreviation, std::string const &name);

private:
  // Generated implementation, see gdcmDefaultGroupNames
  void FillDefaultGroupName();

  GroupDict &operator=(const GroupDict &_val); // purposely not implemented
  GroupDict(const GroupDict &_val); // purposely not implemented

  GroupStringVector Abbreviations;
  GroupStringVector Names;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const GroupDict &_val)
{
  size_t size = _val.Size();
  for(size_t i=0; i<size; ++i)
    {
    _os « std::hex « std::setw(4) « std::setfill( '0' ) « i « ","
      « _val.GetAbbreviation((uint16_t)i) « "," « _val.GetName((uint16_t)i) « "\n";
    }
  return _os;
}

} // end namespace gdcm

#endif //GDCMGROUPDICT_H
```

13.105 gdcmSOPClassUIDToIOD.h File Reference

`#include "gdcmUIDs.h"`

Include dependency graph for gdcmSOPClassUIDToIOD.h:



Classes

- class gdcm::SOPClassUIDToIOD

  Class convert a class SOP Class UID into IOD.

Namespaces

- namespace gdcm

13.106 gdcmSOPClassUIDToIOD.h

Go to the documentation of this file.

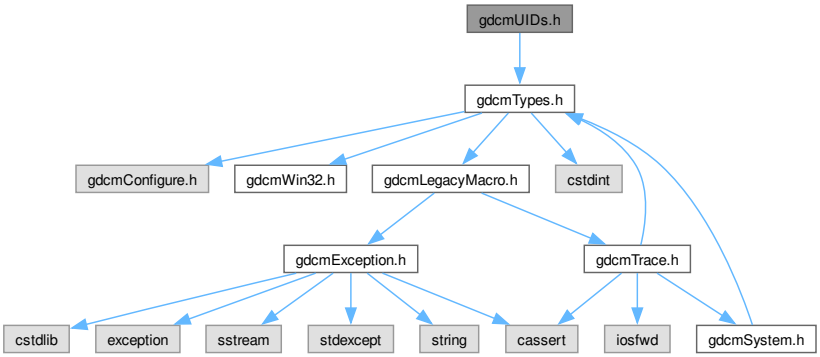
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMSOPCLASSUIDTOIOD_H
#define GDCMSOPCLASSUIDTOIOD_H

```

```
#include "gdcmUIDs.h"

namespace gdcm
{

class GDCM_EXPORT SOPClassUIDToIOD
{
public:
  static const char *GetIOD(UIDs const & uid);

  static unsigned int GetNumberOfSOPClassToIOD();

  typedef const char* const (SOPClassUIDToIODType)[2];
  static SOPClassUIDToIODType* GetSOPClassUIDToIODs();

  static SOPClassUIDToIODType& GetSOPClassUIDToIOD(unsigned int i);

  static const char *GetSOPClassUIDFromIOD(const char *iod);
  static const char *GetIODFromSOPClassUID(const char *sopclassuid);
};

} // end namespace gdcm

#endif //GDCMSOPCLASSUIDTOIOD_H
```

13.107 gdcmUIDs.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmUIDs.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::UIDs

  all known uids

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)

13.108 gdcmUIDs.h

Go to the documentation of this file.

```
00001
00002 // GENERATED FILE DO NOT EDIT
00003 // $ xsltproc UIDToC++.xsl Part6.xml > gdcmUIDs.h
00004
00005 /*=========================================================================
00006
00007   Program: GDCM (Grassroots DICOM). A DICOM library
00008
00009   Copyright (c) 2006-2011 Mathieu Malaterre
00010   All rights reserved.
00011   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00012
00013      This software is distributed WITHOUT ANY WARRANTY; without even
00014      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00015      PURPOSE.  See the above copyright notice for more information.
00016
00017 =========================================================================*/
00018
00019 #ifndef GDCMUIDS_H
00020 #define GDCMUIDS_H
00021
00022 #include "gdcmTypes.h"
00023
00024 namespace gdcm
00025 {
00026
00030 class GDCM_EXPORT UIDs
00031 {
00032 public:
00033   typedef enum {
00034 uid_1_2_840_10008_1_1 = 1, // Verification SOP Class
00035 uid_1_2_840_10008_1_2 = 2, // Implicit VR Little Endian: Default Transfer Syntax for DICOM
00036 uid_1_2_840_10008_1_2_1 = 3, // Explicit VR Little Endian
00037 uid_1_2_840_10008_1_2_1_99 = 4, // Deflated Explicit VR Little Endian
00038 uid_1_2_840_10008_1_2_2 = 5, // Explicit VR Big Endian
00039 uid_1_2_840_10008_1_2_4_50 = 6, // JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit
      Image Compression
00040 uid_1_2_840_10008_1_2_4_51 = 7, // JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG
      12 Bit Image Compression (Process 4 only)
00041 uid_1_2_840_10008_1_2_4_52 = 8, // JPEG Extended (Process 3 & 5)
00042 uid_1_2_840_10008_1_2_4_53 = 9, // JPEG Spectral Selection, Non-Hierarchical (Process 6 & 8)
00043 uid_1_2_840_10008_1_2_4_54 = 10, // JPEG Spectral Selection, Non-Hierarchical (Process 7 & 9)
00044 uid_1_2_840_10008_1_2_4_55 = 11, // JPEG Full Progression, Non-Hierarchical (Process 10 & 12)
00045 uid_1_2_840_10008_1_2_4_56 = 12, // JPEG Full Progression, Non-Hierarchical (Process 11 & 13)
00046 uid_1_2_840_10008_1_2_4_57 = 13, // JPEG Lossless, Non-Hierarchical (Process 14)
00047 uid_1_2_840_10008_1_2_4_58 = 14, // JPEG Lossless, Non-Hierarchical (Process 15)
00048 uid_1_2_840_10008_1_2_4_59 = 15, // JPEG Extended, Hierarchical (Process 16 & 18)
00049 uid_1_2_840_10008_1_2_4_60 = 16, // JPEG Extended, Hierarchical (Process 17 & 19)
00050 uid_1_2_840_10008_1_2_4_61 = 17, // JPEG Spectral Selection, Hierarchical (Process 20 & 22)
00051 uid_1_2_840_10008_1_2_4_62 = 18, // JPEG Spectral Selection, Hierarchical (Process 21 & 23)
00052 uid_1_2_840_10008_1_2_4_63 = 19, // JPEG Full Progression, Hierarchical (Process 24 & 26)
00053 uid_1_2_840_10008_1_2_4_64 = 20, // JPEG Full Progression, Hierarchical (Process 25 & 27)
00054 uid_1_2_840_10008_1_2_4_65 = 21, // JPEG Lossless, Hierarchical (Process 28)
00055 uid_1_2_840_10008_1_2_4_66 = 22, // JPEG Lossless, Hierarchical (Process 29)
00056 uid_1_2_840_10008_1_2_4_70 = 23, // JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14
      [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression
00057 uid_1_2_840_10008_1_2_4_80 = 24, // JPEG-LS Lossless Image Compression
00058 uid_1_2_840_10008_1_2_4_81 = 25, // JPEG-LS Lossy (Near-Lossless) Image Compression
00059 uid_1_2_840_10008_1_2_4_90 = 26, // JPEG 2000 Image Compression (Lossless Only)
```

```
uid_1_2_840_10008_1_2_4_91 = 27, // JPEG 2000 Image Compression
uid_1_2_840_10008_1_2_4_92 = 28, // JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)
uid_1_2_840_10008_1_2_4_93 = 29, // JPEG 2000 Part 2 Multi-component Image Compression
uid_1_2_840_10008_1_2_4_94 = 30, // JPIP Referenced
uid_1_2_840_10008_1_2_4_95 = 31, // JPIP Referenced Deflate
uid_1_2_840_10008_1_2_4_100 = 32, // MPEG2 Main Profile @ Main Level
uid_1_2_840_10008_1_2_5 = 33, // RLE Lossless
uid_1_2_840_10008_1_2_6_1 = 34, // RFC 2557 MIME encapsulation
uid_1_2_840_10008_1_2_6_2 = 35, // XML Encoding
uid_1_2_840_10008_1_3_10 = 36, // Media Storage Directory Storage
uid_1_2_840_10008_1_4_1_1 = 37, // Talairach Brain Atlas Frame of Reference
uid_1_2_840_10008_1_4_1_2 = 38, // SPM2 T1 Frame of Reference
uid_1_2_840_10008_1_4_1_3 = 39, // SPM2 T2 Frame of Reference
uid_1_2_840_10008_1_4_1_4 = 40, // SPM2 PD Frame of Reference
uid_1_2_840_10008_1_4_1_5 = 41, // SPM2 EPI Frame of Reference
uid_1_2_840_10008_1_4_1_6 = 42, // SPM2 FIL T1 Frame of Reference
uid_1_2_840_10008_1_4_1_7 = 43, // SPM2 PET Frame of Reference
uid_1_2_840_10008_1_4_1_8 = 44, // SPM2 TRANSM Frame of Reference
uid_1_2_840_10008_1_4_1_9 = 45, // SPM2 SPECT Frame of Reference
uid_1_2_840_10008_1_4_1_10 = 46, // SPM2 GRAY Frame of Reference
uid_1_2_840_10008_1_4_1_11 = 47, // SPM2 WHITE Frame of Reference
uid_1_2_840_10008_1_4_1_12 = 48, // SPM2 CSF Frame of Reference
uid_1_2_840_10008_1_4_1_13 = 49, // SPM2 BRAINMASK Frame of Reference
uid_1_2_840_10008_1_4_1_14 = 50, // SPM2 AVG305T1 Frame of Reference
uid_1_2_840_10008_1_4_1_15 = 51, // SPM2 AVG152T1 Frame of Reference
uid_1_2_840_10008_1_4_1_16 = 52, // SPM2 AVG152T2 Frame of Reference
uid_1_2_840_10008_1_4_1_17 = 53, // SPM2 AVG152PD Frame of Reference
uid_1_2_840_10008_1_4_1_18 = 54, // SPM2 SINGLESUBJT1 Frame of Reference
uid_1_2_840_10008_1_4_2_1 = 55, // ICBM 452 T1 Frame of Reference
uid_1_2_840_10008_1_4_2_2 = 56, // ICBM Single Subject MRI Frame of Reference
uid_1_2_840_10008_1_9 = 57, // Basic Study Content Notification SOP Class
uid_1_2_840_10008_1_20_1 = 58, // Storage Commitment Push Model SOP Class
uid_1_2_840_10008_1_20_1_1 = 59, // Storage Commitment Push Model SOP Instance
uid_1_2_840_10008_1_20_2 = 60, // Storage Commitment Pull Model SOP Class
uid_1_2_840_10008_1_20_2_1 = 61, // Storage Commitment Pull Model SOP Instance
uid_1_2_840_10008_1_40 = 62, // Procedural Event Logging SOP Class
uid_1_2_840_10008_1_40_1 = 63, // Procedural Event Logging SOP Instance
uid_1_2_840_10008_1_42 = 64, // Substance Administration Logging SOP Class
uid_1_2_840_10008_1_42_1 = 65, // Substance Administration Logging SOP Instance
uid_1_2_840_10008_2_6_1 = 66, // DICOM UID Registry
uid_1_2_840_10008_2_16_4 = 67, // DICOM Controlled Terminology
uid_1_2_840_10008_3_1_1_1 = 68, // DICOM Application Context Name
uid_1_2_840_10008_3_1_2_1_1 = 69, // Detached Patient Management SOP Class
uid_1_2_840_10008_3_1_2_1_4 = 70, // Detached Patient Management Meta SOP Class
uid_1_2_840_10008_3_1_2_2_1 = 71, // Detached Visit Management SOP Class
uid_1_2_840_10008_3_1_2_3_1 = 72, // Detached Study Management SOP Class
uid_1_2_840_10008_3_1_2_3_2 = 73, // Study Component Management SOP Class
uid_1_2_840_10008_3_1_2_3_3 = 74, // Modality Performed Procedure Step SOP Class
uid_1_2_840_10008_3_1_2_3_4 = 75, // Modality Performed Procedure Step Retrieve SOP Class
uid_1_2_840_10008_3_1_2_3_5 = 76, // Modality Performed Procedure Step Notification SOP Class
uid_1_2_840_10008_3_1_2_5_1 = 77, // Detached Results Management SOP Class
uid_1_2_840_10008_3_1_2_5_4 = 78, // Detached Results Management Meta SOP Class
uid_1_2_840_10008_3_1_2_5_5 = 79, // Detached Study Management Meta SOP Class
uid_1_2_840_10008_3_1_2_6_1 = 80, // Detached Interpretation Management SOP Class
uid_1_2_840_10008_4_2 = 81, // Storage Service Class
uid_1_2_840_10008_5_1_1_1 = 82, // Basic Film Session SOP Class
uid_1_2_840_10008_5_1_1_2 = 83, // Basic Film Box SOP Class
uid_1_2_840_10008_5_1_1_4 = 84, // Basic Grayscale Image Box SOP Class
uid_1_2_840_10008_5_1_1_4_1 = 85, // Basic Color Image Box SOP Class
uid_1_2_840_10008_5_1_1_4_2 = 86, // Referenced Image Box SOP Class
uid_1_2_840_10008_5_1_1_9 = 87, // Basic Grayscale Print Management Meta SOP Class
uid_1_2_840_10008_5_1_1_9_1 = 88, // Referenced Grayscale Print Management Meta SOP Class
uid_1_2_840_10008_5_1_1_14 = 89, // Print Job SOP Class
uid_1_2_840_10008_5_1_1_15 = 90, // Basic Annotation Box SOP Class
uid_1_2_840_10008_5_1_1_16 = 91, // Printer SOP Class
uid_1_2_840_10008_5_1_1_16_376 = 92, // Printer Configuration Retrieval SOP Class
uid_1_2_840_10008_5_1_1_17 = 93, // Printer SOP Instance
uid_1_2_840_10008_5_1_1_17_376 = 94, // Printer Configuration Retrieval SOP Instance
uid_1_2_840_10008_5_1_1_18 = 95, // Basic Color Print Management Meta SOP Class
uid_1_2_840_10008_5_1_1_18_1 = 96, // Referenced Color Print Management Meta SOP Class
uid_1_2_840_10008_5_1_1_22 = 97, // VOI LUT Box SOP Class
uid_1_2_840_10008_5_1_1_23 = 98, // Presentation LUT SOP Class
uid_1_2_840_10008_5_1_1_24 = 99, // Image Overlay Box SOP Class
uid_1_2_840_10008_5_1_1_24_1 = 100, // Basic Print Image Overlay Box SOP Class
uid_1_2_840_10008_5_1_1_25 = 101, // Print Queue SOP Instance
uid_1_2_840_10008_5_1_1_26 = 102, // Print Queue Management SOP Class
uid_1_2_840_10008_5_1_1_27 = 103, // Stored Print Storage SOP Class
uid_1_2_840_10008_5_1_1_29 = 104, // Hardcopy Grayscale Image Storage SOP Class
uid_1_2_840_10008_5_1_1_30 = 105, // Hardcopy Color Image Storage SOP Class
uid_1_2_840_10008_5_1_1_31 = 106, // Pull Print Request SOP Class
uid_1_2_840_10008_5_1_1_32 = 107, // Pull Stored Print Management Meta SOP Class
```

```
uid_1_2_840_10008_5_1_1_33 = 108, // Media Creation Management SOP Class UID
uid_1_2_840_10008_5_1_4_1_1_1 = 109, // Computed Radiography Image Storage
uid_1_2_840_10008_5_1_4_1_1_1_1 = 110, // Digital X-Ray Image Storage - For Presentation
uid_1_2_840_10008_5_1_4_1_1_1_1_1 = 111, // Digital X-Ray Image Storage - For Processing
uid_1_2_840_10008_5_1_4_1_1_1_2 = 112, // Digital Mammography X-Ray Image Storage - For Presentation
uid_1_2_840_10008_5_1_4_1_1_1_2_1 = 113, // Digital Mammography X-Ray Image Storage - For Processing
uid_1_2_840_10008_5_1_4_1_1_1_3 = 114, // Digital Intra-oral X-Ray Image Storage - For Presentation
uid_1_2_840_10008_5_1_4_1_1_1_3_1 = 115, // Digital Intra-oral X-Ray Image Storage - For Processing
uid_1_2_840_10008_5_1_4_1_1_2 = 116, // CT Image Storage
uid_1_2_840_10008_5_1_4_1_1_2_1 = 117, // Enhanced CT Image Storage
uid_1_2_840_10008_5_1_4_1_1_3 = 118, // Ultrasound Multi-frame Image Storage
uid_1_2_840_10008_5_1_4_1_1_3_1 = 119, // Ultrasound Multi-frame Image Storage
uid_1_2_840_10008_5_1_4_1_1_4 = 120, // MR Image Storage
uid_1_2_840_10008_5_1_4_1_1_4_1 = 121, // Enhanced MR Image Storage
uid_1_2_840_10008_5_1_4_1_1_4_2 = 122, // MR Spectroscopy Storage
uid_1_2_840_10008_5_1_4_1_1_5 = 123, // Nuclear Medicine Image Storage
uid_1_2_840_10008_5_1_4_1_1_6 = 124, // Ultrasound Image Storage
uid_1_2_840_10008_5_1_4_1_1_6_1 = 125, // Ultrasound Image Storage
uid_1_2_840_10008_5_1_4_1_1_7 = 126, // Secondary Capture Image Storage
uid_1_2_840_10008_5_1_4_1_1_7_1 = 127, // Multi-frame Single Bit Secondary Capture Image Storage
uid_1_2_840_10008_5_1_4_1_1_7_2 = 128, // Multi-frame Grayscale Byte Secondary Capture Image Storage
uid_1_2_840_10008_5_1_4_1_1_7_3 = 129, // Multi-frame Grayscale Word Secondary Capture Image Storage
uid_1_2_840_10008_5_1_4_1_1_7_4 = 130, // Multi-frame True Color Secondary Capture Image Storage
uid_1_2_840_10008_5_1_4_1_1_8 = 131, // Standalone Overlay Storage
uid_1_2_840_10008_5_1_4_1_1_9 = 132, // Standalone Curve Storage
uid_1_2_840_10008_5_1_4_1_1_9_1 = 133, // Waveform Storage - Trial
uid_1_2_840_10008_5_1_4_1_1_9_1_1 = 134, // 12-lead ECG Waveform Storage
uid_1_2_840_10008_5_1_4_1_1_9_1_2 = 135, // General ECG Waveform Storage
uid_1_2_840_10008_5_1_4_1_1_9_1_3 = 136, // Ambulatory ECG Waveform Storage
uid_1_2_840_10008_5_1_4_1_1_9_2_1 = 137, // Hemodynamic Waveform Storage
uid_1_2_840_10008_5_1_4_1_1_9_3_1 = 138, // Cardiac Electrophysiology Waveform Storage
uid_1_2_840_10008_5_1_4_1_1_9_4_1 = 139, // Basic Voice Audio Waveform Storage
uid_1_2_840_10008_5_1_4_1_1_10 = 140, // Standalone Modality LUT Storage
uid_1_2_840_10008_5_1_4_1_1_11 = 141, // Standalone VOI LUT Storage
uid_1_2_840_10008_5_1_4_1_1_11_1 = 142, // Grayscale Softcopy Presentation State Storage SOP Class
uid_1_2_840_10008_5_1_4_1_1_11_2 = 143, // Color Softcopy Presentation State Storage SOP Class
uid_1_2_840_10008_5_1_4_1_1_11_3 = 144, // Pseudo-Color Softcopy Presentation State Storage SOP Class
uid_1_2_840_10008_5_1_4_1_1_11_4 = 145, // Blending Softcopy Presentation State Storage SOP Class
uid_1_2_840_10008_5_1_4_1_1_12_1 = 146, // X-Ray Angiographic Image Storage
uid_1_2_840_10008_5_1_4_1_1_12_1_1 = 147, // Enhanced XA Image Storage
uid_1_2_840_10008_5_1_4_1_1_12_2 = 148, // X-Ray Radiofluoroscopic Image Storage
uid_1_2_840_10008_5_1_4_1_1_12_2_1 = 149, // Enhanced XRF Image Storage
uid_1_2_840_10008_5_1_4_1_1_13_1_1 = 150, // X-Ray 3D Angiographic Image Storage
uid_1_2_840_10008_5_1_4_1_1_13_1_2 = 151, // X-Ray 3D Craniofacial Image Storage
uid_1_2_840_10008_5_1_4_1_1_12_3 = 152, // X-Ray Angiographic Bi-Plane Image Storage
uid_1_2_840_10008_5_1_4_1_1_20 = 153, // Nuclear Medicine Image Storage
uid_1_2_840_10008_5_1_4_1_1_66 = 154, // Raw Data Storage
uid_1_2_840_10008_5_1_4_1_1_66_1 = 155, // Spatial Registration Storage
uid_1_2_840_10008_5_1_4_1_1_66_2 = 156, // Spatial Fiducials Storage
uid_1_2_840_10008_5_1_4_1_1_66_3 = 157, // Deformable Spatial Registration Storage
uid_1_2_840_10008_5_1_4_1_1_66_4 = 158, // Segmentation Storage
uid_1_2_840_10008_5_1_4_1_1_67 = 159, // Real World Value Mapping Storage
uid_1_2_840_10008_5_1_4_1_1_77_1 = 160, // VL Image Storage - Trial
uid_1_2_840_10008_5_1_4_1_1_77_2 = 161, // VL Multi-frame Image Storage - Trial
uid_1_2_840_10008_5_1_4_1_1_77_1_1 = 162, // VL Endoscopic Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1 = 163, // Video Endoscopic Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_2 = 164, // VL Microscopic Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1 = 165, // Video Microscopic Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_3 = 166, // VL Slide-Coordinates Microscopic Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_4 = 167, // VL Photographic Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1 = 168, // Video Photographic Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1 = 169, // Ophthalmic Photography 8 Bit Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2 = 170, // Ophthalmic Photography 16 Bit Image Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3 = 171, // Stereometric Relationship Storage
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4 = 172, // Ophthalmic Tomography Image Storage
uid_1_2_840_10008_5_1_4_1_1_88_1 = 173, // Text SR Storage - Trial
uid_1_2_840_10008_5_1_4_1_1_88_2 = 174, // Audio SR Storage - Trial
uid_1_2_840_10008_5_1_4_1_1_88_3 = 175, // Detail SR Storage - Trial
uid_1_2_840_10008_5_1_4_1_1_88_4 = 176, // Comprehensive SR Storage - Trial
uid_1_2_840_10008_5_1_4_1_1_88_11 = 177, // Basic Text SR Storage
uid_1_2_840_10008_5_1_4_1_1_88_22 = 178, // Enhanced SR Storage
uid_1_2_840_10008_5_1_4_1_1_88_33 = 179, // Comprehensive SR Storage
uid_1_2_840_10008_5_1_4_1_1_88_40 = 180, // Procedure Log Storage
uid_1_2_840_10008_5_1_4_1_1_88_50 = 181, // Mammography CAD SR Storage
uid_1_2_840_10008_5_1_4_1_1_88_59 = 182, // Key Object Selection Document Storage
uid_1_2_840_10008_5_1_4_1_1_88_65 = 183, // Chest CAD SR Storage
uid_1_2_840_10008_5_1_4_1_1_88_67 = 184, // X-Ray Radiation Dose SR Storage
uid_1_2_840_10008_5_1_4_1_1_104_1 = 185, // Encapsulated PDF Storage
uid_1_2_840_10008_5_1_4_1_1_104_2 = 186, // Encapsulated CDA Storage
uid_1_2_840_10008_5_1_4_1_1_128 = 187, // Positron Emission Tomography Image Storage
uid_1_2_840_10008_5_1_4_1_1_129 = 188, // Standalone PET Curve Storage
```

```
uid_1_2_840_10008_5_1_4_1_1_481_1 = 189, // RT Image Storage
uid_1_2_840_10008_5_1_4_1_1_481_2 = 190, // RT Dose Storage
uid_1_2_840_10008_5_1_4_1_1_481_3 = 191, // RT Structure Set Storage
uid_1_2_840_10008_5_1_4_1_1_481_4 = 192, // RT Beams Treatment Record Storage
uid_1_2_840_10008_5_1_4_1_1_481_5 = 193, // RT Plan Storage
uid_1_2_840_10008_5_1_4_1_1_481_6 = 194, // RT Brachy Treatment Record Storage
uid_1_2_840_10008_5_1_4_1_1_481_7 = 195, // RT Treatment Summary Record Storage
uid_1_2_840_10008_5_1_4_1_1_481_8 = 196, // RT Ion Plan Storage
uid_1_2_840_10008_5_1_4_1_1_481_9 = 197, // RT Ion Beams Treatment Record Storage
uid_1_2_840_10008_5_1_4_1_2_1_1 = 198, // Patient Root Query/Retrieve Information Model - FIND
uid_1_2_840_10008_5_1_4_1_2_1_2 = 199, // Patient Root Query/Retrieve Information Model - MOVE
uid_1_2_840_10008_5_1_4_1_2_1_3 = 200, // Patient Root Query/Retrieve Information Model - GET
uid_1_2_840_10008_5_1_4_1_2_2_1 = 201, // Study Root Query/Retrieve Information Model - FIND
uid_1_2_840_10008_5_1_4_1_2_2_2 = 202, // Study Root Query/Retrieve Information Model - MOVE
uid_1_2_840_10008_5_1_4_1_2_2_3 = 203, // Study Root Query/Retrieve Information Model - GET
uid_1_2_840_10008_5_1_4_1_2_3_1 = 204, // Patient/Study Only Query/Retrieve Information Model - FIND
uid_1_2_840_10008_5_1_4_1_2_3_2 = 205, // Patient/Study Only Query/Retrieve Information Model - MOVE
uid_1_2_840_10008_5_1_4_1_2_3_3 = 206, // Patient/Study Only Query/Retrieve Information Model - GET
uid_1_2_840_10008_5_1_4_31 = 207, // Modality Worklist Information Model - FIND
uid_1_2_840_10008_5_1_4_32_1 = 208, // General Purpose Worklist Information Model - FIND
uid_1_2_840_10008_5_1_4_32_2 = 209, // General Purpose Scheduled Procedure Step SOP Class
uid_1_2_840_10008_5_1_4_32_3 = 210, // General Purpose Performed Procedure Step SOP Class
uid_1_2_840_10008_5_1_4_32 = 211, // General Purpose Worklist Management Meta SOP Class
uid_1_2_840_10008_5_1_4_33 = 212, // Instance Availability Notification SOP Class
uid_1_2_840_10008_5_1_4_34_1 = 213, // RT Beams Delivery Instruction Storage (Supplement 74 Frozen Draft)
uid_1_2_840_10008_5_1_4_34_2 = 214, // RT Conventional Machine Verification (Supplement 74 Frozen Draft)
uid_1_2_840_10008_5_1_4_34_3 = 215, // RT Ion Machine Verification (Supplement 74 Frozen Draft)
uid_1_2_840_10008_5_1_4_34_4 = 216, // Unified Worklist and Procedure Step Service Class
uid_1_2_840_10008_5_1_4_34_4_1 = 217, // Unified Procedure Step - Push SOP Class
uid_1_2_840_10008_5_1_4_34_4_2 = 218, // Unified Procedure Step - Watch SOP Class
uid_1_2_840_10008_5_1_4_34_4_3 = 219, // Unified Procedure Step - Pull SOP Class
uid_1_2_840_10008_5_1_4_34_4_4 = 220, // Unified Procedure Step - Event SOP Class
uid_1_2_840_10008_5_1_4_34_5 = 221, // Unified Worklist and Procedure Step SOP Instance
uid_1_2_840_10008_5_1_4_37_1 = 222, // General Relevant Patient Information Query
uid_1_2_840_10008_5_1_4_37_2 = 223, // Breast Imaging Relevant Patient Information Query
uid_1_2_840_10008_5_1_4_37_3 = 224, // Cardiac Relevant Patient Information Query
uid_1_2_840_10008_5_1_4_38_1 = 225, // Hanging Protocol Storage
uid_1_2_840_10008_5_1_4_38_2 = 226, // Hanging Protocol Information Model - FIND
uid_1_2_840_10008_5_1_4_38_3 = 227, // Hanging Protocol Information Model - MOVE
uid_1_2_840_10008_5_1_4_41 = 228, // Product Characteristics Query SOP Class
uid_1_2_840_10008_5_1_4_42 = 229, // Substance Approval Query SOP Class
uid_1_2_840_10008_15_0_3_1 = 230, // dicomDeviceName
uid_1_2_840_10008_15_0_3_2 = 231, // dicomDescription
uid_1_2_840_10008_15_0_3_3 = 232, // dicomManufacturer
uid_1_2_840_10008_15_0_3_4 = 233, // dicomManufacturerModelName
uid_1_2_840_10008_15_0_3_5 = 234, // dicomSoftwareVersion
uid_1_2_840_10008_15_0_3_6 = 235, // dicomVendorData
uid_1_2_840_10008_15_0_3_7 = 236, // dicomAETitle
uid_1_2_840_10008_15_0_3_8 = 237, // dicomNetworkConnectionReference
uid_1_2_840_10008_15_0_3_9 = 238, // dicomApplicationCluster
uid_1_2_840_10008_15_0_3_10 = 239, // dicomAssociationInitiator
uid_1_2_840_10008_15_0_3_11 = 240, // dicomAssociationAcceptor
uid_1_2_840_10008_15_0_3_12 = 241, // dicomHostname
uid_1_2_840_10008_15_0_3_13 = 242, // dicomPort
uid_1_2_840_10008_15_0_3_14 = 243, // dicomSOPClass
uid_1_2_840_10008_15_0_3_15 = 244, // dicomTransferRole
uid_1_2_840_10008_15_0_3_16 = 245, // dicomTransferSyntax
uid_1_2_840_10008_15_0_3_17 = 246, // dicomPrimaryDeviceType
uid_1_2_840_10008_15_0_3_18 = 247, // dicomRelatedDeviceReference
uid_1_2_840_10008_15_0_3_19 = 248, // dicomPreferredCalledAETitle
uid_1_2_840_10008_15_0_3_20 = 249, // dicomTLSCyphersuite
uid_1_2_840_10008_15_0_3_21 = 250, // dicomAuthorizedNodeCertificateReference
uid_1_2_840_10008_15_0_3_22 = 251, // dicomThisNodeCertificateReference
uid_1_2_840_10008_15_0_3_23 = 252, // dicomInstalled
uid_1_2_840_10008_15_0_3_24 = 253, // dicomStationName
uid_1_2_840_10008_15_0_3_25 = 254, // dicomDeviceSerialNumber
uid_1_2_840_10008_15_0_3_26 = 255, // dicomInstitutionName
uid_1_2_840_10008_15_0_3_27 = 256, // dicomInstitutionAddress
uid_1_2_840_10008_15_0_3_28 = 257, // dicomInstitutionDepartmentName
uid_1_2_840_10008_15_0_3_29 = 258, // dicomIssuerOfPatientID
uid_1_2_840_10008_15_0_3_30 = 259, // dicomPreferredCallingAETitle
uid_1_2_840_10008_15_0_3_31 = 260, // dicomSupportedCharacterSet
uid_1_2_840_10008_15_0_4_1 = 261, // dicomConfigurationRoot
uid_1_2_840_10008_15_0_4_2 = 262, // dicomDevicesRoot
uid_1_2_840_10008_15_0_4_3 = 263, // dicomUniqueAETitlesRegistryRoot
uid_1_2_840_10008_15_0_4_4 = 264, // dicomDevice
uid_1_2_840_10008_15_0_4_5 = 265, // dicomNetworkAE
uid_1_2_840_10008_15_0_4_6 = 266, // dicomNetworkConnection
uid_1_2_840_10008_15_0_4_7 = 267, // dicomUniqueAETitle
uid_1_2_840_10008_15_0_4_8 = 268, // dicomTransferCapability
//
```

```
uid_1_2_840_10008_5_1_4_1_1_77_1_6 = 269, // VL Whole Slide Microscopy
uid_1_2_840_10008_5_1_4_1_1_6_2    = 270, // Enhanced US Volume Storage
uid_1_2_840_10008_5_1_4_1_1_66_5   = 271, // Surface Segmentation Storage
uid_1_2_840_10008_5_1_4_1_1_13_1_3 = 272, // Breast Tomosynthesis Image Storage
uid_1_2_840_10008_5_1_4_1_1_2_2    = 273, // Legacy Converted Enhanced CT
uid_1_2_840_10008_5_1_4_1_1_4_4    = 274, // Legacy Converted Enhanced MR
uid_1_2_840_10008_5_1_4_1_1_128_1  = 275, // Legacy Converted Enhanced PET
uid_1_2_840_10008_1_2_4_101        = 276, // MPEG2 Main Profile High Level
uid_1_2_840_10008_1_2_4_102        = 277, // MPEG-4 AVC/H.264 High Profile Lev. 4.1
uid_1_2_840_10008_1_2_4_103        = 278, // MPEG-4 AVC/H.264 BD-comp High Profile Lev. 4.1

//
// 2019b
//
uid_1_2_840_10008_1_5_2              = 279,
uid_1_2_840_10008_1_5_3              = 280,
uid_1_2_840_10008_1_5_4              = 281,
uid_1_2_840_10008_1_5_5              = 282,
uid_1_2_840_10008_1_5_6              = 283,
uid_1_2_840_10008_1_5_7              = 284,
uid_1_2_840_10008_1_5_8              = 285,
uid_1_2_840_10008_1_20               = 286,
uid_1_2_840_10008_2_16_5             = 287,
uid_1_2_840_10008_2_16_6             = 288,
uid_1_2_840_10008_2_16_7             = 289,
uid_1_2_840_10008_2_16_8             = 290,
uid_1_2_840_10008_2_16_9             = 291,
uid_1_2_840_10008_2_16_10            = 292,
uid_1_2_840_10008_2_16_11            = 293,
uid_1_2_840_10008_2_16_12            = 294,
uid_1_2_840_10008_2_16_13            = 295,
uid_1_2_840_10008_2_16_14            = 296,
uid_1_2_840_10008_5_1_1_40           = 297,
uid_1_2_840_10008_5_1_1_40_1         = 298,
uid_1_2_840_10008_5_1_4_1_1_9_4_2    = 299,
uid_1_2_840_10008_5_1_4_1_1_9_5_1    = 300,
uid_1_2_840_10008_5_1_4_1_1_9_6_1    = 301,
uid_1_2_840_10008_5_1_4_1_1_11_5     = 302,
uid_1_2_840_10008_5_1_4_1_1_11_6     = 303,
uid_1_2_840_10008_1_2_4_104          = 304,
uid_1_2_840_10008_1_2_4_105          = 305,
uid_1_2_840_10008_1_2_4_106          = 306,
uid_1_2_840_10008_1_2_4_107          = 307,
uid_1_2_840_10008_1_2_4_108          = 308,
uid_1_2_840_10008_1_5_1              = 309,
uid_1_2_840_10008_5_1_4_1_1_11_7     = 310,
uid_1_2_840_10008_5_1_4_1_1_11_8     = 311,
uid_1_2_840_10008_5_1_4_1_1_11_9     = 312,
uid_1_2_840_10008_5_1_4_1_1_11_10    = 313,
uid_1_2_840_10008_5_1_4_1_1_11_11    = 314,
uid_1_2_840_10008_5_1_4_1_1_12_77    = 315,
uid_1_2_840_10008_5_1_4_1_1_13_1_4   = 316,
uid_1_2_840_10008_5_1_4_1_1_13_1_5   = 317,
uid_1_2_840_10008_5_1_4_1_1_14_1     = 318,
uid_1_2_840_10008_5_1_4_1_1_14_2     = 319,
uid_1_2_840_10008_5_1_4_1_1_30       = 320,
uid_1_2_840_10008_5_1_4_1_1_40       = 321,
uid_1_2_840_10008_5_1_4_1_1_66_6     = 322,
uid_1_2_840_10008_5_1_4_1_1_68_1     = 323,
uid_1_2_840_10008_5_1_4_1_1_68_2     = 324,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_5 = 325,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_6 = 326,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_7 = 327,
uid_1_2_840_10008_5_1_4_1_1_77_1_5_8 = 328,
uid_1_2_840_10008_5_1_4_1_1_78_1     = 329,
uid_1_2_840_10008_5_1_4_1_1_78_2     = 330,
uid_1_2_840_10008_5_1_4_1_1_78_3     = 331,
uid_1_2_840_10008_5_1_4_1_1_78_4     = 332,
uid_1_2_840_10008_5_1_4_1_1_78_5     = 333,
uid_1_2_840_10008_5_1_4_1_1_78_6     = 334,
uid_1_2_840_10008_5_1_4_1_1_78_7     = 335,
uid_1_2_840_10008_5_1_4_1_1_78_8     = 336,
uid_1_2_840_10008_5_1_4_1_1_79_1     = 337,
uid_1_2_840_10008_5_1_4_1_1_80_1     = 338,
uid_1_2_840_10008_5_1_4_1_1_81_1     = 339,
uid_1_2_840_10008_5_1_4_1_1_82_1     = 340,
uid_1_2_840_10008_5_1_4_1_1_88_34    = 341,
uid_1_2_840_10008_5_1_4_1_1_88_35    = 342,
uid_1_2_840_10008_5_1_4_1_1_88_68    = 343,
uid_1_2_840_10008_5_1_4_1_1_88_69    = 344,
uid_1_2_840_10008_5_1_4_1_1_88_70    = 345,
```

```
uid_1_2_840_10008_5_1_4_1_1_88_71     = 346,
uid_1_2_840_10008_5_1_4_1_1_88_72     = 347,
uid_1_2_840_10008_5_1_4_1_1_88_73     = 348,
uid_1_2_840_10008_5_1_4_1_1_88_74     = 349,
uid_1_2_840_10008_5_1_4_1_1_88_75     = 350,
uid_1_2_840_10008_5_1_4_1_1_90_1      = 351,
uid_1_2_840_10008_5_1_4_1_1_104_3     = 352,
uid_1_2_840_10008_5_1_4_1_1_130       = 353,
uid_1_2_840_10008_5_1_4_1_1_131       = 354,
uid_1_2_840_10008_5_1_4_1_1_200_1     = 355,
uid_1_2_840_10008_5_1_4_1_1_200_2     = 356,
uid_1_2_840_10008_5_1_4_1_1_200_3     = 357,
uid_1_2_840_10008_5_1_4_1_1_200_4     = 358,
uid_1_2_840_10008_5_1_4_1_1_200_5     = 359,
uid_1_2_840_10008_5_1_4_1_1_200_6     = 360,
uid_1_2_840_10008_5_1_4_1_1_481_10    = 361,
uid_1_2_840_10008_5_1_4_1_1_481_11    = 362,
uid_1_2_840_10008_5_1_4_1_1_501_1     = 363,
uid_1_2_840_10008_5_1_4_1_1_501_2_1   = 364,
uid_1_2_840_10008_5_1_4_1_1_501_2_2   = 365,
uid_1_2_840_10008_5_1_4_1_1_501_3     = 366,
uid_1_2_840_10008_5_1_4_1_1_501_4     = 367,
uid_1_2_840_10008_5_1_4_1_1_501_5     = 368,
uid_1_2_840_10008_5_1_4_1_1_501_6     = 369,
uid_1_2_840_10008_5_1_4_1_1_601_1     = 370,
uid_1_2_840_10008_5_1_4_1_1_601_2     = 371,
uid_1_2_840_10008_5_1_4_1_2_4_2       = 372,
uid_1_2_840_10008_5_1_4_1_2_4_3       = 373,
uid_1_2_840_10008_5_1_4_1_2_5_3       = 374,
uid_1_2_840_10008_5_1_4_20_1          = 375,
uid_1_2_840_10008_5_1_4_20_2          = 376,
uid_1_2_840_10008_5_1_4_20_3          = 377,
uid_1_2_840_10008_5_1_4_34_5_1        = 378,
uid_1_2_840_10008_5_1_4_34_6          = 379,
uid_1_2_840_10008_5_1_4_34_6_1        = 380,
uid_1_2_840_10008_5_1_4_34_6_2        = 381,
uid_1_2_840_10008_5_1_4_34_6_3        = 382,
uid_1_2_840_10008_5_1_4_34_6_4        = 383,
uid_1_2_840_10008_5_1_4_34_7          = 384,
uid_1_2_840_10008_5_1_4_34_8          = 385,
uid_1_2_840_10008_5_1_4_34_9          = 386,
uid_1_2_840_10008_5_1_4_34_10         = 387,
uid_1_2_840_10008_5_1_4_38_4          = 388,
uid_1_2_840_10008_5_1_4_39_1          = 389,
uid_1_2_840_10008_5_1_4_39_2          = 390,
uid_1_2_840_10008_5_1_4_39_3          = 391,
uid_1_2_840_10008_5_1_4_39_4          = 392,
uid_1_2_840_10008_5_1_4_43_1          = 393,
uid_1_2_840_10008_5_1_4_43_2          = 394,
uid_1_2_840_10008_5_1_4_43_3          = 395,
uid_1_2_840_10008_5_1_4_43_4          = 396,
uid_1_2_840_10008_5_1_4_44_1          = 397,
uid_1_2_840_10008_5_1_4_44_2          = 398,
uid_1_2_840_10008_5_1_4_44_3          = 399,
uid_1_2_840_10008_5_1_4_44_4          = 400,
uid_1_2_840_10008_5_1_4_45_1          = 401,
uid_1_2_840_10008_5_1_4_45_2          = 402,
uid_1_2_840_10008_5_1_4_45_3          = 403,
uid_1_2_840_10008_5_1_4_45_4          = 404,
uid_1_2_840_10008_7_1_1               = 405,
uid_1_2_840_10008_7_1_2               = 406,
uid_1_2_840_10008_8_1_1               = 407,
uid_1_2_840_10008_5_1_4_1_1_4_3       = 408,
uid_1_2_840_10008_15_1_1              = 409
//
//

//
// Optionally private UIDs
//
#if 0
uid_1_2_840_113619_4_2,
uid_1_2_840_113619_4_3,
uid_1_3_12_2_1107_5_9_1,
uid_1_2_840_113619_4_26,
uid_1_2_840_113619_4_30,
uid_2_16_840_1_113709_1_5_1,
uid_2_16_840_1_113709_1_2_2,
uid_1_2_840_113543_6_6_1_3_10002,
uid_1_2_392_200036_9116_7_8_1_1_1,
uid_1_2_392_200036_9125_1_1_2,
```

```
uid_1_2_840_113619_4_27,
uid_1_3_46_670589_11_0_0_12_1,
uid_1_3_46_670589_11_0_0_12_2,
uid_1_3_46_670589_11_0_0_12_4,
uid_1_3_46_670589_2_3_1_1,
uid_1_3_46_670589_2_4_1_1,
uid_1_3_46_670589_2_5_1_1,
uid_1_3_46_670589_5_0_1,
uid_1_3_46_670589_5_0_1_1,
uid_1_3_46_670589_5_0_10,
uid_1_3_46_670589_5_0_11,
uid_1_3_46_670589_5_0_11_1,
uid_1_3_46_670589_5_0_12,
uid_1_3_46_670589_5_0_13,
uid_1_3_46_670589_5_0_14,
uid_1_3_46_670589_5_0_2,
uid_1_3_46_670589_5_0_2_1,
uid_1_3_46_670589_5_0_3,
uid_1_3_46_670589_5_0_3_1,
uid_1_3_46_670589_5_0_4,
uid_1_3_46_670589_5_0_7,
uid_1_3_46_670589_5_0_8,
uid_1_3_46_670589_5_0_9,
uid_1_2_752_24_3_7_6,
uid_1_2_752_24_3_7_7,
uid_1_2_840_113619_5_2,
uid_1_3_46_670589_33_1_4_1
#endif
//
//

} TSType;
  typedef enum {
VerificationSOPClass = 1, // Verification SOP Class
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM = 2, // Implicit VR Little Endian: Default Transfer Syntax for DICOM
ExplicitVRLittleEndian = 3, // Explicit VR Little Endian
DeflatedExplicitVRLittleEndian = 4, // Deflated Explicit VR Little Endian
ExplicitVRBigEndian = 5, // Explicit VR Big Endian
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression = 6, // JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression
JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only = 7, // JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)
JPEGExtendedProcess35Retired = 8, // JPEG Extended (Process 3 & 5)
JPEGSpectralSelectionNonHierarchicalProcess68Retired = 9, // JPEG Spectral Selection, Non-Hierarchical (Process 6 & 8)
JPEGSpectralSelectionNonHierarchicalProcess79Retired = 10, // JPEG Spectral Selection, Non-Hierarchical (Process 7 & 9)
JPEGFullProgressionNonHierarchicalProcess1012Retired = 11, // JPEG Full Progression, Non-Hierarchical (Process 10 & 12)
JPEGFullProgressionNonHierarchicalProcess1113Retired = 12, // JPEG Full Progression, Non-Hierarchical (Process 11 & 13)
JPEGLosslessNonHierarchicalProcess14 = 13, // JPEG Lossless, Non-Hierarchical (Process 14)
JPEGLosslessNonHierarchicalProcess15Retired = 14, // JPEG Lossless, Non-Hierarchical (Process 15)
JPEGExtendedHierarchicalProcess1618Retired = 15, // JPEG Extended, Hierarchical (Process 16 & 18)
JPEGExtendedHierarchicalProcess1719Retired = 16, // JPEG Extended, Hierarchical (Process 17 & 19)
JPEGSpectralSelectionHierarchicalProcess2022Retired = 17, // JPEG Spectral Selection, Hierarchical (Process 20 & 22)
JPEGSpectralSelectionHierarchicalProcess2123Retired = 18, // JPEG Spectral Selection, Hierarchical (Process 21 & 23)
JPEGFullProgressionHierarchicalProcess2426Retired = 19, // JPEG Full Progression, Hierarchical (Process 24 & 26)
JPEGFullProgressionHierarchicalProcess2527Retired = 20, // JPEG Full Progression, Hierarchical (Process 25 & 27)
JPEGLosslessHierarchicalProcess28Retired = 21, // JPEG Lossless, Hierarchical (Process 28)
JPEGLosslessHierarchicalProcess29Retired = 22, // JPEG Lossless, Hierarchical (Process 29)

JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression = 23, // JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression
JPEGLSLosslessImageCompression = 24, // JPEG-LS Lossless Image Compression
JPEGLSLossyNearLosslessImageCompression = 25, // JPEG-LS Lossy (Near-Lossless) Image Compression
JPEG2000ImageCompressionLosslessOnly = 26, // JPEG 2000 Image Compression (Lossless Only)
JPEG2000ImageCompression = 27, // JPEG 2000 Image Compression
JPEG2000Part2MulticomponentImageCompressionLosslessOnly = 28, // JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only)
JPEG2000Part2MulticomponentImageCompression = 29, // JPEG 2000 Part 2 Multi-component Image Compression
JPIPReferenced = 30, // JPIP Referenced
JPIPReferencedDeflate = 31, // JPIP Referenced Deflate
MPEG2MainProfileMainLevel = 32, // MPEG2 Main Profile @ Main Level
RLELossless = 33, // RLE Lossless
```

```
RFC2557MIMEencapsulation = 34, // RFC 2557 MIME encapsulation
XMLEncoding = 35, // XML Encoding
MediaStorageDirectoryStorage = 36, // Media Storage Directory Storage
TalairachBrainAtlasFrameofReference = 37, // Talairach Brain Atlas Frame of Reference
SPM2T1FrameofReference = 38, // SPM2 T1 Frame of Reference
SPM2T2FrameofReference = 39, // SPM2 T2 Frame of Reference
SPM2PDFrameofReference = 40, // SPM2 PD Frame of Reference
SPM2EPIFrameofReference = 41, // SPM2 EPI Frame of Reference
SPM2FILT1FrameofReference = 42, // SPM2 FIL T1 Frame of Reference
SPM2PETFrameofReference = 43, // SPM2 PET Frame of Reference
SPM2TRANSMFrameofReference = 44, // SPM2 TRANSM Frame of Reference
SPM2SPECTFrameofReference = 45, // SPM2 SPECT Frame of Reference
SPM2GRAYFrameofReference = 46, // SPM2 GRAY Frame of Reference
SPM2WHITEFrameofReference = 47, // SPM2 WHITE Frame of Reference
SPM2CSFFrameofReference = 48, // SPM2 CSF Frame of Reference
SPM2BRAINMASKFrameofReference = 49, // SPM2 BRAINMASK Frame of Reference
SPM2AVG305T1FrameofReference = 50, // SPM2 AVG305T1 Frame of Reference
SPM2AVG152T1FrameofReference = 51, // SPM2 AVG152T1 Frame of Reference
SPM2AVG152T2FrameofReference = 52, // SPM2 AVG152T2 Frame of Reference
SPM2AVG152PDFrameofReference = 53, // SPM2 AVG152PD Frame of Reference
SPM2SINGLESUBJT1FrameofReference = 54, // SPM2 SINGLESUBJT1 Frame of Reference
ICBM452T1FrameofReference = 55, // ICBM 452 T1 Frame of Reference
ICBMSingleSubjectMRIFrameofReference = 56, // ICBM Single Subject MRI Frame of Reference
BasicStudyContentNotificationSOPClassRetired = 57, // Basic Study Content Notification SOP Class
StorageCommitmentPushModelSOPClass = 58, // Storage Commitment Push Model SOP Class
StorageCommitmentPushModelSOPInstance = 59, // Storage Commitment Push Model SOP Instance
StorageCommitmentPullModelSOPClassRetired = 60, // Storage Commitment Pull Model SOP Class
StorageCommitmentPullModelSOPInstanceRetired = 61, // Storage Commitment Pull Model SOP Instance
ProceduralEventLoggingSOPClass = 62, // Procedural Event Logging SOP Class
ProceduralEventLoggingSOPInstance = 63, // Procedural Event Logging SOP Instance
SubstanceAdministrationLoggingSOPClass = 64, // Substance Administration Logging SOP Class
SubstanceAdministrationLoggingSOPInstance = 65, // Substance Administration Logging SOP Instance
DICOMUIDRegistry = 66, // DICOM UID Registry
DICOMControlledTerminology = 67, // DICOM Controlled Terminology
DICOMApplicationContextName = 68, // DICOM Application Context Name
DetachedPatientManagementSOPClassRetired = 69, // Detached Patient Management SOP Class
DetachedPatientManagementMetaSOPClassRetired = 70, // Detached Patient Management Meta SOP Class
DetachedVisitManagementSOPClassRetired = 71, // Detached Visit Management SOP Class
DetachedStudyManagementSOPClassRetired = 72, // Detached Study Management SOP Class
StudyComponentManagementSOPClassRetired = 73, // Study Component Management SOP Class
ModalityPerformedProcedureStepSOPClass = 74, // Modality Performed Procedure Step SOP Class
ModalityPerformedProcedureStepRetrieveSOPClass = 75, // Modality Performed Procedure Step Retrieve SOP Class
ModalityPerformedProcedureStepNotificationSOPClass = 76, // Modality Performed Procedure Step Notification SOP Class
DetachedResultsManagementSOPClassRetired = 77, // Detached Results Management SOP Class
DetachedResultsManagementMetaSOPClassRetired = 78, // Detached Results Management Meta SOP Class
DetachedStudyManagementMetaSOPClassRetired = 79, // Detached Study Management Meta SOP Class
DetachedInterpretationManagementSOPClassRetired = 80, // Detached Interpretation Management SOP Class
StorageServiceClass = 81, // Storage Service Class
BasicFilmSessionSOPClass = 82, // Basic Film Session SOP Class
BasicFilmBoxSOPClass = 83, // Basic Film Box SOP Class
BasicGrayscaleImageBoxSOPClass = 84, // Basic Grayscale Image Box SOP Class
BasicColorImageBoxSOPClass = 85, // Basic Color Image Box SOP Class
ReferencedImageBoxSOPClassRetired = 86, // Referenced Image Box SOP Class
BasicGrayscalePrintManagementMetaSOPClass = 87, // Basic Grayscale Print Management Meta SOP Class
ReferencedGrayscalePrintManagementMetaSOPClassRetired = 88, // Referenced Grayscale Print Management Meta SOP Class
PrintJobSOPClass = 89, // Print Job SOP Class
BasicAnnotationBoxSOPClass = 90, // Basic Annotation Box SOP Class
PrinterSOPClass = 91, // Printer SOP Class
PrinterConfigurationRetrievalSOPClass = 92, // Printer Configuration Retrieval SOP Class
PrinterSOPInstance = 93, // Printer SOP Instance
PrinterConfigurationRetrievalSOPInstance = 94, // Printer Configuration Retrieval SOP Instance
BasicColorPrintManagementMetaSOPClass = 95, // Basic Color Print Management Meta SOP Class
ReferencedColorPrintManagementMetaSOPClassRetired = 96, // Referenced Color Print Management Meta SOP Class
VOILUTBoxSOPClass = 97, // VOI LUT Box SOP Class
PresentationLUTSOPClass = 98, // Presentation LUT SOP Class
ImageOverlayBoxSOPClassRetired = 99, // Image Overlay Box SOP Class
BasicPrintImageOverlayBoxSOPClassRetired = 100, // Basic Print Image Overlay Box SOP Class
PrintQueueSOPInstanceRetired = 101, // Print Queue SOP Instance
PrintQueueManagementSOPClassRetired = 102, // Print Queue Management SOP Class
StoredPrintStorageSOPClassRetired = 103, // Stored Print Storage SOP Class
HardcopyGrayscaleImageStorageSOPClassRetired = 104, // Hardcopy Grayscale Image Storage SOP Class
HardcopyColorImageStorageSOPClassRetired = 105, // Hardcopy Color Image Storage SOP Class
PullPrintRequestSOPClassRetired = 106, // Pull Print Request SOP Class
PullStoredPrintManagementMetaSOPClassRetired = 107, // Pull Stored Print Management Meta SOP Class
MediaCreationManagementSOPClassUID = 108, // Media Creation Management SOP Class UID
ComputedRadiographyImageStorage = 109, // Computed Radiography Image Storage
DigitalXRayImageStorageForPresentation = 110, // Digital X-Ray Image Storage - For Presentation
```

```
DigitalXRayImageStorageForProcessing = 111, // Digital X-Ray Image Storage - For Processing
DigitalMammographyXRayImageStorageForPresentation = 112, // Digital Mammography X-Ray Image Storage - For Presentation
DigitalMammographyXRayImageStorageForProcessing = 113, // Digital Mammography X-Ray Image Storage - For Processing
DigitalIntraoralXRayImageStorageForPresentation = 114, // Digital Intra-oral X-Ray Image Storage - For Presentation
DigitalIntraoralXRayImageStorageForProcessing = 115, // Digital Intra-oral X-Ray Image Storage - For Processing
CTImageStorage = 116, // CT Image Storage
EnhancedCTImageStorage = 117, // Enhanced CT Image Storage
UltrasoundMultiframeImageStorageRetired = 118, // Ultrasound Multi-frame Image Storage
UltrasoundMultiframeImageStorage = 119, // Ultrasound Multi-frame Image Storage
MRImageStorage = 120, // MR Image Storage
EnhancedMRImageStorage = 121, // Enhanced MR Image Storage
MRSpectroscopyStorage = 122, // MR Spectroscopy Storage
NuclearMedicineImageStorageRetired = 123, // Nuclear Medicine Image Storage
UltrasoundImageStorageRetired = 124, // Ultrasound Image Storage
UltrasoundImageStorage = 125, // Ultrasound Image Storage
SecondaryCaptureImageStorage = 126, // Secondary Capture Image Storage
MultiframeSingleBitSecondaryCaptureImageStorage = 127, // Multi-frame Single Bit Secondary Capture Image Storage
MultiframeGrayscaleByteSecondaryCaptureImageStorage = 128, // Multi-frame Grayscale Byte Secondary Capture Image Storage
MultiframeGrayscaleWordSecondaryCaptureImageStorage = 129, // Multi-frame Grayscale Word Secondary Capture Image Storage
MultiframeTrueColorSecondaryCaptureImageStorage = 130, // Multi-frame True Color Secondary Capture Image Storage
StandaloneOverlayStorageRetired = 131, // Standalone Overlay Storage
StandaloneCurveStorageRetired = 132, // Standalone Curve Storage
WaveformStorageTrialRetired = 133, // Waveform Storage - Trial
ECG12leadWaveformStorage = 134, // 12-lead ECG Waveform Storage
GeneralECGWaveformStorage = 135, // General ECG Waveform Storage
AmbulatoryECGWaveformStorage = 136, // Ambulatory ECG Waveform Storage
HemodynamicWaveformStorage = 137, // Hemodynamic Waveform Storage
CardiacElectrophysiologyWaveformStorage = 138, // Cardiac Electrophysiology Waveform Storage
BasicVoiceAudioWaveformStorage = 139, // Basic Voice Audio Waveform Storage
StandaloneModalityLUTStorageRetired = 140, // Standalone Modality LUT Storage
StandaloneVOILUTStorageRetired = 141, // Standalone VOI LUT Storage
GrayscaleSoftcopyPresentationStateStorageSOPClass = 142, // Grayscale Softcopy Presentation State Storage SOP Class
ColorSoftcopyPresentationStateStorageSOPClass = 143, // Color Softcopy Presentation State Storage SOP Class
PseudoColorSoftcopyPresentationStateStorageSOPClass = 144, // Pseudo-Color Softcopy Presentation State Storage SOP Class
BlendingSoftcopyPresentationStateStorageSOPClass = 145, // Blending Softcopy Presentation State Storage SOP Class
XRayAngiographicImageStorage = 146, // X-Ray Angiographic Image Storage
EnhancedXAImageStorage = 147, // Enhanced XA Image Storage
XRayRadiofluoroscopicImageStorage = 148, // X-Ray Radiofluoroscopic Image Storage
EnhancedXRFImageStorage = 149, // Enhanced XRF Image Storage
XRay3DAngiographicImageStorage = 150, // X-Ray 3D Angiographic Image Storage
XRay3DCraniofacialImageStorage = 151, // X-Ray 3D Craniofacial Image Storage
XRayAngiographicBiPlaneImageStorageRetired = 152, // X-Ray Angiographic Bi-Plane Image Storage
NuclearMedicineImageStorage = 153, // Nuclear Medicine Image Storage
RawDataStorage = 154, // Raw Data Storage
SpatialRegistrationStorage = 155, // Spatial Registration Storage
SpatialFiducialsStorage = 156, // Spatial Fiducials Storage
DeformableSpatialRegistrationStorage = 157, // Deformable Spatial Registration Storage
SegmentationStorage = 158, // Segmentation Storage
RealWorldValueMappingStorage = 159, // Real World Value Mapping Storage
VLImageStorageTrialRetired = 160, // VL Image Storage - Trial
VLMultiframeImageStorageTrialRetired = 161, // VL Multi-frame Image Storage - Trial
VLEndoscopicImageStorage = 162, // VL Endoscopic Image Storage
VideoEndoscopicImageStorage = 163, // Video Endoscopic Image Storage
VLMicroscopicImageStorage = 164, // VL Microscopic Image Storage
VideoMicroscopicImageStorage = 165, // Video Microscopic Image Storage
VLSlideCoordinatesMicroscopicImageStorage = 166, // VL Slide-Coordinates Microscopic Image Storage
VLPhotographicImageStorage = 167, // VL Photographic Image Storage
VideoPhotographicImageStorage = 168, // Video Photographic Image Storage
OphthalmicPhotography8BitImageStorage = 169, // Ophthalmic Photography 8 Bit Image Storage
OphthalmicPhotography16BitImageStorage = 170, // Ophthalmic Photography 16 Bit Image Storage
StereometricRelationshipStorage = 171, // Stereometric Relationship Storage
OphthalmicTomographyImageStorage = 172, // Ophthalmic Tomography Image Storage
TextSRStorageTrialRetired = 173, // Text SR Storage - Trial
AudioSRStorageTrialRetired = 174, // Audio SR Storage - Trial
DetailSRStorageTrialRetired = 175, // Detail SR Storage - Trial
ComprehensiveSRStorageTrialRetired = 176, // Comprehensive SR Storage - Trial
BasicTextSRStorage = 177, // Basic Text SR Storage
EnhancedSRStorage = 178, // Enhanced SR Storage
ComprehensiveSRStorage = 179, // Comprehensive SR Storage
```

```
ProcedureLogStorage = 180, // Procedure Log Storage
MammographyCADSRStorage = 181, // Mammography CAD SR Storage
KeyObjectSelectionDocumentStorage = 182, // Key Object Selection Document Storage
ChestCADSRStorage = 183, // Chest CAD SR Storage
XRayRadiationDoseSRStorage = 184, // X-Ray Radiation Dose SR Storage
EncapsulatedPDFStorage = 185, // Encapsulated PDF Storage
EncapsulatedCDAStorage = 186, // Encapsulated CDA Storage
PositronEmissionTomographyImageStorage = 187, // Positron Emission Tomography Image Storage
StandalonePETCurveStorageRetired = 188, // Standalone PET Curve Storage
RTImageStorage = 189, // RT Image Storage
RTDoseStorage = 190, // RT Dose Storage
RTStructureSetStorage = 191, // RT Structure Set Storage
RTBeamsTreatmentRecordStorage = 192, // RT Beams Treatment Record Storage
RTPlanStorage = 193, // RT Plan Storage
RTBrachyTreatmentRecordStorage = 194, // RT Brachy Treatment Record Storage
RTTreatmentSummaryRecordStorage = 195, // RT Treatment Summary Record Storage
RTIonPlanStorage = 196, // RT Ion Plan Storage
RTIonBeamsTreatmentRecordStorage = 197, // RT Ion Beams Treatment Record Storage
PatientRootQueryRetrieveInformationModelFIND = 198, // Patient Root Query/Retrieve Information Model - FIND
PatientRootQueryRetrieveInformationModelMOVE = 199, // Patient Root Query/Retrieve Information Model - MOVE
PatientRootQueryRetrieveInformationModelGET = 200, // Patient Root Query/Retrieve Information Model - GET
StudyRootQueryRetrieveInformationModelFIND = 201, // Study Root Query/Retrieve Information Model - FIND
StudyRootQueryRetrieveInformationModelMOVE = 202, // Study Root Query/Retrieve Information Model - MOVE
StudyRootQueryRetrieveInformationModelGET = 203, // Study Root Query/Retrieve Information Model - GET
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired = 204, // Patient/Study Only Query/Retrieve Information Model - FIND
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired = 205, // Patient/Study Only Query/Retrieve Information Model - MOVE
PatientStudyOnlyQueryRetrieveInformationModelGETRetired = 206, // Patient/Study Only Query/Retrieve Information Model - GET
ModalityWorklistInformationModelFIND = 207, // Modality Worklist Information Model - FIND
GeneralPurposeWorklistInformationModelFIND = 208, // General Purpose Worklist Information Model - FIND
GeneralPurposeScheduledProcedureStepSOPClass = 209, // General Purpose Scheduled Procedure Step SOP Class
GeneralPurposePerformedProcedureStepSOPClass = 210, // General Purpose Performed Procedure Step SOP Class
GeneralPurposeWorklistManagementMetaSOPClass = 211, // General Purpose Worklist Management Meta SOP Class
InstanceAvailabilityNotificationSOPClass = 212, // Instance Availability Notification SOP Class
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft = 213, // RT Beams Delivery Instruction Storage (Supplement 74 Frozen Draft)
RTConventionalMachineVerificationSupplement74FrozenDraft = 214, // RT Conventional Machine Verification (Supplement 74 Frozen Draft)
RTIonMachineVerificationSupplement74FrozenDraft = 215, // RT Ion Machine Verification (Supplement 74 Frozen Draft)
UnifiedWorklistandProcedureStepServiceClass = 216, // Unified Worklist and Procedure Step Service Class
UnifiedProcedureStepPushSOPClass = 217, // Unified Procedure Step - Push SOP Class
UnifiedProcedureStepWatchSOPClass = 218, // Unified Procedure Step - Watch SOP Class
UnifiedProcedureStepPullSOPClass = 219, // Unified Procedure Step - Pull SOP Class
UnifiedProcedureStepEventSOPClass = 220, // Unified Procedure Step - Event SOP Class
UnifiedWorklistandProcedureStepSOPInstance = 221, // Unified Worklist and Procedure Step SOP Instance
GeneralRelevantPatientInformationQuery = 222, // General Relevant Patient Information Query
BreastImagingRelevantPatientInformationQuery = 223, // Breast Imaging Relevant Patient Information Query
CardiacRelevantPatientInformationQuery = 224, // Cardiac Relevant Patient Information Query
HangingProtocolStorage = 225, // Hanging Protocol Storage
HangingProtocolInformationModelFIND = 226, // Hanging Protocol Information Model - FIND
HangingProtocolInformationModelMOVE = 227, // Hanging Protocol Information Model - MOVE
ProductCharacteristicsQuerySOPClass = 228, // Product Characteristics Query SOP Class
SubstanceApprovalQuerySOPClass = 229, // Substance Approval Query SOP Class
dicomDeviceName = 230, // dicomDeviceName
dicomDescription = 231, // dicomDescription
dicomManufacturer = 232, // dicomManufacturer
dicomManufacturerModelName = 233, // dicomManufacturerModelName
dicomSoftwareVersion = 234, // dicomSoftwareVersion
dicomVendorData = 235, // dicomVendorData
dicomAETitle = 236, // dicomAETitle
dicomNetworkConnectionReference = 237, // dicomNetworkConnectionReference
dicomApplicationCluster = 238, // dicomApplicationCluster
dicomAssociationInitiator = 239, // dicomAssociationInitiator
dicomAssociationAcceptor = 240, // dicomAssociationAcceptor
dicomHostname = 241, // dicomHostname
dicomPort = 242, // dicomPort
dicomSOPClass = 243, // dicomSOPClass
dicomTransferRole = 244, // dicomTransferRole
dicomTransferSyntax = 245, // dicomTransferSyntax
dicomPrimaryDeviceType = 246, // dicomPrimaryDeviceType
dicomRelatedDeviceReference = 247, // dicomRelatedDeviceReference
dicomPreferredCalledAETitle = 248, // dicomPreferredCalledAETitle
dicomTLSCyphersuite = 249, // dicomTLSCyphersuite
dicomAuthorizedNodeCertificateReference = 250, // dicomAuthorizedNodeCertificateReference
dicomThisNodeCertificateReference = 251, // dicomThisNodeCertificateReference
dicomInstalled = 252, // dicomInstalled
```

```
dicomStationName = 253, // dicomStationName
dicomDeviceSerialNumber = 254, // dicomDeviceSerialNumber
dicomInstitutionName = 255, // dicomInstitutionName
dicomInstitutionAddress = 256, // dicomInstitutionAddress
dicomInstitutionDepartmentName = 257, // dicomInstitutionDepartmentName
dicomIssuerOfPatientID = 258, // dicomIssuerOfPatientID
dicomPreferredCallingAETitle = 259, // dicomPreferredCallingAETitle
dicomSupportedCharacterSet = 260, // dicomSupportedCharacterSet
dicomConfigurationRoot = 261, // dicomConfigurationRoot
dicomDevicesRoot = 262, // dicomDevicesRoot
dicomUniqueAETitlesRegistryRoot = 263, // dicomUniqueAETitlesRegistryRoot
dicomDevice = 264, // dicomDevice
dicomNetworkAE = 265, // dicomNetworkAE
dicomNetworkConnection = 266, // dicomNetworkConnection
dicomUniqueAETitle = 267, // dicomUniqueAETitle
dicomTransferCapability = 268, // dicomTransferCapability
//
VLWholeSlideMicroscopyImageStorage           = 269,
EnhancedUSVolumeStorage                      = 270,
SurfaceSegmentationStorage                   = 271,
BreastTomosynthesisImageStorage              = 272,
LegacyConvertedEnhancedCTImageStorage        = 273,
LegacyConvertedEnhancedMRImageStorage        = 274,
LegacyConvertedEnhancedPETImageStorage       = 275,
MPEG2MainProfileHighLevel                    = 276,
MPEG4AVCH_264HighProfileLevel4_1             = 277,
MPEG4AVCH_264BDcompatibleHighProfileLevel4_1 = 278,

//
// 2019b
//
PETColorPaletteSOPInstance                                        = 279,
HotMetalBlueColorPaletteSOPInstance                               = 280,
PET20StepColorPaletteSOPInstance                                  = 281,
SpringColorPaletteSOPInstance                                     = 282,
SummerColorPaletteSOPInstance                                     = 283,
FallColorPaletteSOPInstance                                       = 284,
WinterColorPaletteSOPInstance                                     = 285,
Papyrus3ImplicitVRLittleEndian                                    = 286,
AdultMouseAnatomyOntology                                         = 287,
UberonOntology                                                    = 288,
IntegratedTaxonomicInformationSystemITISTaxonomicSerialNumberTSN  = 289,
MouseGenomeInitiativeMGI                                          = 290,
PubChemCompoundCID                                                = 291,
ICD11                                                             = 292,
NewYorkUniversityMelanomaClinicalCooperativeGroup                 = 293,
MayoClinicNonradiologicalImagesSBSAnatomicalSurfaceRegionGuide    = 294,
ImageBiomarkerStandardisationInitiative                           = 295,
RadiomicsOntology                                                 = 296,
DisplaySystemSOPClass                                             = 297,
DisplaySystemSOPInstance                                          = 298,
GeneralAudioWaveformStorage                                       = 299,
ArterialPulseWaveformStorage                                      = 300,
RespiratoryWaveformStorage                                        = 301,
XAXRFGrayscaleSoftcopyPresentationStateStorage                    = 302,
GrayscalePlanarMPRVolumetricPresentationStateStorage              = 303,
MPEG4AVCH_264HighProfileLevel4_2For2DVideo                        = 304,
MPEG4AVCH_264HighProfileLevel4_2For3DVideo                        = 305,
MPEG4AVCH_264StereoHighProfileLevel4_2                            = 306,
HEVCH_265MainProfileLevel5_1                                      = 307,
HEVCH_265Main10ProfileLevel5_1                                    = 308,
HotIronColorPaletteSOPInstance                                    = 309,
CompositingPlanarMPRVolumetricPresentationStateStorage            = 310,
AdvancedBlendingPresentationStateStorage                          = 311,
VolumeRenderingVolumetricPresentationStateStorage                 = 312,
SegmentedVolumeRenderingVolumetricPresentationStateStorage        = 313,
MultipleVolumeRenderingVolumetricPresentationStateStorage         = 314,
Null0                                                             = 315,
BreastProjectionXRayImageStorageForPresentation                   = 316,
BreastProjectionXRayImageStorageForProcessing                     = 317,
IntravascularOpticalCoherenceTomographyImageStorageForPresentation = 318,
IntravascularOpticalCoherenceTomographyImageStorageForProcessing  = 319,
ParametricMapStorage                                              = 320,
Null1                                                             = 321,
TractographyResultsStorage                                        = 322,
SurfaceScanMeshStorage                                            = 323,
SurfaceScanPointCloudStorage                                      = 324,
WideFieldOphthalmicPhotographyStereographicProjectionImageStorage = 325,
WideFieldOphthalmicPhotography3DCoordinatesImageStorage           = 326,
OphthalmicOpticalCoherenceTomographyEnFaceImageStorage            = 327,
OphthalmicOpticalCoherenceTomographyBscanVolumeAnalysisStorage    = 328,
```

```
LensometryMeasurementsStorage                                    = 329,
AutorefractionMeasurementsStorage                                = 330,
KeratometryMeasurementsStorage                                   = 331,
SubjectiveRefractionMeasurementsStorage                          = 332,
VisualAcuityMeasurementsStorage                                  = 333,
SpectaclePrescriptionReportStorage                               = 334,
OphthalmicAxialMeasurementsStorage                               = 335,
IntraocularLensCalculationsStorage                               = 336,
MacularGridThicknessandVolumeReportStorage                       = 337,
OphthalmicVisualFieldStaticPerimetryMeasurementsStorage          = 338,
OphthalmicThicknessMapStorage                                    = 339,
CornealTopographyMapStorage                                      = 340,
Comprehensive3DSRStorage                                         = 341,
ExtensibleSRStorage                                              = 342,
RadiopharmaceuticalRadiationDoseSRStorage                        = 343,
ColonCADSRStorage                                                = 344,
ImplantationPlanSRStorage                                        = 345,
AcquisitionContextSRStorage                                      = 346,
SimplifiedAdultEchoSRStorage                                     = 347,
PatientRadiationDoseSRStorage                                    = 348,
PlannedImagingAgentAdministrationSRStorage                       = 349,
PerformedImagingAgentAdministrationSRStorage                     = 350,
ContentAssessmentResultsStorage                                  = 351,
EncapsulatedSTLStorage                                           = 352,
EnhancedPETImageStorage                                          = 353,
BasicStructuredDisplayStorage                                    = 354,
CTDefinedProcedureProtocolStorage                                = 355,
CTPerformedProcedureProtocolStorage                              = 356,
ProtocolApprovalStorage                                          = 357,
ProtocolApprovalInformationModelFIND                             = 358,
ProtocolApprovalInformationModelMOVE                             = 359,
ProtocolApprovalInformationModelGET                              = 360,
RTPhysicianIntentStorage                                         = 361,
RTSegmentAnnotationStorage                                       = 362,
DICOSCTImageStorage                                              = 363,
DICOSDigitalXRayImageStorageForPresentation                      = 364,
DICOSDigitalXRayImageStorageForProcessing                        = 365,
DICOSThreatDetectionReportStorage                                = 366,
DICOS2DAITStorage                                                = 367,
DICOS3DAITStorage                                                = 368,
DICOSQuadrupoleResonanceQRStorage                                = 369,
EddyCurrentImageStorage                                          = 370,
EddyCurrentMultiframeImageStorage                                = 371,
CompositeInstanceRootRetrieveMOVE                                = 372,
CompositeInstanceRootRetrieveGET                                 = 373,
CompositeInstanceRetrieveWithoutBulkDataGET                      = 374,
DefinedProcedureProtocolInformationModelFIND                     = 375,
DefinedProcedureProtocolInformationModelMOVE                     = 376,
DefinedProcedureProtocolInformationModelGET                      = 377,
UPSFilteredGlobalSubscriptionSOPInstance                         = 378,
UnifiedWorklistandProcedureStepServiceClass1                     = 379,
UnifiedProcedureStepPushSOPClass1                                = 380,
UnifiedProcedureStepWatchSOPClass1                               = 381,
UnifiedProcedureStepPullSOPClass1                                = 382,
UnifiedProcedureStepEventSOPClass1                               = 383,
RTBeamsDeliveryInstructionStorage                                = 384,
RTConventionalMachineVerification                                = 385,
RTIonMachineVerification                                         = 386,
RTBrachyApplicationSetupDeliveryInstructionStorage               = 387,
HangingProtocolInformationModelGET                               = 388,
ColorPaletteStorage                                              = 389,
ColorPaletteQueryRetrieveInformationModelFIND                    = 390,
ColorPaletteQueryRetrieveInformationModelMOVE                    = 391,
ColorPaletteQueryRetrieveInformationModelGET                     = 392,
GenericImplantTemplateStorage                                    = 393,
GenericImplantTemplateInformationModelFIND                       = 394,
GenericImplantTemplateInformationModelMOVE                       = 395,
GenericImplantTemplateInformationModelGET                        = 396,
ImplantAssemblyTemplateStorage                                   = 397,
ImplantAssemblyTemplateInformationModelFIND                      = 398,
ImplantAssemblyTemplateInformationModelMOVE                      = 399,
ImplantAssemblyTemplateInformationModelGET                       = 400,
ImplantTemplateGroupStorage                                      = 401,
ImplantTemplateGroupInformationModelFIND                         = 402,
ImplantTemplateGroupInformationModelMOVE                         = 403,
ImplantTemplateGroupInformationModelGET                          = 404,
NativeDICOMModel                                                 = 405,
AbstractMultiDimensionalImageModel                               = 406,
DICOMContentMappingResource                                      = 407,
EnhancedMRColorImageStorage                                      = 408,
UniversalCoordinatedTime                                         = 409
```

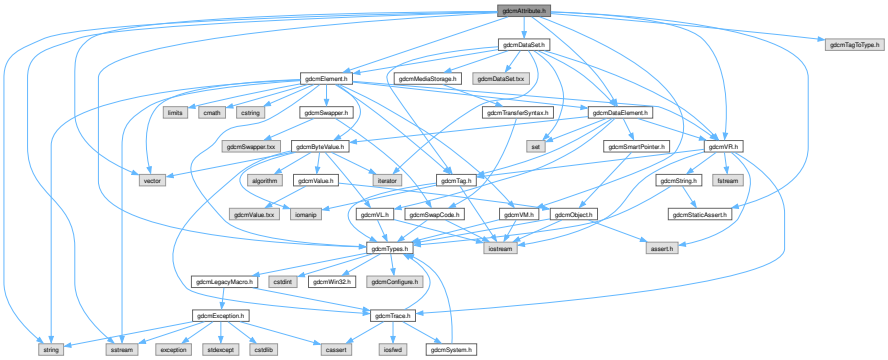
```
//
//

//
// Optionally private UIDs
//
#if 0
Private_1_2_840_113619_4_2,
Private_1_2_840_113619_4_3,
Private_1_3_12_2_1107_5_9_1,
Private_1_2_840_113619_4_26,
Private_1_2_840_113619_4_30,
Private_2_16_840_1_113709_1_5_1,
Private_2_16_840_1_113709_1_2_2,
Private_1_2_840_113543_6_6_1_3_10002,
Private_1_2_392_200036_9116_7_8_1_1_1,
Private_1_2_392_200036_9125_1_1_2,
Private_1_2_840_113619_4_27,
Private_1_3_46_670589_11_0_0_12_1,
Private_1_3_46_670589_11_0_0_12_2,
Private_1_3_46_670589_11_0_0_12_4,
Private_1_3_46_670589_2_3_1_1,
Private_1_3_46_670589_2_4_1_1,
Private_1_3_46_670589_2_5_1_1,
Private_1_3_46_670589_5_0_1,
Private_1_3_46_670589_5_0_1_1,
Private_1_3_46_670589_5_0_10,
Private_1_3_46_670589_5_0_11,
Private_1_3_46_670589_5_0_11_1,
Private_1_3_46_670589_5_0_12,
Private_1_3_46_670589_5_0_13,
Private_1_3_46_670589_5_0_14,
Private_1_3_46_670589_5_0_2,
Private_1_3_46_670589_5_0_2_1,
Private_1_3_46_670589_5_0_3,
Private_1_3_46_670589_5_0_3_1,
Private_1_3_46_670589_5_0_4,
Private_1_3_46_670589_5_0_7,
Private_1_3_46_670589_5_0_8,
Private_1_3_46_670589_5_0_9,
Private_1_2_752_24_3_7_6,
Private_1_2_752_24_3_7_7,
Private_1_2_840_113619_5_2,
Private_1_3_46_670589_33_1_4_1
#endif
//
//

} TSName;


  typedef const char* const (*TransferSyntaxStringsType)[2];
  static TransferSyntaxStringsType GetTransferSyntaxStrings();
  static const char * const *GetTransferSyntaxString(unsigned int ts);
  static unsigned int GetNumberOfTransferSyntaxStrings();


  // TODO: Because I would like a dual signature for TSType and TSName, C++ won't let me do it...
  static const char* GetUIDString(/*TSType*/ unsigned int ts);
  static const char* GetUIDName(/*TSType*/ unsigned int ts);

  bool SetFromUID(const char *str);

  const char *GetName() const;

  const char *GetString() const;

  operator TSType () const { return TSField; }

private:
  TSType TSField;
};
//-----------------------------------------------------------------------------
inline std::ostream &operator«(std::ostream &_os, const UIDs &uid)
{
  _os « uid.GetString() « " -> " « uid.GetName();
  return _os;

}

} // end namespace gdcm
```

```
01007
01008 #endif //GDCMUIDS_H
```

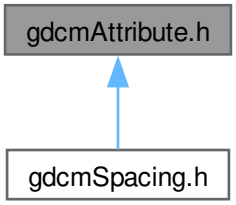
13.109 gdcmAttribute.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTagToType.h"
#include "gdcmVM.h"
#include "gdcmElement.h"
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmStaticAssert.h"
#include <string>
#include <vector>
#include <sstream>
```

Include dependency graph for gdcmAttribute.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Attribute< Group, Element, TVR, TVM >

  Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary.
- class gdcm::Attribute< Group, Element, TVR, VM::VM1 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >
- class gdcm::Attribute< Group, Element, TVR, VM::VM1_n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM2_n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >
- class gdcm::Attribute< Group, Element, TVR, VM::VM3_n >
- class gdcm::VRVLSize< 0 >
- class gdcm::VRVLSize< 1 >

Namespaces

- namespace gdcm

13.110 gdcmAttribute.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMATTRIBUTE_H
#define GDCMATTRIBUTE_H

#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTagToType.h"
#include "gdcmVM.h"
#include "gdcmElement.h"
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmStaticAssert.h"

#include <string>
#include <vector>
#include <sstream>

namespace gdcm_ns
{

struct void_;

// Declaration, also serve as forward declaration
template<int T> class VRVLSize;

// Implementation when VL is coded on 16 bits:
template<> class VRVLSize<0> {
```

```
public:
  static inline uint16_t Read(std::istream &_is) {
    uint16_t l;
    _is.read((char*)&l, 2);
    return l;
    }

  static inline void Write(std::ostream &os)  { (void)os;
    }
};
// Implementation when VL is coded on 32 bits:
template<> class VRVLSize<1> {
public:
  static inline uint32_t Read(std::istream &_is) {
    char dummy[2];
    _is.read(dummy, 2);

    uint32_t l;
    _is.read((char*)&l, 4);
    return l;
    }

  static inline void Write(std::ostream &os)  { (void)os;
    }
};

template<uint16_t Group, uint16_t Element,
   long long TVR = TagToType<Group, Element>::VRType, // can the user override this value ?
   int TVM = TagToType<Group, Element>::VMType // can the user override this value ?
   /*typename SQAttribute = void_*/ > // if only I had variadic template...
class Attribute
{
public:
  typedef typename VRToType<TVR>::Type ArrayType;
  enum { VMType = VMToLength<TVM>::Length };
  ArrayType Internal[VMToLength<TVM>::Length];

  // Make sure that user specified VR/VM are compatible with the public dictionary:
  GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
  GDCM_STATIC_ASSERT( ((VM::VMType)TVM & (VM::VMType)(TagToType<Group, Element>::VMType)) );
  GDCM_STATIC_ASSERT( ((((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)TVM == VM::VM1) )
                    || !((VR::VRType)TVR & VR::VR_VM1) ) );

  static Tag GetTag() { return Tag(Group,Element); }
  static VR  GetVR()  { return (VR::VRType)TVR; }
  static VM  GetVM()  { return (VM::VMType)TVM; }

  // The following two methods do make sense only in case of public element,
  // when the template is intanciated with private element the VR/VM are simply
  // defaulted to allow everything (see gdcmTagToType.h default template for TagToType)
  static VR  GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
  static VM  GetDictVM() { return (VM::VMType)(TagToType<Group, Element>::VMType); }

  // Some extra dummy checks:
  // Data Elements with a VR of SQ, OF, OW, OB or UN shall always have a Value Multiplicity of one.

  unsigned int GetNumberOfValues() const {
    return VMToLength<TVM>::Length;
  }
  // Implementation of Print is common to all Mode (ASCII/Binary)
  // TODO: Can we print a \ when in ASCII...well I don't think so
  // it would mean we used a bad VM then, right ?
  void Print(std::ostream &os) const {
    os « GetTag() « " ";
    os « TagToType<Group,Element>::GetVRString()  « " ";
    os « TagToType<Group,Element>::GetVMString()  « " ";
    os « Internal[0]; // VM is at least guarantee to be one
    for(unsigned int i=1; i<GetNumberOfValues(); ++i)
      os « "," « Internal[i];
    }

  // copy:
  //ArrayType GetValue(unsigned int idx = 0) {
  //  assert( idx < GetNumberOfValues() );
  //  return Internal[idx];
  //}
  //ArrayType operator[] (unsigned int idx) {
  //  return GetValue(idx);
  //}
  // FIXME: is this always a good idea ?
  // I do not think so, I prefer operator
```

```
  //operator ArrayType () const { return Internal[0]; }

  bool operator==(const Attribute &att) const
    {
    return std::equal(Internal, Internal+GetNumberOfValues(),
      att.GetValues());
    }
  bool operator!=(const Attribute &att) const
    {
    return !std::equal(Internal, Internal+GetNumberOfValues(),
      att.GetValues());
    }
  bool operator<(const Attribute &att) const
    {
    return std::lexicographical_compare(Internal, Internal+GetNumberOfValues(),
      att.GetValues(), att.GetValues() + att.GetNumberOfValues() );
    }

  ArrayType &GetValue(unsigned int idx = 0) {
    assert( idx < GetNumberOfValues() );
    return Internal[idx];
  }
  ArrayType & operator[] (unsigned int idx) {
    return GetValue(idx);
  }
  // const reference
  ArrayType const &GetValue(unsigned int idx = 0) const {
    assert( idx < GetNumberOfValues() );
    return Internal[idx];
  }
  ArrayType const & operator[] (unsigned int idx) const {
    return GetValue(idx);
  }
  void SetValue(ArrayType v, unsigned int idx = 0) {
    assert( idx < GetNumberOfValues() );
    Internal[idx] = v;
  }
  void SetValues(const ArrayType* array, unsigned int numel = VMType ) {
    assert( array && numel && numel == GetNumberOfValues() );
    // std::copy is smarted than a memcpy, and will call memcpy when POD type
    std::copy(array, array+numel, Internal);
  }
  const ArrayType* GetValues() const {
    return Internal;
  }

  // API to talk to the run-time layer: gdcm::DataElement
  DataElement GetAsDataElement() const {
    DataElement ret( GetTag() );
    std::ostringstream os;
    // os.imbue(std::locale::classic()); // This is not required AFAIK
    EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetNumberOfValues(),os);
    ret.SetVR( GetVR() );
    assert( ret.GetVR() != VR::SQ );
    if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
      {
      if( GetVR() != VR::UI )
        {
        if( os.str().size() % 2 )
          {
          os « " ";
          }
        }
      }
    VL::Type osStrSize = (VL::Type)os.str().size();
    ret.SetByteValue( os.str().c_str(), osStrSize );
    return ret;
  }

  void SetFromDataElement(DataElement const &de) {
    // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I should be ok:
    assert( Tag(Group,Element) == de.GetTag() || Group == 0x6000 || Group == 0x5000 );
    assert( GetVR() != VR::INVALID );
    assert( GetVR().Compatible( de.GetVR() ) || de.GetVR() == VR::INVALID ); // In case of VR::INVALID cannot use the & operator
    if( de.IsEmpty() ) return;
    const ByteValue *bv = de.GetByteValue();
#ifdef GDCM_WORDS_BIGENDIAN
    if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
```

```
#else
    if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
#endif
      {
      SetByteValue(bv);
      }
    else
      {
      SetByteValueNoSwap(bv);
      }
  }
  void Set(DataSet const &ds) {
    SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
  }
  void SetFromDataSet(DataSet const &ds) {
    if( ds.FindDataElement( Tag(Group,Element) ) &&
      !ds.GetDataElement( Tag(Group,Element) ).IsEmpty() )
      {
      SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
      }
  }
protected:
  void SetByteValueNoSwap(const ByteValue *bv) {
    if( !bv ) return; // That would be bad...
    assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
    //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
    //  {
    //  // always do a copy !
    //  SetValues(bv->GetPointer(), bv->GetLength());
    //  }
    //else
      {
      std::stringstream ss;
      std::string s = std::string( bv->GetPointer(), bv->GetLength() );
      ss.str( s );
      EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
        GetNumberOfValues(),ss);
      }
  }
  void SetByteValue(const ByteValue *bv) {
    if( !bv ) return; // That would be bad...
    assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
    //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
    //  {
    //  // always do a copy !
    //  SetValues(bv->GetPointer(), bv->GetLength());
    //  }
    //else
      {
      std::stringstream ss;
      std::string s = std::string( bv->GetPointer(), bv->GetLength() );
      ss.str( s );
      EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
        GetNumberOfValues(),ss);
      }
  }
#if 0 // TODO  FIXME the implicit way:
  // explicit:
  void Read(std::istream &_is) {
    const uint16_t cref[] = { Group, Element };
    uint16_t c[2];
    _is.read((char*)&c, sizeof(c));
    assert( c[0] == cref[0] && c[1] == cref[1] );
    char vr[2];
    _is.read(vr, 2); // Check consistency ?
    const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
    uint32_t l = VRVLSize< (TVR & VR::VL32) >::Read(_is);
    l /= sizeof( typename VRToType<TVR>::Type );
    return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
      l,_is);
  }
  void Write(std::ostream &_os) const {
    uint16_t c[] = { Group, Element };
    _os.write((char*)&c, 4);
    uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
    _os.write((char*)&l, 4);
    return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetLength(),_os);
    }
  void Read(std::istream &_is) {
    uint16_t cref[] = { Group, Element };
```

```
    uint16_t c[2];
    _is.read((char*)&c, 4);
    const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
    uint32_t l;
    _is.read((char*)&l, 4);
    l /= sizeof( typename VRToType<TVR>::Type );
     return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
      l,_is);
    }
  void Write(std::ostream &_os) const {
    uint16_t c[] = { Group, Element };
    _os.write((char*)&c, 4);
    uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
    _os.write((char*)&l, 4);
    return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetLength(),_os);
    }
#endif

};

template<uint16_t Group, uint16_t Element, long long TVR >
class Attribute<Group,Element,TVR,VM::VM1>
{
public:
  typedef typename VRToType<TVR>::Type ArrayType;
  enum { VMType = VMToLength<VM::VM1>::Length };
  //ArrayType Internal[VMToLength<TVM>::Length];
  ArrayType Internal;
  GDCM_STATIC_ASSERT( VMToLength<VM::VM1>::Length == 1 );

  // Make sure that user specified VR/VM are compatible with the public dictionary:
  GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
  GDCM_STATIC_ASSERT( ((VM::VMType)VM::VM1 & (VM::VMType)(TagToType<Group, Element>::VMType)) );
  GDCM_STATIC_ASSERT( ((((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)VM::VM1 == VM::VM1) )
                      || !((VR::VRType)TVR & VR::VR_VM1) ) );

  static Tag GetTag() { return Tag(Group,Element); }
  static VR  GetVR()  { return (VR::VRType)TVR; }
  static VM  GetVM()  { return (VM::VMType)VM::VM1; }

  // The following two methods do make sense only in case of public element,
  // when the template is intanciated with private element the VR/VM are simply
  // defaulted to allow everything (see gdcmTagToType.h default template for TagToType)
  static VR  GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
  static VM  GetDictVM() { return (VM::VMType)(TagToType<Group, Element>::VMType); }

  // Some extra dummy checks:
  // Data Elements with a VR of SQ, OF, OW, OB or UN shall always have a Value Multiplicity of one.

  unsigned int GetNumberOfValues() const {
    return VMToLength<VM::VM1>::Length;
  }
  // Implementation of Print is common to all Mode (ASCII/Binary)
  // TODO: Can we print a \ when in ASCII...well I don't think so
  // it would mean we used a bad VM then, right ?
  void Print(std::ostream &os) const {
    os « GetTag() « " ";
    os « TagToType<Group,Element>::GetVRString()  « " ";
    os « TagToType<Group,Element>::GetVMString()  « " ";
    os « Internal; // VM is at least guarantee to be one
  }
  // copy:
  //ArrayType GetValue(unsigned int idx = 0) {
  //  assert( idx < GetNumberOfValues() );
  //  return Internal[idx];
  //}
  //ArrayType operator[] (unsigned int idx) {
  //  return GetValue(idx);
  //}
  // FIXME: is this always a good idea ?
  // I do not think so, I prefer operator
  //operator ArrayType () const { return Internal[0]; }

  bool operator==(const Attribute &att) const
    {
    return std::equal(&Internal, &Internal+GetNumberOfValues(),
      att.GetValues());
    }
  bool operator!=(const Attribute &att) const
    {
```

```
    return !std::equal(&Internal, &Internal+GetNumberOfValues(),
      att.GetValues());
    }
  bool operator<(const Attribute &att) const
    {
    return std::lexicographical_compare(&Internal, &Internal+GetNumberOfValues(),
      att.GetValues(), att.GetValues() + att.GetNumberOfValues() );
    }

  ArrayType &GetValue() {
//    assert( idx < GetNumberOfValues() );
    return Internal;
  }
//  ArrayType & operator[] (unsigned int idx) {
//    return GetValue(idx);
//  }
  // const reference
  ArrayType const &GetValue() const {
    //assert( idx < GetNumberOfValues() );
    return Internal;
  }
  //ArrayType const & operator[] () const {
  //  return GetValue();
  //}
  void SetValue(ArrayType v) {
//    assert( idx < GetNumberOfValues() );
    Internal = v;
  }
/*  void SetValues(const ArrayType* array, unsigned int numel = VMType ) {
    assert( array && numel && numel == GetNumberOfValues() );
    // std::copy is smarted than a memcpy, and will call memcpy when POD type
    std::copy(array, array+numel, Internal);
  }
*/

  // FIXME Should we remove this function ?
  const ArrayType* GetValues() const {
    return &Internal;
  }

  // API to talk to the run-time layer: gdcm::DataElement
  DataElement GetAsDataElement() const {
    DataElement ret( Tag(Group,Element) );
    std::ostringstream os;
    // os.imbue(std::locale::classic()); // This is not required AFAIK
    EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(&Internal,
      GetNumberOfValues(),os);
    ret.SetVR( GetVR() );
    assert( ret.GetVR() != VR::SQ );
    if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
      {
      if( GetVR() != VR::UI )
        {
        if( os.str().size() % 2 )
          {
          os « " ";
          }
        }
      }
    VL::Type osStrSize = (VL::Type)os.str().size();
    ret.SetByteValue( os.str().c_str(), osStrSize );
    return ret;
  }

  void SetFromDataElement(DataElement const &de) {
    // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I should be ok:
    assert( Tag(Group,Element) == de.GetTag() || Group == 0x6000 || Group == 0x5000 );
    assert( GetVR() != VR::INVALID );
    assert( GetVR().Compatible( de.GetVR() ) || de.GetVR() == VR::INVALID ); // In case of VR::INVALID cannot use the & operator
    if( de.IsEmpty() ) return;
    const ByteValue *bv = de.GetByteValue();
#ifdef GDCM_WORDS_BIGENDIAN
    if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
#else
    if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
#endif
      {
      SetByteValue(bv);
      }
```

```
    else
      {
      SetByteValueNoSwap(bv);
      }
  }
  void Set(DataSet const &ds) {
    SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
  }
  void SetFromDataSet(DataSet const &ds) {
    if( ds.FindDataElement( Tag(Group,Element) ) &&
      !ds.GetDataElement( Tag(Group,Element) ).IsEmpty() )
      {
      SetFromDataElement( ds.GetDataElement( Tag(Group,Element) ) );
      }
  }
protected:
  void SetByteValueNoSwap(const ByteValue *bv) {
    if( !bv ) return; // That would be bad...
    assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
    //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
    //  {
    //  // always do a copy !
    //  SetValues(bv->GetPointer(), bv->GetLength());
    //  }
    //else
      {
      std::stringstream ss;
      std::string s = std::string( bv->GetPointer(), bv->GetLength() );
      ss.str( s );
      EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(&Internal,
        GetNumberOfValues(),ss);
      }
  }
  void SetByteValue(const ByteValue *bv) {
    if( !bv ) return; // That would be bad...
    assert( bv->GetPointer() && bv->GetLength() ); // [123]C element can be empty
    //if( VRToEncoding<TVR>::Mode == VR::VRBINARY )
    //  {
    //  // always do a copy !
    //  SetValues(bv->GetPointer(), bv->GetLength());
    //  }
    //else
      {
      std::stringstream ss;
      std::string s = std::string( bv->GetPointer(), bv->GetLength() );
      ss.str( s );
      EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(&Internal,
        GetNumberOfValues(),ss);
      }
  }
#if 0 // TODO  FIXME the implicit way:
  // explicit:
  void Read(std::istream &_is) {
    const uint16_t cref[] = { Group, Element };
    uint16_t c[2];
    _is.read((char*)&c, sizeof(c));
    assert( c[0] == cref[0] && c[1] == cref[1] );
    char vr[2];
    _is.read(vr, 2); // Check consistency ?
    const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
    uint32_t l = VRVLSize< (TVR & VR::VL32) >::Read(_is);
    l /= sizeof( typename VRToType<TVR>::Type );
    return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
      l,_is);
  }
  void Write(std::ostream &_os) const {
    uint16_t c[] = { Group, Element };
    _os.write((char*)&c, 4);
    uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
    _os.write((char*)&l, 4);
    return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetLength(),_os);
    }
  void Read(std::istream &_is) {
    uint16_t cref[] = { Group, Element };
    uint16_t c[2];
    _is.read((char*)&c, 4);
    const uint32_t lref = GetLength() * sizeof( typename VRToType<TVR>::Type );
    uint32_t l;
    _is.read((char*)&l, 4);
    l /= sizeof( typename VRToType<TVR>::Type );
```

```
      return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
        l,_is);
    }
  void Write(std::ostream &_os) const {
    uint16_t c[] = { Group, Element };
    _os.write((char*)&c, 4);
    uint32_t l = GetLength() * sizeof( typename VRToType<TVR>::Type );
    _os.write((char*)&l, 4);
    return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetLength(),_os);
    }
#endif

};

// No need to repeat default template arg, since primary template
// will be used to generate the default arguments
template<uint16_t Group, uint16_t Element, long long TVR >
class Attribute<Group,Element,TVR,VM::VM1_n>
{
public:
  typedef typename VRToType<TVR>::Type ArrayType;

  // Make sure that user specified VR/VM are compatible with the public dictionary:
  GDCM_STATIC_ASSERT( ((VR::VRType)TVR & (VR::VRType)(TagToType<Group, Element>::VRType)) );
  GDCM_STATIC_ASSERT( (VM::VM1_n & (VM::VMType)(TagToType<Group, Element>::VMType)) );
  GDCM_STATIC_ASSERT( ((((VR::VRType)TVR & VR::VR_VM1) && ((VM::VMType)TagToType<Group,Element>::VMType ==
VM::VM1) )
                      || !((VR::VRType)TVR & VR::VR_VM1) ) );

  static Tag GetTag() { return Tag(Group,Element); }
  static VR  GetVR()  { return (VR::VRType)TVR; }
  static VM  GetVM()  { return VM::VM1_n; }

  static VR  GetDictVR() { return (VR::VRType)(TagToType<Group, Element>::VRType); }
  static VM  GetDictVM() { return GetVM(); }

  // This the way to prevent default initialization
  explicit Attribute() { Internal=nullptr; Length=0; Own = true; }
  ~Attribute() {
    if( Own ) {
      delete[] Internal;
    }
    Internal = nullptr; // paranoid
  }

  unsigned int GetNumberOfValues() const { return Length; }

  void SetNumberOfValues(unsigned int numel)
    {
    SetValues(nullptr, numel, true);
    }

  const ArrayType* GetValues() const {
    return Internal;
  }
  void Print(std::ostream &os) const {
    os « GetTag() « " ";
    os « GetVR()  « " ";
    os « GetVM()  « " ";
    os « Internal[0]; // VM is at least guarantee to be one
    for(unsigned int i=1; i<GetNumberOfValues(); ++i)
      os « "," « Internal[i];
    }
  ArrayType &GetValue(unsigned int idx = 0) {
    assert( idx < GetNumberOfValues() );
    return Internal[idx];
  }
  ArrayType &operator[] (unsigned int idx) {
    return GetValue(idx);
  }
  // const reference
  ArrayType const &GetValue(unsigned int idx = 0) const {
    assert( idx < GetNumberOfValues() );
    return Internal[idx];
  }
  ArrayType const & operator[] (unsigned int idx) const {
    return GetValue(idx);
  }
  void SetValue(unsigned int idx, ArrayType v) {
    assert( idx < GetNumberOfValues() );
```

```
    Internal[idx] = v;
  }
  void SetValue(ArrayType v) { SetValue(0, v); }

  void SetValues(const ArrayType *array, unsigned int numel, bool own = false)
    {
    if( Internal ) // were we used before ?
      {
      // yes !
      if( Own ) delete[] Internal;
      Internal = nullptr;
      }
    Own = own;
    Length = numel;
    assert( Internal == nullptr );
    if( own ) // make a copy:
      {
      Internal = new ArrayType[numel];
      if( array && numel )
        std::copy(array, array+numel, Internal);
      }
    else // pass pointer
      {
      Internal = const_cast<ArrayType*>(array);
      }
    // postcondition
    assert( numel == GetNumberOfValues() );
    }

  DataElement GetAsDataElement() const {
    DataElement ret( GetTag() );
    std::ostringstream os;
    if( Internal )
      {
      EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
        GetNumberOfValues(),os);
      if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
        {
        if( GetVR() != VR::UI )
          {
          if( os.str().size() % 2 )
            {
            os « " ";
            }
          }
        }
      }
    ret.SetVR( GetVR() );
    assert( ret.GetVR() != VR::SQ );
    VL::Type osStrSize = (VL::Type) os.str().size();
    ret.SetByteValue( os.str().c_str(), osStrSize);
    return ret;
  }
  void SetFromDataElement(DataElement const &de) {
    // This is kind of hackish but since I do not generate other element than the first one: 0x6000 I should be ok:
    assert( GetTag() == de.GetTag() || GetTag().GetGroup() == 0x6000
      || GetTag().GetGroup() == 0x5000 );
    assert( GetVR().Compatible( de.GetVR() ) ); // In case of VR::INVALID cannot use the & operator
    assert( !de.IsEmpty() );
    const ByteValue *bv = de.GetByteValue();
    SetByteValue(bv);
  }
  void Set(DataSet const &ds) {
    SetFromDataElement( ds.GetDataElement( GetTag() ) );
  }
  void SetFromDataSet(DataSet const &ds) {
    if( ds.FindDataElement( GetTag() ) &&
      !ds.GetDataElement( GetTag() ).IsEmpty() )
      {
      SetFromDataElement( ds.GetDataElement( GetTag() ) );
      }
  }
protected:
  void SetByteValue(const ByteValue *bv) {
    assert( bv ); // FIXME
    std::stringstream ss;
    std::string s = std::string( bv->GetPointer(), bv->GetLength() );
    Length = bv->GetLength(); // HACK FIXME
    ss.str( s );
    ArrayType *internal;
```

```
    ArrayType buffer[256];
    if( bv->GetLength() < 256 )
      {
      internal = buffer;
      }
    else
      {
      internal = new ArrayType[(VL::Type)bv->GetLength()]; // over allocation
      }
    EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadComputeLength(internal, Length, ss);
    SetValues( internal, Length, true );
    if( !(bv->GetLength() < 256) )
      {
      delete[] internal;
      }
    //EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
    //  GetNumberOfValues(),ss);
  }

private:
  ArrayType *Internal;
  unsigned int Length;
  bool Own : 1;
};

template<uint16_t Group, uint16_t Element, long long TVR>
class Attribute<Group,Element,TVR,VM::VM1_3> : public Attribute<Group,Element,TVR,VM::VM1_n>
{
public:
  VM  GetVM() const { return VM::VM1_3; }
};

template<uint16_t Group, uint16_t Element, long long TVR>
class Attribute<Group,Element,TVR,VM::VM1_8> : public Attribute<Group,Element,TVR,VM::VM1_n>
{
public:
  VM  GetVM() const { return VM::VM1_8; }
};

template<uint16_t Group, uint16_t Element, long long TVR>
class Attribute<Group,Element,TVR,VM::VM2_n> : public Attribute<Group,Element,TVR,VM::VM1_n>
{
public:
  VM  GetVM() const { return VM::VM2_n; }
};

template<uint16_t Group, uint16_t Element, long long TVR>
class Attribute<Group,Element,TVR,VM::VM2_2n> : public Attribute<Group,Element,TVR,VM::VM2_n>
{
public:
  static VM  GetVM() { return VM::VM2_2n; }
};

template<uint16_t Group, uint16_t Element, long long TVR>
class Attribute<Group,Element,TVR,VM::VM3_n> : public Attribute<Group,Element,TVR,VM::VM1_n>
{
public:
  static VM  GetVM() { return VM::VM3_n; }
};

template<uint16_t Group, uint16_t Element, long long TVR>
class Attribute<Group,Element,TVR,VM::VM3_3n> : public Attribute<Group,Element,TVR,VM::VM3_n>
{
public:
  static VM  GetVM() { return VM::VM3_3n; }
};


// For particular case for ASCII string
// WARNING: This template explicitly instantiates a particular
// EncodingImplementation THEREFORE it is required to be declared after the
// EncodingImplementation is needs (doh!)
#if 0
template<int TVM>
class Attribute<TVM>
{
public:
  Attribute(const char array[])
    {
    unsigned int i = 0;
    const char sep = '\\';
```

```
    std::string sarray = array;
    std::string::size_type pos1 = 0;
    std::string::size_type pos2 = sarray.find(sep, pos1+1);
    while(pos2 != std::string::npos)
      {
      Internal[i++] = sarray.substr(pos1, pos2-pos1);
      pos1 = pos2+1;
      pos2 = sarray.find(sep, pos1+1);
      }
    Internal[i] = sarray.substr(pos1, pos2-pos1);
    // Shouldn't we do the contrary, since we know how many separators
    // (and default behavior is to discard anything after the VM declared
    assert( GetLength()-1 == i );
    }

  unsigned long GetLength() const {
    return VMToLength<TVM>::Length;
  }
  // Implementation of Print is common to all Mode (ASCII/Binary)
  void Print(std::ostream &_os) const {
    _os « Internal[0]; // VM is at least guarantee to be one
    for(int i=1; i<VMToLength<TVM>::Length; ++i)
      _os « "," « Internal[i];
    }

  void Read(std::istream &_is) {
    EncodingImplementation<VR::VRASCII>::Read(Internal, GetLength(),_is);
    }
  void Write(std::ostream &_os) const {
    EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(),_os);
    }
private:
  typename String Internal[VMToLength<TVM>::Length];
};

template< int TVM>
class Attribute<VR::PN, TVM> : public StringAttribute<TVM>
{
};
#endif

#if 0

// Implementation for the undefined length (dynamically allocated array)
template<int TVR>
class Attribute<TVR, VM::VM1_n>
{
public:
  // This the way to prevent default initialization
  explicit Attribute() { Internal=0; Length=0; }
  ~Attribute() {
    delete[] Internal;
    Internal = 0;
  }

  // Length manipulation
  // SetLength should really be protected anyway...all operation
  // should go through SetArray
  unsigned long GetLength() const { return Length; }
  typedef typename VRToType<TVR>::Type ArrayType;
  void SetLength(unsigned long len) {
    const unsigned int size = sizeof(ArrayType);
    if( len ) {
      if( len > Length ) {
        // perform realloc
        assert( (len / size) * size == len );
        ArrayType *internal = new ArrayType[len / size];
        memcpy(internal, Internal, Length * size);
        delete[] Internal;
        Internal = internal;
        }
      }
    Length = len / size;
  }

  // If save is set to zero user should not delete the pointer
  //void SetArray(const typename VRToType<TVR>::Type *array, int len, bool save = false)
  void SetArray(const ArrayType *array, unsigned long len,
    bool save = false) {
    if( save ) {
      SetLength(len); // realloc
```

```
      memcpy(Internal, array, len/*/sizeof(ArrayType)*/);
      }
    else {
      // TODO rewrite this stupid code:
      Length = len;
      //Internal = array;
      assert(0);
      }
  }
  // Implementation of Print is common to all Mode (ASCII/Binary)
  void Print(std::ostream &_os) const {
    assert( Length );
    assert( Internal );
    _os « Internal[0]; // VM is at least guarantee to be one
    const unsigned long length = GetLength() < 25 ? GetLength() : 25;
    for(unsigned long i=1; i<length; ++i)
      _os « "," « Internal[i];
    }
  void Read(std::istream &_is) {
    EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
      GetLength(),_is);
    }
  void Write(std::ostream &_os) const {
    EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetLength(),_os);
    }

  Attribute(const Attribute&_val) {
    if( this != &_val) {
      *this = _val;
      }
    }

  Attribute &operator=(const Attribute &_val) {
    Length = 0; // SYITF
    Internal = 0;
    SetArray(_val.Internal, _val.Length, true);
    return *this;
    }

private:
  typename VRToType<TVR>::Type *Internal;
  unsigned long Length; // unsigned int ??
};

//template <int TVM = VM::VM1_n>
//class Attribute<VR::OB, TVM > : public Attribute<VR::OB, VM::VM1_n> {};

// Partial specialization for derivatives of 1-n : 2-n, 3-n ...
template<int TVR>
class Attribute<TVR, VM::VM2_n> : public Attribute<TVR, VM::VM1_n>
{
public:
  typedef Attribute<TVR, VM::VM1_n> Parent;
  void SetLength(int len) {
    if( len <= 1 ) return;
    Parent::SetLength(len);
  }
};
template<int TVR>
class Attribute<TVR, VM::VM2_2n> : public Attribute<TVR, VM::VM2_n>
{
public:
  typedef Attribute<TVR, VM::VM2_n> Parent;
  void SetLength(int len) {
    if( len % 2 ) return;
    Parent::SetLength(len);
  }
};
template<int TVR>
class Attribute<TVR, VM::VM3_n> : public Attribute<TVR, VM::VM1_n>
{
public:
  typedef Attribute<TVR, VM::VM1_n> Parent;
  void SetLength(int len) {
    if( len <= 2 ) return;
    Parent::SetLength(len);
  }
};
template<int TVR>
class Attribute<TVR, VM::VM3_3n> : public Attribute<TVR, VM::VM3_n>
```

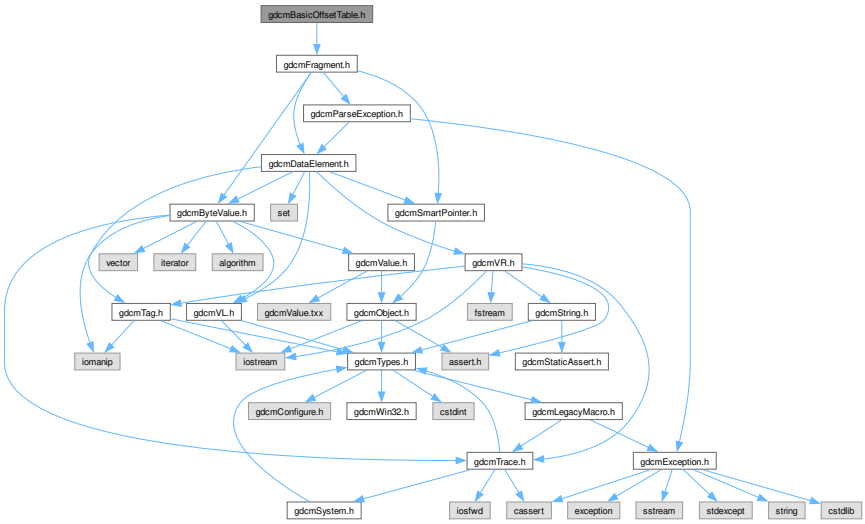
```
{
public:
  typedef Attribute<TVR, VM::VM3_n> Parent;
  void SetLength(int len) {
    if( len % 3 ) return;
    Parent::SetLength(len);
  }
};


//template<int T> struct VRToLength;
//template <> struct VRToLength<VR::AS>
//{ enum { Length  = VM::VM1 }; }
//template<>
//class Attribute<VR::AS> : public Attribute<VR::AS, VRToLength<VR::AS>::Length >

// only 0010 1010 AS 1 Patient's Age
template<>
class Attribute<VR::AS, VM::VM5>
{
public:
  char Internal[VMToLength<VM::VM5>::Length];
  void Print(std::ostream &_os) const {
    _os « Internal;
    }
};

template <>
class Attribute<VR::OB, VM::VM1> : public Attribute<VR::OB, VM::VM1_n> {};
// Make it impossible to compile any other cases:
template <int TVM> class Attribute<VR::OB, TVM>;

// Same for OW:
template <>
class Attribute<VR::OW, VM::VM1> : public Attribute<VR::OW, VM::VM1_n> {};
// Make it impossible to compile any other cases:
template <int TVM> class Attribute<VR::OW, TVM>;
#endif

#if 0
template<>
class Attribute<0x7fe0,0x0010, VR::OW, VM::VM1>
{
public:
  char *Internal;
  unsigned long Length; // unsigned int ??

  void Print(std::ostream &_os) const {
    _os « Internal[0];
    }
  void SetBytes(char *bytes, unsigned long length) {
    Internal = bytes;
    Length = length;
  }
  void Read(std::istream &_is) {
     uint16_t c[2];
    _is.read((char*)&c, 4);
    uint32_t l;
    _is.read((char*)&l, 4);
    Length = l;
    _is.read( Internal, Length );
    }
  void Write(std::ostream &_os) const {
     uint16_t c[] = {0x7fe0, 0x0010};
    _os.write((char*)&c, 4);
    _os.write((char*)&Length, 4);
    _os.write( Internal, Length );
    }
};
#endif

/*
// Removing Attribute for SQ for now...
template<uint16_t Group, uint16_t Element, typename SQA>
class Attribute<Group,Element, VR::SQ, VM::VM1, SQA>
{
public:
  SQA sqa;
  void Print(std::ostream &_os) const {
    _os « Tag(Group,Element);
    sqa.Print(_os « std::endl « '\t');
```

```
    }
 void Write(std::ostream &_os) const {
    uint16_t c[] = {Group, Element};
    _os.write((char*)&c, 4);
    uint32_t undef = 0xffffffff;
    _os.write((char*)&undef, 4);
    uint16_t item_beg[] = {0xfffe,0xe000};
    _os.write((char*)&item_beg, 4);
    _os.write((char*)&undef, 4);
    sqa.Write(_os);
    uint16_t item_end[] = {0xfffe,0xe00d};
    _os.write((char*)&item_end, 4);
    uint32_t zero = 0x0;
    _os.write((char*)&zero, 4);
    uint16_t seq_end[] = {0xfffe, 0xe0dd};
    _os.write((char*)&seq_end, 4);
    _os.write((char*)&zero, 4);
    }
};
*/


} // namespace gdcm_ns

#endif //GDCMATTRIBUTE_H
```

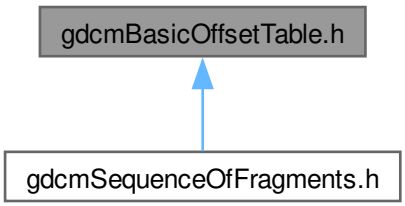
13.111 gdcmBasicOffsetTable.h File Reference

`#include "gdcmFragment.h"`

Include dependency graph for gdcmBasicOffsetTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::BasicOffsetTable

  Class to represent a BasicOffsetTable.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const BasicOffsetTable &val)

13.112 gdcmBasicOffsetTable.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMBASICOFFSETTABLE_H
#define GDCMBASICOFFSETTABLE_H

#include "gdcmFragment.h"

namespace gdcm_ns
{

```

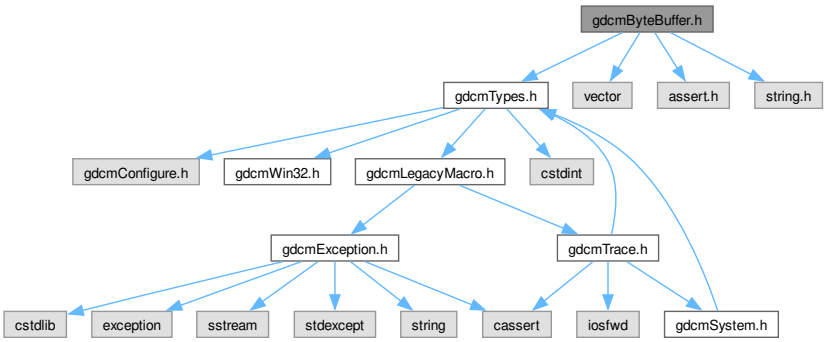
```
class GDCM_EXPORT BasicOffsetTable : public Fragment
{
//protected:
//  void SetTag(const Tag &t);
public:
  BasicOffsetTable() : Fragment() {}
  friend std::ostream &operator«(std::ostream &os, const BasicOffsetTable &val);

/*
  VL GetLength() const {
    assert( !ValueLengthField.IsUndefined() );
    assert( !ValueField || ValueField->GetLength() == ValueLengthField );
    return TagField.GetLength() + ValueLengthField.GetLength()
      + ValueLengthField;
  }
*/

  template <typename TSwap>
  std::istream &Read(std::istream &is) {
    // Superclass
    const Tag itemStart(0xfffe, 0xe000);
    if( !TagField.Read<TSwap>(is) )
      {
      assert(0 && "Should not happen");
      return is;
      }
    //assert( TagField == itemStart );
    if( TagField != itemStart )
      {
      // Bug_Siemens_PrivateIconNoItem.dcm
      //gdcmDebugMacro( "Could be Bug_Siemens_PrivateIconNoItem.dcm" );
      ParseException pe;
      pe.SetLastElement(*this);
      //throw "SIEMENS Icon thingy";
      throw pe;
      }
    if( !ValueLengthField.Read<TSwap>(is) )
      {
      assert(0 && "Should not happen");
      return is;
      }
    // Self
    SmartPointer<ByteValue> bv = new ByteValue;
    bv->SetLength(ValueLengthField);
    if( !bv->Read<TSwap>(is) )
      {
      gdcmAssertAlwaysMacro(0 && "Should not happen");
      return is;
      }
    ValueField = bv;
    return is;
    }

/*
  template <typename TSwap>
  std::ostream &Write(std::ostream &os) const {
    const Tag itemStart(0xfffe, 0xe000);
    const Tag seqDelItem(0xfffe,0xe0dd);
    if( !TagField.Write<TSwap>(os) )
      {
      assert(0 && "Should not happen");
      return os;
      }
    assert( TagField == itemStart );
    if( !ValueLengthField.Write<TSwap>(os) )
      {
      assert(0 && "Should not happen");
      return os;
      }
    if( ValueLengthField )
      {
      // Self
      const ByteValue *bv = GetByteValue();
      assert( bv );
      assert( bv->GetLength() == ValueLengthField );
      if( !bv->Write<TSwap>(os) )
        {
        assert(0 && "Should not happen");
        return os;
        }
      }
```

```
    return os;
    }
*/
};
//-----------------------------------------------------------------------------
inline std::ostream &operator«(std::ostream &os, const BasicOffsetTable &val)
{
  os « " BasicOffsetTable Length=" « val.ValueLengthField « std::endl;
  if( val.ValueField )
    {
    const ByteValue *bv = val.GetByteValue();
    assert( bv );
    os « *bv;
    }

  return os;
}


} // end namespace gdcm_ns

#endif //GDCMBASICOFFSETTABLE_H
```

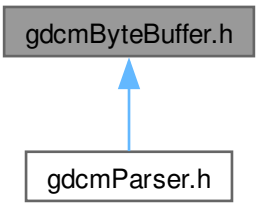
13.113 gdcmByteBuffer.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <assert.h>
#include <string.h>
```

Include dependency graph for gdcmByteBuffer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ByteBuffer

  ByteBuffer.

Namespaces

- namespace gdcm

13.114 gdcmByteBuffer.h

Go to the documentation of this file.

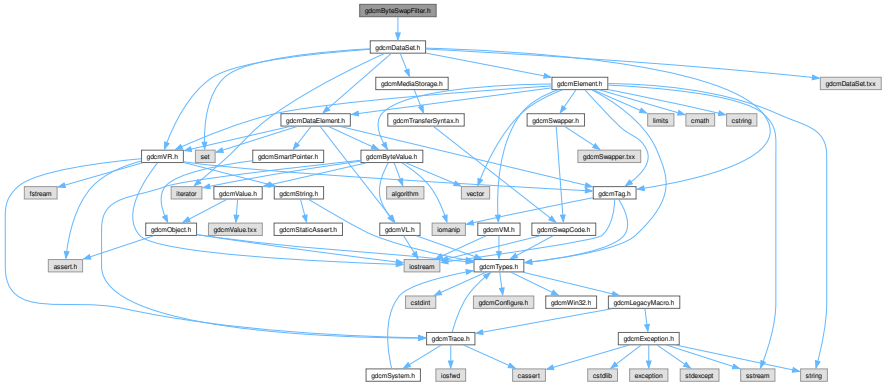
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMBYTEBUFFER_H
#define GDCMBYTEBUFFER_H

#include "gdcmTypes.h"
#include <vector>
#include <assert.h>
#include <string.h> // memmove

#error should not be used

namespace gdcm
{
class ByteBuffer
{
  static const int InitBufferSize = 1024;
public:
  ByteBuffer() : Start(0), End(0),Limit(0) {}
```

```
  char *Get(int len)
    {
    char *buffer = &Internal[0];
    if (len > Limit - End)
      {
      // FIXME avoid integer overflow
      int neededSize = len + (End - Start);
      if (neededSize  <= Limit - buffer)
        {
        memmove(buffer, Start, End - Start);
        End = buffer + (End - Start);
        Start = buffer;
        }
      else
        {
        char *newBuf;
        int bufferSize = Limit - Start;
        if ( bufferSize == 0 )
          {
          bufferSize = InitBufferSize;
          }
        do
          {
          bufferSize *= 2;
          } while (bufferSize < neededSize);
        //newBuf = malloc(bufferSize);
        try
          {
          Internal.reserve(bufferSize);
          newBuf = &Internal[0];
          }
        catch(...)
          {
          //errorCode = NoMemoryError;
          return 0;
          }
        Limit = newBuf + bufferSize;

        if (Start)
          {
          memcpy(newBuf, Start, End - Start);
          }
        End = newBuf + (End - Start);
        Start = /*buffer =*/ newBuf;
        }
      }
    assert( (int)Internal.capacity() >= len );
    return End;
    }

  void UpdatePosition() {}
  void ShiftEnd(int len) {
    End += len;
  }
  const char *GetStart() const {
    return Start;
  }

private:
  typedef std::vector<char> CharVector;
  const char *Start;
        char *End;
  const char *Limit;
  CharVector Internal;
};

} // end namespace gdcm

#endif //GDCMBYTEBUFFER_H
```

13.115 gdcmByteSwapFilter.h File Reference

`#include "gdcmDataSet.h"`

Include dependency graph for gdcmByteSwapFilter.h:



Classes

- class gdcm::ByteSwapFilter

  ByteSwapFilter.

Namespaces

- namespace gdcm

13.116 gdcmByteSwapFilter.h

Go to the documentation of this file.

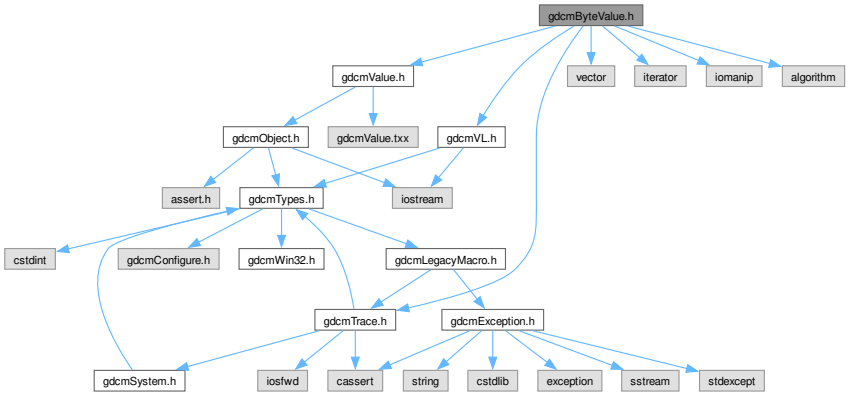
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMBYTESWAPFILTER_H
00015 #define GDCMBYTESWAPFILTER_H
00016
00017 #include "gdcmDataSet.h"
00018
00019 namespace gdcm
00020 {
00021
00027 class GDCM_EXPORT ByteSwapFilter
00028 {
```

```
public:
  ByteSwapFilter(DataSet& ds):DS(ds),ByteSwapTag(false) {}
  ~ByteSwapFilter() = default;
  ByteSwapFilter(const ByteSwapFilter &) = delete;
  ByteSwapFilter& operator=(const ByteSwapFilter &) = delete;

  bool ByteSwap();
  void SetByteSwapTag(bool b) { ByteSwapTag = b; }

private:
  DataSet &DS;
  bool ByteSwapTag;

};

} // end namespace gdcm

#endif //GDCMBYTESWAPFILTER_H
```

13.117 gdcmByteValue.h File Reference

```
#include "gdcmValue.h"
#include "gdcmTrace.h"
#include "gdcmVL.h"
#include <vector>
#include <iterator>
#include <iomanip>
#include <algorithm>
```

Include dependency graph for gdcmByteValue.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::ByteValue

  Class to represent binary value (array of bytes).

Namespaces

- namespace gdcm

13.118 gdcmByteValue.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMBYTEVALUE_H
00015 #define GDCMBYTEVALUE_H
00016
00017 #include "gdcmValue.h"
00018 #include "gdcmTrace.h"
00019 #include "gdcmVL.h"
00020
00021 #include <vector>
00022 #include <iterator>
00023 #include <iomanip>
00024 #include <algorithm>
00025
00026 namespace gdcm_ns
00027 {
00028 #if !defined(SWIGPYTHON) && !defined(SWIGCSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00029 using namespace gdcm;
00030 #endif
00034 class GDCM_EXPORT ByteValue : public Value
00035 {
00036 public:
00037   ByteValue(const char* array = nullptr, VL const &vl = 0):
00038     Internal(array, array+vl),Length(vl) {
00039       if( vl.IsOdd() )
00040         {
00041         gdcmDebugMacro( "Odd length" );
00042         Internal.resize(vl+1);
00043         ++Length;
00044         }
00045   }
00046
00048   ByteValue(std::vector<char> &v):Internal(v),Length((uint32_t)v.size()) {}
00049   //ByteValue(std::ostringstream const &os) {
00050   //  (void)os;
00051   //   assert(0); // TODO
00052   //}
00053   ~ByteValue() override {
00054     Internal.clear();
00055   }
00056
00057   // When 'dumping' dicom file we still have some information from
00058   // Either the VR: eg LO (private tag)
00059   void PrintASCII(std::ostream &os, VL maxlength ) const;
00060
00061   void PrintHex(std::ostream &os, VL maxlength) const;
00062
00063   // Either from Element Number (== 0x0000)
00064   void PrintGroupLength(std::ostream &os) {
00065     assert( Length == 2 );
00066     (void)os;
00067   }
00068
00069   bool IsEmpty() const {
00070 #if 0
```

```
    if( Internal.empty() ) assert( Length == 0 );
    return Internal.empty();
#else
  return Length == 0;
#endif
  }
  VL GetLength() const override { return Length; }

  VL ComputeLength() const { return Length + Length % 2; }
  // Does a reallocation
  void SetLength(VL vl) override;

  operator const std::vector<char>& () const { return Internal; }

  ByteValue &operator=(const ByteValue &val) {
    Internal = val.Internal;
    Length = val.Length;
    return *this;
    }

  bool operator==(const ByteValue &val) const {
    if( Length != val.Length )
      return false;
    if( Internal == val.Internal )
      return true;
    return false;
    }
  bool operator==(const Value &val) const override
    {
    const ByteValue &bv = dynamic_cast<const ByteValue&>(val);
    return Length == bv.Length && Internal == bv.Internal;
    }

  void Append(ByteValue const & bv);

  void Clear() override {
    Internal.clear();
  }
  // Use that only if you understand what you are doing
  const char *GetPointer() const {
    if(!Internal.empty()) return &Internal[0];
    return nullptr;
  }
  // Use that only if you really understand what you are doing
  const void *GetVoidPointer() const {
    if(!Internal.empty()) return &Internal[0];
    return nullptr;
  }
  void *GetVoidPointer() {
    if(!Internal.empty()) return &Internal[0];
    return nullptr;
  }
  void Fill(char c) {
    //if( Internal.empty() ) return;
    std::vector<char>::iterator it = Internal.begin();
    for(; it != Internal.end(); ++it) *it = c;
  }
  bool GetBuffer(char *buffer, unsigned long length) const;
  bool WriteBuffer(std::ostream &os) const {
    if( Length ) {
      //assert( Internal.size() <= Length );
      assert( !(Internal.size() % 2) );
      os.write(&Internal[0], Internal.size() );
      }
    return true;
  }

  template <typename TSwap, typename TType>
  std::istream &Read(std::istream &is, bool readvalues = true) {
    // If Length is odd we have detected that in SetLength
    // and calling std::vector::resize make sure to allocate *AND*
    // initialize values to 0 so we are sure to have a \0 at the end
    // even in this case
    if(Length)
      {
      if( readvalues )
        {
        is.read(&Internal[0], Length);
        assert( Internal.size() == Length || Internal.size() == Length + 1 );
        TSwap::SwapArray((TType*)GetVoidPointer(), Internal.size() / sizeof(TType) );
        }
```

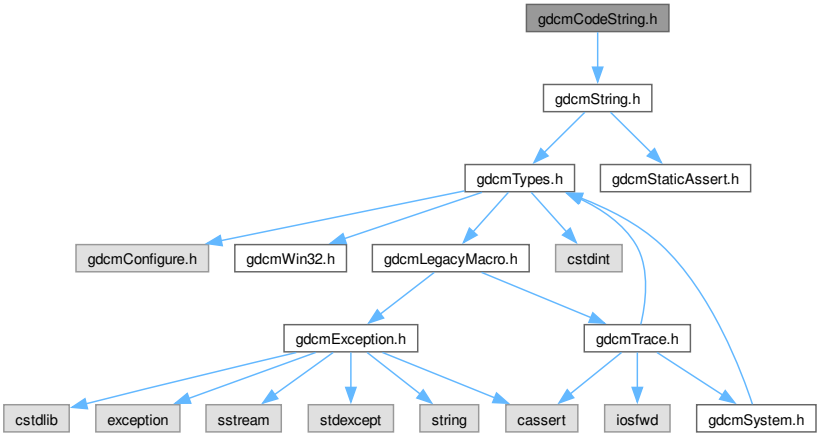
```
      else
        {
        is.seekg(Length, std::ios::cur);
        }
      }
    return is;
  }

  template <typename TSwap>
  std::istream &Read(std::istream &is) {
    return Read<TSwap,uint8_t>(is);
  }


  template <typename TSwap, typename TType>
  std::ostream const &Write(std::ostream &os) const {
    assert( !(Internal.size() % 2) );
    if( !Internal.empty() ) {
      //os.write(&Internal[0], Internal.size());
      std::vector<char> copy = Internal;
      TSwap::SwapArray((TType*)(void*)&copy[0], Internal.size() / sizeof(TType) );
      os.write(&copy[0], copy.size());
      }
    return os;
  }

  template <typename TSwap>
  std::ostream const &Write(std::ostream &os) const {
    return Write<TSwap,uint8_t>(os);
  }

  bool IsPrintable(VL length) const {
    assert( length <= Length );
    for(unsigned int i=0; i<length; i++)
      {
      if ( i == (length-1) && Internal[i] == '\0') continue;
      if ( !( isprint((unsigned char)Internal[i]) || isspace((unsigned char)Internal[i]) ) )
        {
        //gdcmWarningMacro( "Cannot print :" « i );
        return false;
        }
      }
    return true;
    }

  void PrintPNXML(std::ostream &os) const;
  void PrintASCIIXML(std::ostream &os) const;
  void PrintHexXML(std::ostream &os) const;
protected:
  void Print(std::ostream &os) const override {
  // This is perfectly valid to have a Length = 0 , so we cannot check
  // the length for printing
  if( !Internal.empty() )
    {
    if( IsPrintable(Length) )
      {
      // WARNING: Internal.end() != Internal.begin()+Length
      std::vector<char>::size_type length = Length;
      if( Internal.back() == 0 ) --length;
      std::copy(Internal.begin(), Internal.begin()+length,
        std::ostream_iterator<char>(os));
      }
    else
      os « "Loaded:" « Internal.size();
    }
  else
    {
    //os « "Not Loaded";
    os « "(no value available)";
    }
  }
/*
//Introduce check for invalid XML characters
friend std::ostream& operator«(std::ostream &os,const char c);
*/

  void SetLengthOnly(VL vl) override {
    Length = vl;
  }

private:
```

```
  std::vector<char> Internal;

  // WARNING Length IS NOT Internal.size() some *featured* DICOM
  // implementation define odd length, we always load them as even number
  // of byte, so we need to keep the right Length
  VL Length;
};

} // end namespace gdcm_ns

#endif //GDCMBYTEVALUE_H
```

13.119 gdcmCodeString.h File Reference

`#include "gdcmString.h"`

Include dependency graph for gdcmCodeString.h:



Classes

- class gdcm::CodeString

  CodeString.

Namespaces

- namespace gdcm

Functions

- bool gdcm::operator!= (const CodeString &ref, const CodeString &cs)
- std::ostream & gdcm::operator<< (std::ostream &os, const CodeString &str)
- bool gdcm::operator== (const CodeString &ref, const CodeString &cs)

13.120 gdcmCodeString.h

Go to the documentation of this file.

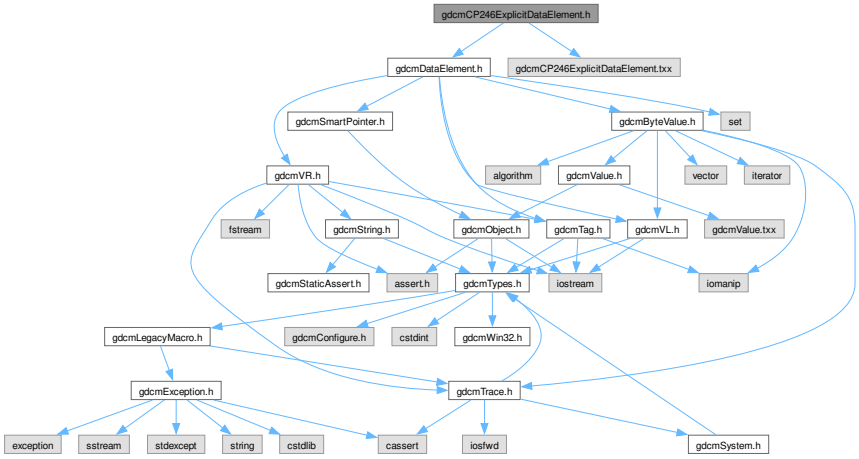
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCODESTRING_H
#define GDCMCODESTRING_H

#include "gdcmString.h"

namespace gdcm
{

// Note to myself: because note all wrapped language support exception
// we could not support throwing an exception during object construction.
class GDCM_EXPORT CodeString
{
  friend std::ostream& operator« (std::ostream& os, const CodeString& str);
  friend bool operator==(const CodeString &ref, const CodeString& cs);
  friend bool operator!=(const CodeString &ref, const CodeString& cs);
  typedef String<'\\',16> InternalClass;
public:
  typedef InternalClass::value_type             value_type;
  typedef InternalClass::pointer                pointer;
  typedef InternalClass::reference              reference;
  typedef InternalClass::const_reference        const_reference;
  typedef InternalClass::size_type              size_type;
  typedef InternalClass::difference_type        difference_type;
  typedef InternalClass::iterator               iterator;
  typedef InternalClass::const_iterator         const_iterator;
  typedef InternalClass::reverse_iterator       reverse_iterator;
  typedef InternalClass::const_reverse_iterator const_reverse_iterator;

  CodeString(): Internal() {}
  CodeString(const value_type* s): Internal(s) { Internal = Internal.Trim(); }
  CodeString(const value_type* s, size_type n): Internal(s, n) {
    Internal = Internal.Trim(); }
  CodeString(const InternalClass& s, size_type pos=0, size_type n=InternalClass::npos):
    Internal(s, pos, n) { Internal = Internal.Trim(); }

  bool IsValid() const;

  std::string GetAsString() const {
    return Internal;
  }

  size_type Size() const { return Internal.size(); }

protected:
  std::string TrimInternal() const {
    return Internal.Trim();
  }

private:
  String<'\\',16> Internal;
};

inline std::ostream& operator« (std::ostream& os, const CodeString& str)
{
  os « str.Internal;
  return os;
}

inline bool operator==(const CodeString &ref, const CodeString& cs)
{
  return ref.Internal == cs.Internal;
}
```

```
inline bool operator!=(const CodeString &ref, const CodeString& cs)
{
  return ref.Internal != cs.Internal;
}


} // end namespace gdcm

#endif //GDCMCODESTRING_H
```

13.121 gdcmCP246ExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmCP246ExplicitDataElement.txx"
```

Include dependency graph for gdcmCP246ExplicitDataElement.h:



Classes

- class gdcm::CP246ExplicitDataElement

  Class to read/write a DataElement as CP246Explicit Data Element.

Namespaces

- namespace gdcm

13.122 gdcmCP246ExplicitDataElement.h

Go to the documentation of this file.

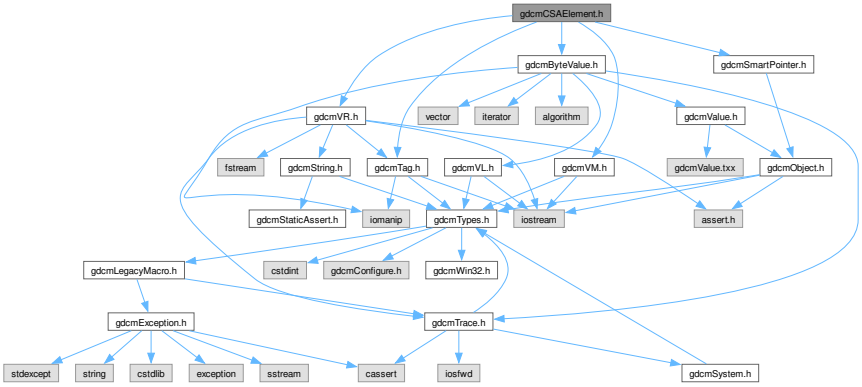
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCP246EXPLICITDATAELEMENT_H
#define GDCMCP246EXPLICITDATAELEMENT_H

#include "gdcmDataElement.h"

namespace gdcm
{
// Data Element (CP246Explicit)
class GDCM_EXPORT CP246ExplicitDataElement : public DataElement
{
public:
  VL GetLength() const;

  template <typename TSwap>
  std::istream &Read(std::istream &is);

  template <typename TSwap>
  std::istream &ReadPreValue(std::istream &is);

  template <typename TSwap>
  std::istream &ReadValue(std::istream &is, bool readvalues = true);

  template <typename TSwap>
  std::istream &ReadWithLength(std::istream &is, VL & length);

  // PURPOSELY do not provide an implementation for writing !
  //template <typename TSwap>
  //const std::ostream &Write(std::ostream &os) const;
};

} // end namespace gdcm

#include "gdcmCP246ExplicitDataElement.txx"

#endif //GDCMCP246EXPLICITDATAELEMENT_H
```

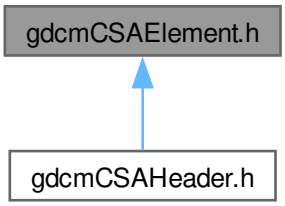
13.123 gdcmCSAElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmCSAElement.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CSAElement

  Class to represent a CSA Element.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)

13.124 gdcmCSAElement.h

Go to the documentation of this file.

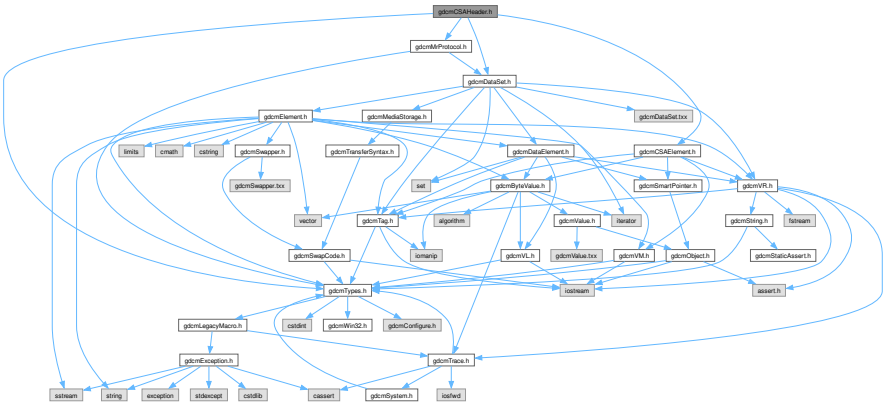
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCSAELEMENT_H
#define GDCMCSAELEMENT_H

#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

namespace gdcm
{
class GDCM_EXPORT CSAElement
{
public:
  CSAElement(unsigned int kf = 0):KeyField(kf) {}

  friend std::ostream& operator«(std::ostream &os, const CSAElement &val);

  unsigned int GetKey() const { return KeyField; }
  void SetKey(unsigned int key) { KeyField = key; }

  const char *GetName() const { return NameField.c_str(); }
  void SetName(const char *name) { NameField = name; }

  const VM& GetVM() const { return ValueMultiplicityField; }
  void SetVM(const VM &vm) { ValueMultiplicityField = vm; }

  VR const &GetVR() const { return VRField; }
  void SetVR(VR const &vr) { VRField = vr; }

  unsigned int GetSyngoDT() const { return SyngoDTField; }
  void SetSyngoDT(unsigned int syngodt) { SyngoDTField = syngodt; }

  unsigned int GetNoOfItems() const { return NoOfItemsField; }
  void SetNoOfItems(unsigned int items) { NoOfItemsField = items; }

  Value const &GetValue() const { return *DataField; }
  Value &GetValue() { return *DataField; }
  void SetValue(Value const & vl) {
    //assert( DataField == 0 );
    DataField = vl;
  }
  bool IsEmpty() const { return DataField == nullptr; }

  void SetByteValue(const char *array, VL length)
    {
    ByteValue *bv = new ByteValue(array,length);
    SetValue( *bv );
    }
  const ByteValue* GetByteValue() const {
    // Get the raw pointer from the gdcm::SmartPointer
    const ByteValue *bv = dynamic_cast<const ByteValue*>(DataField.GetPointer());
    return bv; // Will return NULL if not ByteValue
  }

  CSAElement(const CSAElement &_val)
    {
    if( this != &_val)
      {
      *this = _val;
      }
    }
```

```
  bool operator<(const CSAElement &de) const
    {
    return GetKey() < de.GetKey();
    }
  CSAElement &operator=(const CSAElement &de)
    = default;

  bool operator==(const CSAElement &de) const
    {
    return KeyField == de.KeyField
      && NameField == de.NameField
      && ValueMultiplicityField == de.ValueMultiplicityField
      && VRField == de.VRField
      && SyngoDTField == de.SyngoDTField
      //&& ValueField == de.ValueField;
      ;
    }

protected:
  unsigned int KeyField;
  std::string NameField;
  VM ValueMultiplicityField;
  VR VRField;
  unsigned int SyngoDTField;
  unsigned int NoOfItemsField;
  typedef SmartPointer<Value> DataPtr;
  DataPtr DataField;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const CSAElement &val)
{
  os « val.KeyField;
  os « " - '" « val.NameField;
  os « "' VM " « val.ValueMultiplicityField;
  os « ", VR " « val.VRField;
  os « ", SyngoDT " « val.SyngoDTField;
  os « ", NoOfItems " « val.NoOfItemsField;
  os « ", Data ";
  if( val.DataField )
    {
    //val.DataField->Print( os « "'" );
    const ByteValue * bv = dynamic_cast<ByteValue*>(&*val.DataField);
    assert( bv );
    const char * p = bv->GetPointer();
    std::string str(p, p + bv->GetLength() );
    if( val.ValueMultiplicityField == VM::VM1 )
      {
      os « "'" « str.c_str() « "'";
      }
    else
      {
      std::istringstream is( str );
      std::string s;
      bool sep = false;
      while( std::getline(is, s, '\\' ) )
        {
        if( sep )
          {
          os « '\\';
          }
        sep = true;
        os « "'" « s.c_str() « "'";
        }
      //bv->Print( os « "'" );
      //os « "'";
      }
    }
  return os;
}

} // end namespace gdcm

#endif //GDCMCSAELEMENT_H
```

13.125 gdcmCSAHeader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmCSAElement.h"
#include "gdcmMrProtocol.h"
```

Include dependency graph for gdcmCSAHeader.h:



Classes

- class gdcm::CSAHeader

  Class for CSAHeader.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)

13.126 gdcmCSAHeader.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

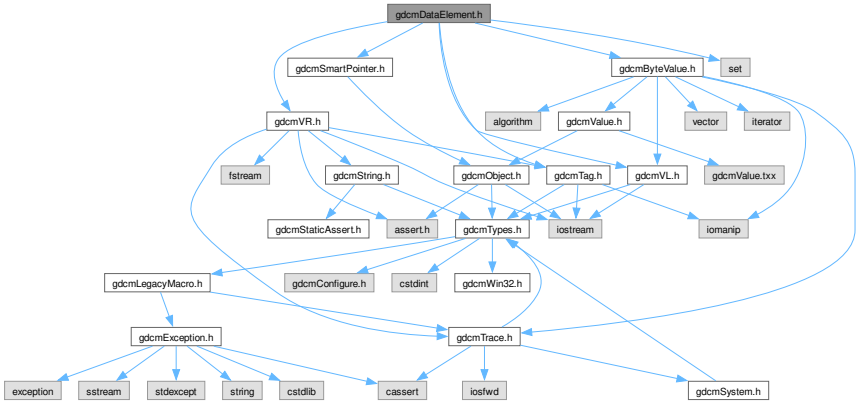
```
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCSAHEADER_H
#define GDCMCSAHEADER_H

#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmCSAElement.h"
#include "gdcmMrProtocol.h"

namespace gdcm
{
/*
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 */
//-----------------------------------------------------------------------------

class DataElement;
class PrivateTag;
class GDCM_EXPORT CSAHeader
{
  friend std::ostream& operator«(std::ostream &_os, const CSAHeader &d);
public :
  CSAHeader():InternalDataSet(),InternalType(UNKNOWN),InterfileData(nullptr) {}
  ~CSAHeader() = default;

  typedef enum {
    UNKNOWN = 0,
    SV10,
    NOMAGIC,
    DATASET_FORMAT,
    INTERFILE,
    ZEROED_OUT
  } CSAHeaderType;

  bool LoadFromDataElement(DataElement const &de);

  void Print(std::ostream &os) const;

  const DataSet& GetDataSet() const { return InternalDataSet; }

  const char * GetInterfile() const { return InterfileData; }

  CSAHeaderType GetFormat() const;

  static const PrivateTag & GetCSAImageHeaderInfoTag();

  static const PrivateTag & GetCSASeriesHeaderInfoTag();

  static const PrivateTag & GetCSADataInfo();

  const CSAElement &GetCSAElementByName(const char *name);

  bool FindCSAElementByName(const char *name);

  bool GetMrProtocol( const DataSet & ds, MrProtocol & mrProtocol );

protected:
  const CSAElement& GetCSAEEnd() const;

private:
  std::set<CSAElement> InternalCSADataSet;
  DataSet InternalDataSet;
  CSAHeaderType InternalType;
  Tag DataElementTag;
  static CSAElement CSAEEnd;
  const char *InterfileData;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const CSAHeader &d)
{
  d.Print( os );
  return os;
}

} // end namespace gdcm
```

```
00138 //-----------------------------------------------------------------------------
00139 #endif //GDCMCSAHEADER_H
```

13.127 gdcmDataElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVL.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include <set>
```

Include dependency graph for gdcmDataElement.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::DataElement

  Class to represent a Data Element either Implicit or Explicit.

Namespaces

- namespace gdcm

Functions

- bool gdcm::operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & gdcm::operator<< (std::ostream &os, const DataElement &val)

13.128 gdcmDataElement.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDATAELEMENT_H
#define GDCMDATAELEMENT_H

#include "gdcmTag.h"
#include "gdcmVL.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

#include <set>

namespace gdcm_ns
{
// Data Element
// Contains multiple fields:
// -> Tag
// -> Optional VR (Explicit Transfer Syntax)
// -> ValueLength
// -> Value
// TODO: This class SHOULD be pure virtual. I don't want a user
// to shoot himself in the foot.

class SequenceOfItems;
class SequenceOfFragments;
class GDCM_EXPORT DataElement
{
public:
  DataElement(const Tag& t = Tag(0), const VL& vl = 0, const VR &vr = VR::INVALID):TagField(t),ValueLengthField(vl),VRField(vr),ValueField(nullptr) {}
  //DataElement( Attribute const &att );

  friend std::ostream& operator«(std::ostream &_os, const DataElement &_val);

  const Tag& GetTag() const { return TagField; }
  Tag& GetTag() { return TagField; }
  void SetTag(const Tag &t) { TagField = t; }

  const VL& GetVL() const { return ValueLengthField; }
  VL& GetVL() { return ValueLengthField; }
  void SetVL(const VL &vl) { ValueLengthField = vl; }
  void SetVLToUndefined();

  VR const &GetVR() const { return VRField; }
  void SetVR(VR const &vr) {
    if( vr.IsVRFile() )
      VRField = vr;
  }

  Value const &GetValue() const { gdcmAssertAlwaysMacro(ValueField); return *ValueField; }
  Value &GetValue() {
    gdcmAssertAlwaysMacro(ValueField);
    return *ValueField;
```

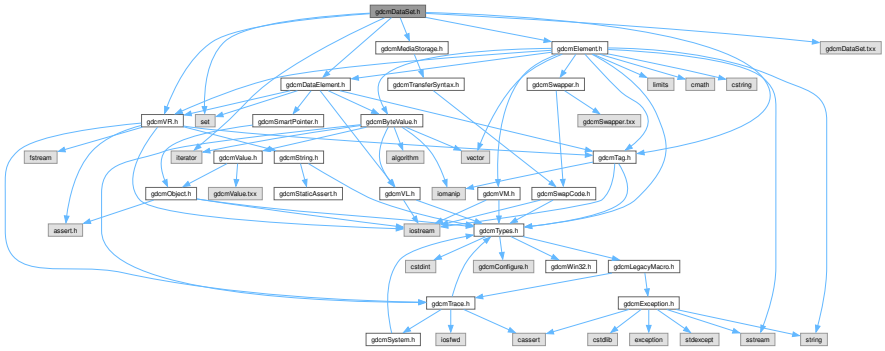
```
}
void SetValue(Value const & vl) {
  //assert( ValueField == 0 );
  ValueField = vl;
  ValueLengthField = vl.GetLength();
}
bool IsEmpty() const { return ValueField == nullptr || (GetByteValue() && GetByteValue()->IsEmpty()); }

void Empty() { ValueField = nullptr; ValueLengthField = 0; }

void Clear()
  {
  TagField = 0;
  VRField = VR::INVALID;
  ValueField = nullptr;
  ValueLengthField = 0;
  }

// Helper:
void SetByteValue(const char *array, VL length)
  {
  ByteValue *bv = new ByteValue(array,length);
  SetValue( *bv );
  }
const ByteValue* GetByteValue() const {
  // Get the raw pointer from the gdcm::SmartPointer
  const ByteValue *bv = dynamic_cast<const ByteValue*>(ValueField.GetPointer());
  return bv; // Will return NULL if not ByteValue
}

SmartPointer<SequenceOfItems> GetValueAsSQ() const;

const SequenceOfFragments* GetSequenceOfFragments() const;
SequenceOfFragments* GetSequenceOfFragments();

bool IsUndefinedLength() const {
  return ValueLengthField.IsUndefined();
}

DataElement(const DataElement &_val)
  {
  if( this != &_val)
    {
    *this = _val;
    }
  }

bool operator<(const DataElement &de) const
  {
  return GetTag() < de.GetTag();
  }
DataElement &operator=(const DataElement &)
  = default;

bool operator==(const DataElement &de) const
  {
  bool b = TagField == de.TagField
    && ValueLengthField == de.ValueLengthField
    && VRField == de.VRField;
  if( !ValueField && !de.ValueField )
    {
    return b;
    }
  if( ValueField && de.ValueField )
    {
    return b && (*ValueField == *de.ValueField);
    }
  // ValueField != de.ValueField
  return false;
  }

// The following functionalities are dependent on:
// # The Transfer Syntax: Explicit or Implicit
// # The Byte encoding: Little Endian / Big Endian

/*
 * The following was inspired by a C++ idiom: Curiously Recurring Template Pattern
 * Ref: http://en.wikipedia.org/wiki/Curiously_Recurring_Template_Pattern
 * The typename TDE is typically a derived class *without* any data
 * while TSwap is a simple template parameter to achieve byteswapping (and allow factorization of
 * highly identical code)
```

```
 */
  template <typename TDE>
  VL GetLength() const {
    return static_cast<const TDE*>(this)->GetLength();
  }

  template <typename TDE, typename TSwap>
  std::istream &Read(std::istream &is) {
    return static_cast<TDE*>(this)->template Read<TSwap>(is);
  }

  template <typename TDE, typename TSwap>
  std::istream &ReadOrSkip(std::istream &is, std::set<Tag> const &skiptags) {
    (void)skiptags;
    return static_cast<TDE*>(this)->template Read<TSwap>(is);
  }

  template <typename TDE, typename TSwap>
  std::istream &ReadPreValue(std::istream &is, std::set<Tag> const &skiptags) {
    (void)skiptags;
    return static_cast<TDE*>(this)->template ReadPreValue<TSwap>(is);
  }
  template <typename TDE, typename TSwap>
  std::istream &ReadValue(std::istream &is, std::set<Tag> const &skiptags) {
    (void)skiptags;
    return static_cast<TDE*>(this)->template ReadValue<TSwap>(is);
  }
  template <typename TDE, typename TSwap>
  std::istream &ReadValueWithLength(std::istream &is, VL & length, std::set<Tag> const &skiptags) {
    (void)skiptags;
    return static_cast<TDE*>(this)->template ReadValueWithLength<TSwap>(is, length);
  }

  template <typename TDE, typename TSwap>
  std::istream &ReadWithLength(std::istream &is, VL &length) {
    return static_cast<TDE*>(this)->template ReadWithLength<TSwap>(is,length);
  }

  template <typename TDE, typename TSwap>
  const std::ostream &Write(std::ostream &os) const {
    return static_cast<const TDE*>(this)->template Write<TSwap>(os);
  }

protected:
  Tag TagField;
  // This is the value read from the file, might be different from the length of Value Field
  VL ValueLengthField; // Can be 0xFFFFFFFF

  // Value Representation
  VR VRField;
  typedef SmartPointer<Value> ValuePtr;
  ValuePtr ValueField;

  void SetValueFieldLength( VL vl, bool readvalues );
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const DataElement &val)
{
  os « val.TagField;
  os « "\t" « val.VRField;
  os « "\t" « val.ValueLengthField;
  if( val.ValueField )
    {
    val.ValueField->Print( os « "\t" );
    }
  return os;
}

inline bool operator!=(const DataElement& lhs, const DataElement& rhs)
{
  return ! ( lhs == rhs );
}

} // end namespace gdcm_ns

#endif //GDCMDATAELEMENT_H
```

13.129 gdcmDataSet.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmTag.h"
#include "gdcmVR.h"
#include "gdcmElement.h"
#include "gdcmMediaStorage.h"
#include <set>
#include <iterator>
#include "gdcmDataSet.txx"
```

Include dependency graph for gdcmDataSet.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::DataElementException
- class gdcm::DataSet

  Class to represent a Data Set (which contains Data Elements).

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)

13.130 gdcmDataSet.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDATASET_H
#define GDCMDATASET_H

#include "gdcmDataElement.h"
#include "gdcmTag.h"
#include "gdcmVR.h"
#include "gdcmElement.h"
#include "gdcmMediaStorage.h"

#include <set>
#include <iterator>

namespace gdcm_ns
{
class GDCM_EXPORT DataElementException : public std::exception {};

class PrivateTag;
class GDCM_EXPORT DataSet
{
  friend class CSAHeader;
public:
  typedef std::set<DataElement> DataElementSet;
  typedef DataElementSet::const_iterator ConstIterator;
  typedef DataElementSet::iterator Iterator;
  typedef DataElementSet::size_type SizeType;
  //typedef typename DataElementSet::iterator iterator;
  ConstIterator Begin() const { return DES.begin(); }
  Iterator Begin() { return DES.begin(); }
  ConstIterator End() const { return DES.end(); }
  Iterator End() { return DES.end(); }
  const DataElementSet &GetDES() const { return DES; }
  DataElementSet &GetDES() { return DES; }
  void Clear() {
    DES.clear();
    assert( DES.empty() );
  }

  SizeType Size() const {
    return DES.size();
  }

  void Print(std::ostream &os, std::string const &indent = "") const {
    // CT_Phillips_JPEG2K_Decompr_Problem.dcm has a SQ of length == 0
    //int s = DES.size();
    //assert( s );
    //std::copy(DES.begin(), DES.end(),
    //  std::ostream_iterator<DataElement>(os, "\n"));
    ConstIterator it = DES.begin();
    for( ; it != DES.end(); ++it)
      {
      os « indent « *it « "\n";
      }
  }

  template <typename TDE>
  unsigned int ComputeGroupLength(Tag const &tag) const
    {
    assert( tag.GetElement() == 0x0 );
    const DataElement r(tag);
    ConstIterator it = DES.find(r);
    unsigned int res = 0;
    for( ++it; it != DES.end()
```

```
      && it->GetTag().GetGroup() == tag.GetGroup(); ++it)
      {
      assert( it->GetTag().GetElement() != 0x0 );
      assert( it->GetTag().GetGroup() == tag.GetGroup() );
      res += it->GetLength<TDE>();
      }
    return res;
    }

  template <typename TDE>
  VL GetLength() const {
    if( DES.empty() ) return 0;
    assert( !DES.empty() );
    VL ll = 0;
    assert( ll == 0 );
    ConstIterator it = DES.begin();
    for( ; it != DES.end(); ++it)
      {
      assert( !(it->GetLength<TDE>().IsUndefined()) );
      if ( it->GetTag() != Tag(0xfffe,0xe00d) )
        {
        ll += it->GetLength<TDE>();
        }
      }
    return ll;
  }
  void Insert(const DataElement& de) {
    // FIXME: there is a special case where a dataset can have value < 0x8, see:
    // $ gdcmdump --csa gdcmData/SIEMENS-JPEG-CorruptFrag.dcm
    if( de.GetTag().GetGroup() >= 0x0008 || de.GetTag().GetGroup() == 0x4 )
      {
      // prevent user error:
      if( de.GetTag() == Tag(0xfffe,0xe00d)
      || de.GetTag() == Tag(0xfffe,0xe0dd)
      || de.GetTag() == Tag(0xfffe,0xe000) )
        {
        }
      else
        {
        InsertDataElement( de );
        }
      }
    else
      {
      gdcmErrorMacro( "Cannot add element with group < 0x0008 and != 0x4 in the dataset: " « de.GetTag()
);
      }
  }
  void Replace(const DataElement& de) {
    ConstIterator it = DES.find(de);
  if( it != DES.end() )
  {
    // detect loop:
    gdcmAssertAlwaysMacro( &*it != &de );
    DES.erase(it);
  }
    DES.insert(de);
  }
  void ReplaceEmpty(const DataElement& de) {
    ConstIterator it = DES.find(de);
    if( it != DES.end() && it->IsEmpty() )
    {
      // detect loop:
    gdcmAssertAlwaysMacro( &*it != &de );
    DES.erase(it);
    }
  DES.insert(de);
  }
  SizeType Remove(const Tag& tag) {
    DataElementSet::size_type count = DES.erase(tag);
    assert( count == 0 || count == 1 );
    return count;
  }

  //DataElement& GetDataElement(const Tag &t) {
  //  DataElement r(t);
  //  Iterator it = DES.find(r);
  //  if( it != DES.end() )
  //    return *it;
  //  return GetDEEnd();
  //  }
```

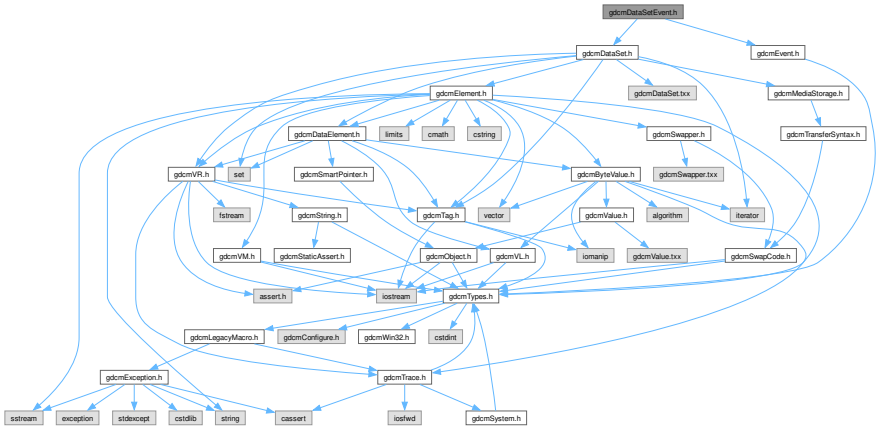
```
  const DataElement& GetDataElement(const Tag &t) const {
    const DataElement r(t);
    ConstIterator it = DES.find(r);
    if( it != DES.end() )
      return *it;
    return GetDEEnd();
    }
  const DataElement& operator[] (const Tag &t) const { return GetDataElement(t); }
  const DataElement& operator() (uint16_t group, uint16_t element) const { return GetDataElement(Tag(group,element) ); }

  std::string GetPrivateCreator(const Tag &t) const;

  PrivateTag GetPrivateTag(const Tag &t) const;

  bool FindDataElement(const PrivateTag &t) const;
  const DataElement& GetDataElement(const PrivateTag &t) const;

  // DUMB: this only search within the level of the current DataSet
  bool FindDataElement(const Tag &t) const {
    const auto it = GetDataElement(t);
    // Return if tag is found
    return it != GetDEEnd();
    }

  // WARNING:
  // This only search at the same level as the DataSet is !
  const DataElement& FindNextDataElement(const Tag &t) const {
    const DataElement r(t);
    ConstIterator it = DES.lower_bound(r);
    if( it != DES.end() )
      return *it;
    return GetDEEnd();
    }

  bool IsEmpty() const { return DES.empty(); }

  DataSet& operator=(DataSet const &)
  = default;

  template <typename TDE, typename TSwap>
  std::istream &ReadNested(std::istream &is);

  template <typename TDE, typename TSwap>
  std::istream &Read(std::istream &is);

  template <typename TDE, typename TSwap>
  std::istream &ReadUpToTag(std::istream &is, const Tag &t, std::set<Tag> const & skiptags);

  template <typename TDE, typename TSwap>
  std::istream &ReadUpToTagWithLength(std::istream &is, const Tag &t, std::set<Tag> const & skiptags, VL & length);

  template <typename TDE, typename TSwap>
  std::istream &ReadSelectedTags(std::istream &is, const std::set<Tag> & tags, bool readvalues = true);
  template <typename TDE, typename TSwap>
  std::istream &ReadSelectedTagsWithLength(std::istream &is, const std::set<Tag> & tags, VL & length, bool readvalues = true);

  template <typename TDE, typename TSwap>
  std::istream &ReadSelectedPrivateTags(std::istream &is, const std::set<PrivateTag> & tags, bool readvalues = true);
  template <typename TDE, typename TSwap>
  std::istream &ReadSelectedPrivateTagsWithLength(std::istream &is, const std::set<PrivateTag> & tags, VL & length, bool readvalues = true);

  template <typename TDE, typename TSwap>
  std::ostream const &Write(std::ostream &os) const;

  template <typename TDE, typename TSwap>
  std::istream &ReadWithLength(std::istream &is, VL &length);

  MediaStorage GetMediaStorage() const;

protected:
  /* GetDEEnd is a Win32 only issue, one cannot use a dllexported
   * static member data in an inline function, otherwise symbol
   * will get reported as missing in any dll using the inlined function
   */
  const DataElement& GetDEEnd() const;

```

```
  // This function is not safe, it does not check for the value of the tag
  // so depending whether we are getting called from a dataset or file meta header
  // the condition is different
  void InsertDataElement(const DataElement& de) {
    //if( de.GetTag() == Tag(0xfffe,0xe00d) ) return;
    //if( de.GetTag() == Tag(0xfffe,0xe0dd) ) return;
#ifndef NDEBUG
    std::pair<Iterator,bool> pr = DES.insert(de);
    if( pr.second == false )
      {
      gdcmWarningMacro( "DataElement: " « de « " was already found, skipping duplicate entry.\n"
        "Original entry kept is: " « *pr.first );
      }
#else
    DES.insert(de);
#endif
    assert( de.IsEmpty() || de.GetVL() == de.GetValue().GetLength() );
    }

protected:
  // Internal function, that will compute the actual Tag (if found) of
  // a requested Private Tag (XXXX,YY,"PRIVATE")
  Tag ComputeDataElement(const PrivateTag & t) const;

private:
  DataElementSet DES;
  static DataElement DEEnd;
  friend std::ostream& operator«(std::ostream &_os, const DataSet &);
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const DataSet &val)
{
  val.Print(os);
  return os;
}

#if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA) || defined(SWIGPHP)
/*
 * HACK: I need this temp class to be able to manipulate a std::set from python,
 * swig does not support wrapping of simple class like std::set...
 */
class SWIGDataSet
{
public:
  SWIGDataSet(DataSet &des):Internal(des),it(des.Begin()) {}
  const DataElement& GetCurrent() const { return *it; }
  void Start() { it = Internal.Begin(); }
  bool IsAtEnd() const { return it == Internal.End(); }
  void Next() { ++it; }
private:
  DataSet & Internal;
  DataSet::ConstIterator it;
};
#endif /* SWIG */


} // end namespace gdcm_ns

#include "gdcmDataSet.txx"

#endif //GDCMDATASET_H
```

13.131 gdcmDataSetEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDataSetEvent.h:



Classes

- class gdcm::DataSetEvent

  DataSetEvent.

Namespaces

- namespace gdcm

13.132 gdcmDataSetEvent.h

Go to the documentation of this file.

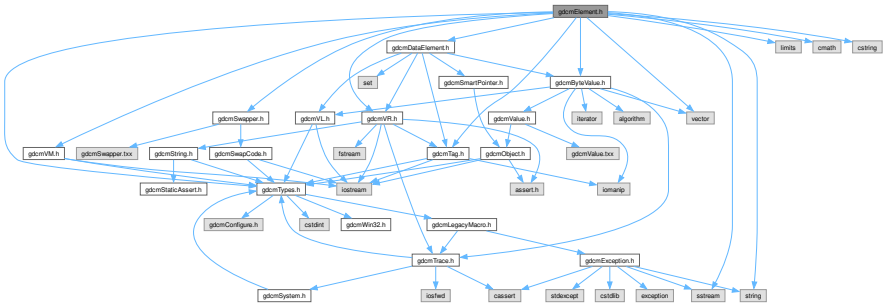
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDATASETEVENT_H
#define GDCMDATASETEVENT_H

#include "gdcmEvent.h"
#include "gdcmDataSet.h"

namespace gdcm
{

class DataSetEvent : public AnyEvent
{
public:
  typedef DataSetEvent Self;
```

```
  typedef AnyEvent Superclass;
  DataSetEvent(DataSet const *ds = nullptr):m_DataSet(ds) {}
  ~DataSetEvent() override = default;
  void operator=(const Self&) = delete;
  const DataSet *m_DataSet;
  const char * GetEventName() const override { return "DataSetEvent"; }
  bool CheckEvent(const ::gdcm::Event* e) const override
  { return (dynamic_cast<const Self*>(e) == nullptr ? false : true) ; }
  ::gdcm::Event* MakeObject() const override
    { return new Self; }
  DataSetEvent(const Self&s) : AnyEvent(s){}

  DataSet const & GetDataSet() const { return *m_DataSet; }
};


} // end namespace gdcm

#endif //GDCMANONYMIZEEVENT_H
```

13.133 gdcmElement.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmByteValue.h"
#include "gdcmDataElement.h"
#include "gdcmSwapper.h"
#include <string>
#include <vector>
#include <sstream>
#include <limits>
#include <cmath>
#include <cstring>
```

Include dependency graph for gdcmElement.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Element< TVR, TVM >

  Element class.
- class gdcm::Element< TVR, VM::VM1_2 >
- class gdcm::Element< TVR, VM::VM1_n >
- class gdcm::Element< TVR, VM::VM2_2n >
- class gdcm::Element< TVR, VM::VM2_n >
- class gdcm::Element< TVR, VM::VM3_3n >
- class gdcm::Element< TVR, VM::VM3_4 >
- class gdcm::Element< TVR, VM::VM3_n >
- class gdcm::Element< VR::AS, VM::VM5 >
- class gdcm::Element< VR::OB, VM::VM1 >
- class gdcm::Element< VR::OW, VM::VM1 >
- class gdcm::ElementDisableCombinations< TVR, TVM >

  A class which is used to produce compile errors for an invalid combination of template parameters.
- class gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >
- class gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >
- class gdcm::EncodingImplementation< VR::VRASCII >
- class gdcm::EncodingImplementation< VR::VRBINARY >
- struct gdcm::ignore_char

Namespaces

- namespace gdcm

Functions

- static int gdcm::add1 (char *buf, int n)
- ignore_char const gdcm::backslash ('\\')
- static void gdcm::clean (char *mant)
- static int gdcm::doround (char *buf, unsigned int n)
- std::istream & gdcm::operator>> (std::istream &in, ignore_char const &ic)
- static int gdcm::roundat (char *buf, size_t bufLen, unsigned int i, int iexp)
- template<typename Float>
  static void gdcm::x16printf (char *buf, int size, Float f)

13.134 gdcmElement.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMELEMENT_H
#define GDCMELEMENT_H

#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmByteValue.h"
#include "gdcmDataElement.h"
#include "gdcmSwapper.h"

#include <string>
#include <vector>
#include <sstream>
#include <limits>
#include <cmath>
#include <cstring>

namespace gdcm_ns
{

// Forward declaration
template<long long T> class EncodingImplementation;


template <long long TVR, int TVM>
class ElementDisableCombinations {};
template <>
class  ElementDisableCombinations<VR::OB, VM::VM1_n> {};
template <>
class  ElementDisableCombinations<VR::OW, VM::VM1_n> {};
// Make it impossible to compile these other cases
template <int TVM>
class  ElementDisableCombinations<VR::OB, TVM>;
template <int TVM>
class ElementDisableCombinations<VR::OW, TVM>;

template<long long TVR, int TVM>
class Element
{
  enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, TVM> ) };
public:
  typename VRToType<TVR>::Type Internal[VMToLength<TVM>::Length];
  typedef typename VRToType<TVR>::Type Type;

  static VR  GetVR()  { return (VR::VRType)TVR; }
  static VM  GetVM()  { return (VM::VMType)TVM; }

  unsigned long GetLength() const {
    return VMToLength<TVM>::Length;
  }
  // Implementation of Print is common to all Mode (ASCII/Binary)
  // TODO: Can we print a \ when in ASCII...well I don't think so
  // it would mean we used a bad VM then, right?
  void Print(std::ostream &_os) const {
    _os « Internal[0]; // VM is at least guarantee to be one
    for(int i=1; i<VMToLength<TVM>::Length; ++i)
      _os « "," « Internal[i];
    }

  const typename VRToType<TVR>::Type *GetValues() const {
    return Internal;
  }
  const typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) const {
    assert( idx < VMToLength<TVM>::Length );
    return Internal[idx];
  }
  typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) {
    assert( idx < VMToLength<TVM>::Length );
    return Internal[idx];
  }
  typename VRToType<TVR>::Type operator[] (unsigned int idx) const {
    return GetValue(idx);
  }
  void SetValue(typename VRToType<TVR>::Type v, unsigned int idx = 0) {
    assert( idx < VMToLength<TVM>::Length );
    Internal[idx] = v;
  }
```

```
  void SetFromDataElement(DataElement const &de) {
    const ByteValue *bv = de.GetByteValue();
    if( !bv ) return;
#ifdef GDCM_WORDS_BIGENDIAN
    if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
#else
    if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
#endif
      {
      Set(de.GetValue());
      }
    else
      {
      SetNoSwap(de.GetValue());
      }
  }

  DataElement GetAsDataElement() const {
    DataElement ret;
    std::ostringstream os;
    EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetLength(),os);
    ret.SetVR( (VR::VRType)TVR );
    assert( ret.GetVR() != VR::SQ );
    if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
      {
      if( GetVR() != VR::UI )
        {
        if( os.str().size() % 2 )
          {
          os « " ";
          }
        }
      }
    VL::Type osStrSize = (VL::Type)os.str().size();
    ret.SetByteValue( os.str().c_str(), osStrSize );

    return ret;
  }

  void Read(std::istream &_is) {
    return EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
      GetLength(),_is);
    }
  void Write(std::ostream &_os) const {
    return EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetLength(),_os);
    }

  // FIXME: remove this function
  // this is only used in gdcm::SplitMosaicFilter / to pass value of a CSAElement
  void Set(Value const &v) {
    const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
    if( bv ) {
      //memcpy(Internal, bv->GetPointer(), bv->GetLength());
      std::stringstream ss;
      std::string s = std::string( bv->GetPointer(), bv->GetLength() );
      ss.str( s );
      EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
        GetLength(),ss);
    }
  }
protected:
  void SetNoSwap(Value const &v) {
    const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
    assert( bv ); // That would be bad...
    //memcpy(Internal, bv->GetPointer(), bv->GetLength());
    std::stringstream ss;
    std::string s = std::string( bv->GetPointer(), bv->GetLength() );
    ss.str( s );
    EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
      GetLength(),ss);
  }
};

struct ignore_char {
  ignore_char(char c): m_char(c) {}
  char m_char;
};
ignore_char const backslash('\\');
```

```
  inline std::istream& operator» (std::istream& in, ignore_char const& ic) {
    if (!in.eof())
      in.clear(in.rdstate() & ~std::ios_base::failbit);
    if (in.get() != ic.m_char)
      in.setstate(std::ios_base::failbit);
    return in;
  }


// Implementation to perform formatted read and write
template<> class EncodingImplementation<VR::VRASCII> {
public:
  template<typename T> // FIXME this should be VRToType<TVR>::Type
  static inline void ReadComputeLength(T* data, unsigned int &length,
                          std::istream &_is) {
    assert( data );
    //assert( length ); // != 0
    length = 0;
    assert( _is );
#if 0
    char sep;
    while( _is » data[length++] )
      {
      // Get the separator in between the values
      assert( _is );
      _is.get(sep);
      assert( sep == '\\' || sep == ' ' ); // FIXME: Bad use of assert
      if( sep == ' ' ) length--; // FIXME
      }
#else
  while( _is » std::ws » data[length++] » std::ws » backslash )
    {
    }
#endif
    }

  template<typename T> // FIXME this should be VRToType<TVR>::Type
  static inline void Read(T* data, unsigned long length,
                          std::istream &_is) {
    assert( data );
    assert( length ); // != 0
    assert( _is );
    // FIXME BUG: what if » operation fails ?
    // gdcmData/MR00010001.dcm / SpacingBetweenSlices
    _is » std::ws » data[0];
    char sep;
    //std::cout « "GetLength: " « af->GetLength() « std::endl;
    for(unsigned long i=1; i<length;++i) {
      //assert( _is );
      // Get the separator in between the values
      _is » std::ws » sep; //_is.get(sep);
      //assert( sep == '\\' ); // FIXME: Bad use of assert
      _is » std::ws » data[i];
      }
    }

  template<typename T>
  static inline void ReadNoSwap(T* data, unsigned long length,
                          std::istream &_is) {
    Read(data,length,_is);
}
  template<typename T>
  static inline void Write(const T* data, unsigned long length,
                           std::ostream &_os)  {
    assert( data );
    assert( length );
    assert( _os );
    _os « data[0];
    for(unsigned long i=1; i<length; ++i) {
      assert( _os );
      _os « "\\" « data[i];
      }
    }
};

//#define VRDS16ILLEGAL

#ifdef VRDS16ILLEGAL
template < typename Float >
std::string to_string ( Float data ) {
```

```
  std::stringstream in;
  // in.imbue(std::locale::classic()); // This is not required AFAIK
  int const digits =
    static_cast< int >(
    - std::log( std::numeric_limits<Float>::epsilon() )
    / static_cast< Float >( std::log( 10.0 ) ) );
  if ( in « std::dec « std::setprecision(/*2+*/digits) « data ) {
    return ( in.str() );
  } else {
    throw "Impossible Conversion"; // should not happen ...
  }
}
#else
//
http://stackoverflow.com/questions/32631178/writing-ieee-754-1985-double-as-ascii-on-a-limited-16-bytes-string

static inline void clean(char *mant) {
  char *ix = mant + strlen(mant) - 1;
  while(('0' == *ix) && (ix > mant)) {
    *ix-- = '\0';
  }
  if ('.' == *ix) {
    *ix = '\0';
  }
}

static int add1(char *buf, int n) {
  if (n < 0) return 1;
  if (buf[n] == '9') {
    buf[n] = '0';
    return add1(buf, n-1);
  }
  else {
    buf[n] = (char)(buf[n] + 1);
  }
  return 0;
}

static int doround(char *buf, unsigned int n) {
  char c;
  if (n >= strlen(buf)) return 0;
  c = buf[n];
  buf[n] = 0;
  if ((c >= '5') && (c <= '9')) return add1(buf, n-1);
  return 0;
}

#if defined(_MSC_VER) && (_MSC_VER < 1900)
#define snprintf _snprintf
#endif

static int roundat(char *buf, size_t bufLen, unsigned int i, int iexp) {
  if (doround(buf, i) != 0) {
    iexp += 1;
    switch(iexp) {
    case -2:
      strcpy(buf, ".01");
      break;
    case -1:
      strcpy(buf, ".1");
      break;
    case 0:
      strcpy(buf, "1.");
      break;
    case 1:
      strcpy(buf, "10");
      break;
    case 2:
      strcpy(buf, "100");
      break;
    default:
      snprintf(buf, bufLen, "1e%d", iexp);
    }
    return 1;
  }
  return 0;
}

template < typename Float >
static void x16printf(char *buf, int size, Float f) {
  char line[40];
```

```
  char *mant = line + 1;
  int iexp, lexp, i;
  char exp[6];

  if (f < 0) {
    f = -f;
    size -= 1;
    *buf++ = '-';
  }
  snprintf(line, sizeof(line), "%1.16e", f);
  if (line[0] == '-') {
    f = -f;
    size -= 1;
    *buf++ = '-';
    snprintf(line, sizeof(line), "%1.16e", f);
  }
  *mant = line[0];
  i = (int)strcspn(mant, "eE");
  mant[i] = '\0';
  iexp = (int)strtol(mant + i + 1, nullptr, 10);
  lexp = snprintf(exp, sizeof(exp), "e%d", iexp);
  if ((iexp >= size) || (iexp < -3)) {
    i = roundat(mant, sizeof(line) - 1, size - 1 -lexp, iexp);
    if(i == 1) {
      strcpy(buf, mant);
      return;
    }
    buf[0] = mant[0];
    buf[1] = '.';
    strncpy(buf + i + 2, mant + 1, size - 2 - lexp);
    buf[size-lexp] = 0;
    clean(buf);
    strcat(buf, exp);
  }
  else if (iexp >= size - 2) {
    roundat(mant, sizeof(line) - 1, iexp + 1, iexp);
    strcpy(buf, mant);
  }
  else if (iexp >= 0) {
    i = roundat(mant, sizeof(line) - 1, size - 1, iexp);
    if (i == 1) {
      strcpy(buf, mant);
      return;
    }
    strncpy(buf, mant, iexp + 1);
    buf[iexp + 1] = '.';
    strncpy(buf + iexp + 2, mant + iexp + 1, size - iexp - 1);
    buf[size] = 0;
    clean(buf);
  }
  else {
    int j;
    i = roundat(mant, sizeof(line) - 1, size + 1 + iexp, iexp);
    if (i == 1) {
      strcpy(buf, mant);
      return;
    }
    buf[0] = '.';
    for(j=0; j< -1 - iexp; j++) {
      buf[j+1] = '0';
    }
    strncpy(buf - iexp, mant, size + 1 + iexp);
    buf[size] = 0;
    clean(buf);
  }
}
#if defined(_MSC_VER) && (_MSC_VER < 1900)
#undef snprintf
#endif

#endif

template<> inline void EncodingImplementation<VR::VRASCII>::Write(const double* data, unsigned long
length, std::ostream &_os)  {
    assert( data );
    assert( length );
    assert( _os );
#ifdef VRDS16ILLEGAL
    _os « to_string(data[0]);
#else
    char buf[16+1];
```

```
    x16printf(buf, 16, data[0]);
    _os « buf;
#endif
    for(unsigned long i=1; i<length; ++i) {
      assert( _os );
#ifdef VRDS16ILLEGAL
      _os « "\\" « to_string(data[i]);
#else
      x16printf(buf, 16, data[i]);
      _os « "\\" « buf;
#endif
      }
    }


// Implementation to perform binary read and write
// TODO rewrite operation so that either:
// #1. dummy implementation use a pointer to Internal and do ++p (faster)
// #2. Actually do some meta programming to unroll the loop
// (no notion of order in VM ...)
template<> class EncodingImplementation<VR::VRBINARY> {
public:
  template<typename T> // FIXME this should be VRToType<TVR>::Type
    static inline void ReadComputeLength(T* data, unsigned int &length,
      std::istream &_is) {
    const unsigned int type_size = sizeof(T);
    assert( data ); // Can we read from pointer ?
    //assert( length );
    length /= type_size;
    assert( _is ); // Is stream valid ?
    _is.read( reinterpret_cast<char*>(data+0), type_size);
    for(unsigned long i=1; i<length; ++i) {
      assert( _is );
      _is.read( reinterpret_cast<char*>(data+i), type_size );
    }
    }
  template<typename T>
  static inline void ReadNoSwap(T* data, unsigned long length,
    std::istream &_is) {
    const unsigned int type_size = sizeof(T);
    assert( data ); // Can we read from pointer ?
    assert( length );
    assert( _is ); // Is stream valid ?
    _is.read( reinterpret_cast<char*>(data+0), type_size);
    for(unsigned long i=1; i<length; ++i) {
      if( _is )
      _is.read( reinterpret_cast<char*>(data+i), type_size );
    }
    //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem(data,
    //  _is.GetSwapCode(), length);
    //SwapperNoOp::SwapArray(data,length);
  }
  template<typename T>
  static inline void Read(T* data, unsigned long length,
    std::istream &_is) {
    const unsigned int type_size = sizeof(T);
    assert( data ); // Can we read from pointer ?
    assert( length );
    assert( _is ); // Is stream valid ?
    _is.read( reinterpret_cast<char*>(data+0), type_size);
    for(unsigned long i=1; i<length; ++i) {
      if( _is )
      _is.read( reinterpret_cast<char*>(data+i), type_size );
    }
    //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem(data,
    //  _is.GetSwapCode(), length);
    SwapperNoOp::SwapArray(data,length);
  }
  template<typename T>
  static inline void Write(const T* data, unsigned long length,
    std::ostream &_os) {
    const unsigned int type_size = sizeof(T);
    assert( data ); // Can we write into pointer ?
    assert( length );
    assert( _os ); // Is stream valid ?
    //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem((T*)data,
    //  _os.GetSwapCode(), length);
    T swappedData = SwapperNoOp::Swap(data[0]);
    _os.write( reinterpret_cast<const char*>(&swappedData), type_size);
    for(unsigned long i=1; i<length;++i) {
      assert( _os );
```

```
      swappedData = SwapperNoOp::Swap(data[i]);
      _os.write( reinterpret_cast<const char*>(&swappedData), type_size );
    }
    //ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem((T*)data,
    //  _os.GetSwapCode(), length);
  }
};

// For particular case for ASCII string
// WARNING: This template explicitly instantiates a particular
// EncodingImplementation THEREFORE it is required to be declared after the
// EncodingImplementation is needs (doh!)
#if 0
template<int TVM>
class Element<TVM>
{
public:
  Element(const char array[])
    {
    unsigned int i = 0;
    const char sep = '\\';
    std::string sarray = array;
    std::string::size_type pos1 = 0;
    std::string::size_type pos2 = sarray.find(sep, pos1+1);
    while(pos2 != std::string::npos)
      {
      Internal[i++] = sarray.substr(pos1, pos2-pos1);
      pos1 = pos2+1;
      pos2 = sarray.find(sep, pos1+1);
      }
    Internal[i] = sarray.substr(pos1, pos2-pos1);
    // Shouldn't we do the contrary, since we know how many separators
    // (and default behavior is to discard anything after the VM declared
    assert( GetLength()-1 == i );
    }

  unsigned long GetLength() const {
    return VMToLength<TVM>::Length;
  }
  // Implementation of Print is common to all Mode (ASCII/Binary)
  void Print(std::ostream &_os) const {
    _os « Internal[0]; // VM is at least guarantee to be one
    for(int i=1; i<VMToLength<TVM>::Length; ++i)
      _os « "," « Internal[i];
    }

  void Read(std::istream &_is) {
    EncodingImplementation<VR::VRASCII>::Read(Internal, GetLength(),_is);
    }
  void Write(std::ostream &_os) const {
    EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(),_os);
    }
private:
  typename String Internal[VMToLength<TVM>::Length];
};

template< int TVM>
class Element<VR::PN, TVM> : public StringElement<TVM>
{
  enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<VR::PN, TVM> ) };
};
#endif

// Implementation for the undefined length (dynamically allocated array)
template<long long TVR>
class Element<TVR, VM::VM1_n>
{
  enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM1_n> ) };
public:
  // This the way to prevent default initialization
  explicit Element() { Internal=nullptr; Length=0; Save = false; }
  ~Element() {
    if( Save ) {
      delete[] Internal;
    }
    Internal = nullptr;
  }

  static VR  GetVR()  { return (VR::VRType)TVR; }
  static VM  GetVM()  { return VM::VM1_n; }

```

```
// Length manipulation
// SetLength should really be protected anyway...all operation
// should go through SetArray
unsigned long GetLength() const { return Length; }
typedef typename VRToType<TVR>::Type Type;

void SetLength(unsigned long len) {
  const unsigned int size = sizeof(Type);
  if( len ) {
    if( len > Length ) {
      // perform realloc
      assert( (len / size) * size == len );
      Type *internal = new Type[len / size];
      assert( Save == false );
      Save = true; // ????
      if( Internal )
        {
        memcpy(internal, Internal, len);
        delete[] Internal;
        }
      Internal = internal;
      }
    }
  Length = len / size;
}

// If save is set to zero user should not delete the pointer
//void SetArray(const typename VRToType<TVR>::Type *array, int len, bool save = false)
void SetArray(const Type *array, unsigned long len,
  bool save = false) {
  if( save ) {
    SetLength(len); // realloc
    memcpy(Internal, array, len/*/sizeof(Type)*/);
    assert( Save == false );
    }
  else {
    // TODO rewrite this stupid code:
    assert( Length == 0 );
    assert( Internal == nullptr );
    assert( Save == false );
    Length = len / sizeof(Type);
    //assert( (len / sizeof(Type)) * sizeof(Type) == len );
    // MR00010001.dcm is a tough kid: 0019,105a is supposed to be VR::FL, VM::VM3 but
    // length is 14 bytes instead of 12 bytes. Simply consider value is total garbage.
    if( (len / sizeof(Type)) * sizeof(Type) != len ) { Internal = nullptr; Length = 0; }
    else Internal = const_cast<Type*>(array);
    }
    Save = save;
}
void SetValue(typename VRToType<TVR>::Type v, unsigned int idx = 0) {
  assert( idx < Length );
  Internal[idx] = v;
}
const typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) const {
  assert( idx < Length );
  return Internal[idx];
}
typename VRToType<TVR>::Type &GetValue(unsigned int idx = 0) {
  //assert( idx < Length );
  return Internal[idx];
}
typename VRToType<TVR>::Type operator[] (unsigned int idx) const {
  return GetValue(idx);
}
void Set(Value const &v) {
  const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
  assert( bv ); // That would be bad...
  if( (VR::VRType)(VRToEncoding<TVR>::Mode) == VR::VRBINARY )
    {
    const Type* array = (const Type*)bv->GetVoidPointer();
    if( array ) {
      assert( array ); // That would be bad...
      assert( Internal == nullptr );
      SetArray(array, bv->GetLength() ); }
    }
  else
    {
    std::stringstream ss;
    std::string s = std::string( bv->GetPointer(), bv->GetLength() );
    ss.str( s );
    EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
```

```
        GetLength(),ss);
      }
  }
  void SetFromDataElement(DataElement const &de) {
    const ByteValue *bv = de.GetByteValue();
    if( !bv ) return;
#ifdef GDCM_WORDS_BIGENDIAN
    if( de.GetVR() == VR::UN /*|| de.GetVR() == VR::INVALID*/ )
#else
    if( de.GetVR() == VR::UN || de.GetVR() == VR::INVALID )
#endif
      {
      Set(de.GetValue());
      }
    else
      {
      SetNoSwap(de.GetValue());
      }
  }


  // Need to be placed after definition of EncodingImplementation<VR::VRASCII>
  void WriteASCII(std::ostream &os) const {
    return EncodingImplementation<VR::VRASCII>::Write(Internal, GetLength(), os);
    }

  // Implementation of Print is common to all Mode (ASCII/Binary)
  void Print(std::ostream &_os) const {
    assert( Length );
    assert( Internal );
    _os « Internal[0]; // VM is at least guarantee to be one
    const unsigned long length = GetLength() < 25 ? GetLength() : 25;
    for(unsigned long i=1; i<length; ++i)
      _os « "," « Internal[i];
    }
  void Read(std::istream &_is) {
    if( !Internal ) return;
    EncodingImplementation<VRToEncoding<TVR>::Mode>::Read(Internal,
      GetLength(),_is);
    }
  //void ReadComputeLength(std::istream &_is) {
  //  if( !Internal ) return;
  //  EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadComputeLength(Internal,
  //    Length,_is);
  //  }
  void Write(std::ostream &_os) const {
    EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
      GetLength(),_os);
    }

  DataElement GetAsDataElement() const {
    DataElement ret;
    ret.SetVR( (VR::VRType)TVR );
    assert( ret.GetVR() != VR::SQ );
    if( Internal )
      {
      std::ostringstream os;
      EncodingImplementation<VRToEncoding<TVR>::Mode>::Write(Internal,
        GetLength(),os);
      if( (VR::VRType)VRToEncoding<TVR>::Mode == VR::VRASCII )
        {
        if( GetVR() != VR::UI )
          {
          if( os.str().size() % 2 )
            {
            os « " ";
            }
          }
        }
      VL::Type osStrSize = (VL::Type)os.str().size();
      ret.SetByteValue( os.str().c_str(), osStrSize );
      }
    return ret;
  }

  Element(const Element&_val) {
    if( this != &_val) {
      *this = _val;
      }
    }

```

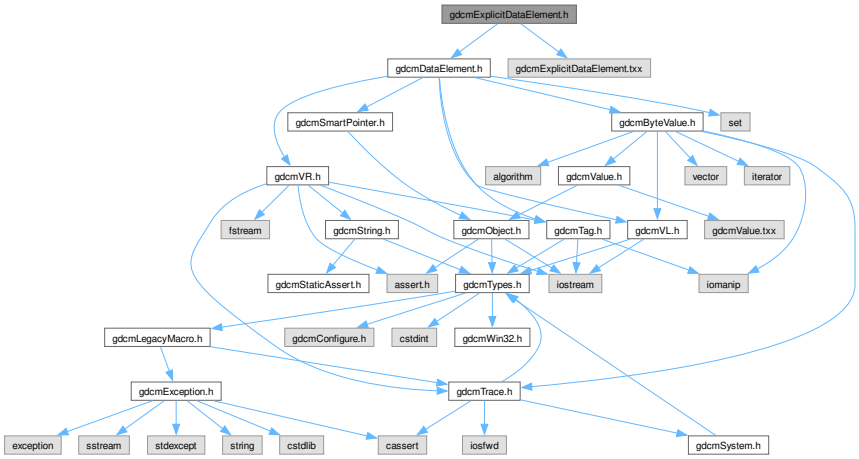
```
  Element &operator=(const Element &_val) {
    Length = 0; // SYITF
    Internal = 0;
    SetArray(_val.Internal, _val.Length, true);
    return *this;
    }
protected:
  void SetNoSwap(Value const &v) {
    const ByteValue *bv = dynamic_cast<const ByteValue*>(&v);
    assert( bv ); // That would be bad...
    if( (VR::VRType)(VRToEncoding<TVR>::Mode) == VR::VRBINARY )
      {
      const Type* array = (const Type*)bv->GetPointer();
      if( array ) {
        assert( array ); // That would be bad...
        assert( Internal == nullptr );
        SetArray(array, bv->GetLength() ); }
      }
    else
      {
      std::stringstream ss;
      std::string s = std::string( bv->GetPointer(), bv->GetLength() );
      ss.str( s );
      EncodingImplementation<VRToEncoding<TVR>::Mode>::ReadNoSwap(Internal,
        GetLength(),ss);
      }
  }

private:
  typename VRToType<TVR>::Type *Internal;
  unsigned long Length; // unsigned int ??
  bool Save;
};

//template <int TVM = VM::VM1_n>
//class Element<VR::OB, TVM > : public Element<VR::OB, VM::VM1_n> {};

// Partial specialization for derivatives of 1-n : 2-n, 3-n ...
template<long long TVR>
class Element<TVR, VM::VM1_2> : public Element<TVR, VM::VM1_n>
{
public:
  typedef Element<TVR, VM::VM1_n> Parent;
  void SetLength(int len) {
    if( len != 1 && len != 2 ) return;
    Parent::SetLength(len);
  }
};
template<long long TVR>
class Element<TVR, VM::VM2_n> : public Element<TVR, VM::VM1_n>
{
  enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM2_n> ) };
public:
  typedef Element<TVR, VM::VM1_n> Parent;
  void SetLength(int len) {
    if( len <= 1 ) return;
    Parent::SetLength(len);
  }
};
template<long long TVR>
class Element<TVR, VM::VM2_2n> : public Element<TVR, VM::VM2_n>
{
  enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM2_2n> ) };
public:
  typedef Element<TVR, VM::VM2_n> Parent;
  void SetLength(int len) {
    if( len % 2 ) return;
    Parent::SetLength(len);
  }
};
template<long long TVR>
class Element<TVR, VM::VM3_n> : public Element<TVR, VM::VM1_n>
{
  enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM3_n> ) };
public:
  typedef Element<TVR, VM::VM1_n> Parent;
  void SetLength(int len) {
    if( len <= 2 ) return;
    Parent::SetLength(len);
  }
};
```

```
template<long long TVR>
class Element<TVR, VM::VM3_3n> : public Element<TVR, VM::VM3_n>
{
  enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<TVR, VM::VM3_3n> ) };
public:
  typedef Element<TVR, VM::VM3_n> Parent;
  void SetLength(int len) {
    if( len % 3 ) return;
    Parent::SetLength(len);
  }
};
template<long long TVR>
class Element<TVR, VM::VM3_4> : public Element<TVR, VM::VM1_n>
{
public:
  typedef Element<TVR, VM::VM1_n> Parent;
  void SetLength(int len) {
    if( len != 3 && len != 4 ) return;
    Parent::SetLength(len);
  }
};


//template<int T> struct VRToLength;
//template <> struct VRToLength<VR::AS>
//{ enum { Length  = VM::VM1 }; }
//template<>
//class Element<VR::AS> : public Element<VR::AS, VRToLength<VR::AS>::Length >

// only 0010 1010 AS 1 Patient's Age
template<>
class Element<VR::AS, VM::VM5>
{
  enum { ElementDisableCombinationsCheck = sizeof ( ElementDisableCombinations<VR::AS, VM::VM5> ) };
public:
  char Internal[VMToLength<VM::VM5>::Length * sizeof( VRToType<VR::AS>::Type )];
  void Print(std::ostream &_os) const {
    _os « Internal;
    }
  unsigned long GetLength() const {
    return VMToLength<VM::VM5>::Length;
  }
};


template <>
class Element<VR::OB, VM::VM1> : public Element<VR::OB, VM::VM1_n> {};

// Same for OW:
template <>
class Element<VR::OW, VM::VM1> : public Element<VR::OW, VM::VM1_n> {};


} // namespace gdcm_ns

#endif //GDCMELEMENT_H
```

13.135 gdcmExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmExplicitDataElement.txx"
```

Include dependency graph for gdcmExplicitDataElement.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.

Namespaces

- namespace gdcm

13.136 gdcmExplicitDataElement.h

Go to the documentation of this file.

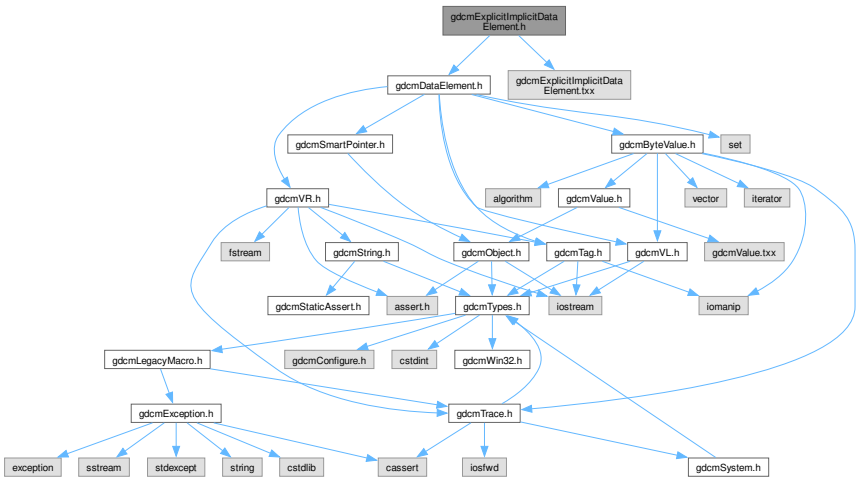
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMEXPLICITDATAELEMENT_H
00015 #define GDCMEXPLICITDATAELEMENT_H
00016
00017 #include "gdcmDataElement.h"
00018
00019 namespace gdcm_ns
00020 {
00025 class GDCM_EXPORT ExplicitDataElement : public DataElement
00026 {
00027 public:
00028   VL GetLength() const;
00029
00030   template <typename TSwap>
00031   std::istream &Read(std::istream &is);
00032
00033   template <typename TSwap>
00034   std::istream &ReadPreValue(std::istream &is);
00035
00036   template <typename TSwap>
00037   std::istream &ReadValue(std::istream &is, bool readvalues = true);
00038
00039   template <typename TSwap>
00040   std::istream &ReadWithLength(std::istream &is, VL & length);
00041
00042   template <typename TSwap>
00043   const std::ostream &Write(std::ostream &os) const;
00044 };
00045
00046 } // end namespace gdcm_ns
00047
00048 #include "gdcmExplicitDataElement.txx"
00049
00050 #endif //GDCMEXPLICITDATAELEMENT_H
```

13.137 gdcmExplicitImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmExplicitImplicitDataElement.txx"
```

Include dependency graph for gdcmExplicitImplicitDataElement.h:



Classes

- class gdcm::ExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.

Namespaces

- namespace gdcm

13.138 gdcmExplicitImplicitDataElement.h

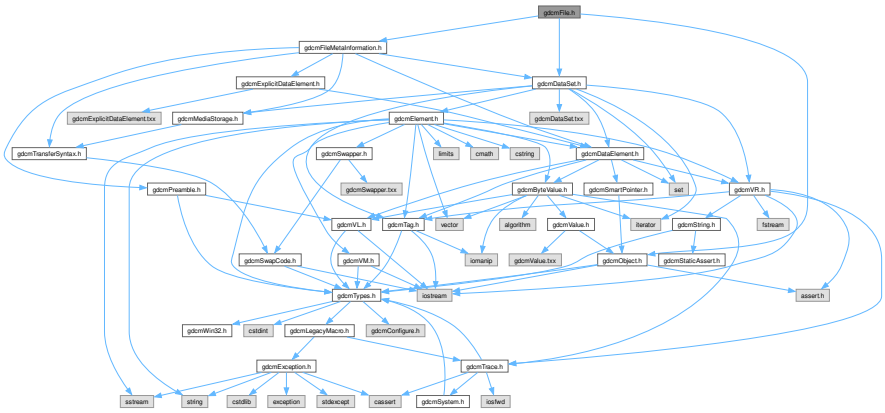
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMEXPLICITIMPLICITDATAELEMENT_H
#define GDCMEXPLICITIMPLICITDATAELEMENT_H

#include "gdcmDataElement.h"

namespace gdcm
{
// Data Element (ExplicitImplicit)
class GDCM_EXPORT ExplicitImplicitDataElement : public DataElement
{
public:
  VL GetLength() const;

  template <typename TSwap>
  std::istream &Read(std::istream &is);

  template <typename TSwap>
  std::istream &ReadPreValue(std::istream &is);

  template <typename TSwap>
  std::istream &ReadValue(std::istream &is, bool readvalues = true);

  template <typename TSwap>
  std::istream &ReadWithLength(std::istream &is, VL & length)
    {
    (void)length;
    return Read<TSwap>(is);
    }

  // PURPOSELY do not provide an implementation for writing !
  //template <typename TSwap>
  //const std::ostream &Write(std::ostream &os) const;
};

} // end namespace gdcm

#include "gdcmExplicitImplicitDataElement.txx"

#endif //GDCMEXPLICITIMPLICITDATAELEMENT_H
```

13.139 gdcmFile.h File Reference

```
#include "gdcmObject.h"
#include "gdcmDataSet.h"
#include "gdcmFileMetaInformation.h"
```

Include dependency graph for gdcmFile.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::File

  a DICOM File

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const File &val)

13.140 gdcmFile.h

Go to the documentation of this file.

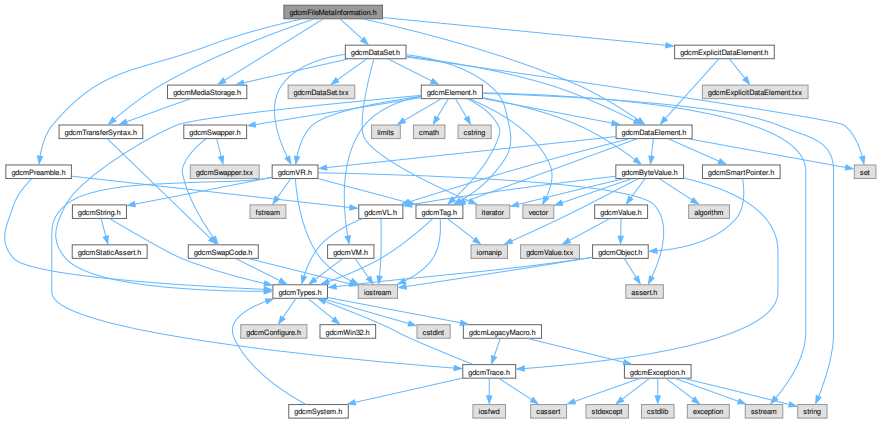
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILE_H
#define GDCMFILE_H

#include "gdcmObject.h"
#include "gdcmDataSet.h"
#include "gdcmFileMetaInformation.h"

namespace gdcm_ns
{

class GDCM_EXPORT File : public Object
{
public:
  File();
  ~File() override;

  friend std::ostream &operator«(std::ostream &os, const File &val);

  std::istream &Read(std::istream &is);

  std::ostream const &Write(std::ostream &os) const;

  const FileMetaInformation &GetHeader() const { return Header; }

  FileMetaInformation &GetHeader() { return Header; }

  void SetHeader( const FileMetaInformation &fmi ) { Header = fmi; }

  const DataSet &GetDataSet() const { return DS; }

  DataSet &GetDataSet() { return DS; }

  void SetDataSet( const DataSet &ds) { DS = ds; }

private:
  FileMetaInformation Header;
  DataSet DS;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const File &val)
{
  os « val.GetHeader() « std::endl;
  //os « val.GetDataSet() « std::endl; // FIXME
  assert(0);
  return os;
}

} // end namespace gdcm_ns

#endif //GDCMFILE_H
```

13.141 gdcmFileMetaInformation.h File Reference

```
#include "gdcmPreamble.h"
#include "gdcmDataSet.h"
```

```
#include "gdcmDataElement.h"
#include "gdcmMediaStorage.h"
#include "gdcmTransferSyntax.h"
#include "gdcmExplicitDataElement.h"
```

Include dependency graph for gdcmFileMetaInformation.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::FileMetaInformation

  Class to represent a File Meta Information.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileMetaInformation &val)

13.142 gdcmFileMetaInformation.h

Go to the documentation of this file.

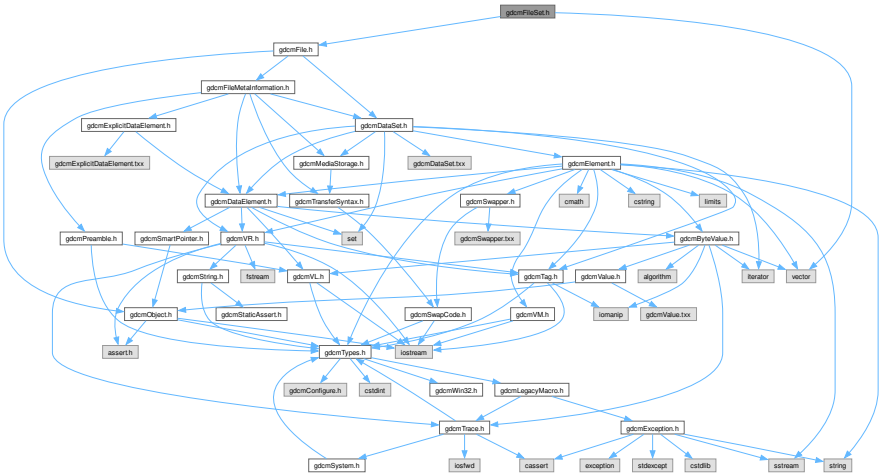
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMFILEMETAINFORMATION_H
00015 #define GDCMFILEMETAINFORMATION_H
00016
00017 #include "gdcmPreamble.h"
00018 #include "gdcmDataSet.h"
00019 #include "gdcmDataElement.h"
00020 #include "gdcmMediaStorage.h"
00021 #include "gdcmTransferSyntax.h"
00022 #include "gdcmExplicitDataElement.h"
00023
00024 namespace gdcm_ns
00025 {
00040 class GDCM_EXPORT FileMetaInformation : public DataSet
00041 {
00042 public:
00043   // FIXME: TransferSyntax::TS_END -> TransferSyntax::ImplicitDataElement
00044   FileMetaInformation();
00045   ~FileMetaInformation();
00046
00047   friend std::ostream &operator«(std::ostream &_os, const FileMetaInformation &_val);
00048
00049   bool IsValid() const { return true; }
00050
00051   TransferSyntax::NegociatedType GetMetaInformationTS() const { return MetaInformationTS; }
00052   void SetDataSetTransferSyntax(const TransferSyntax &ts);
00053   const TransferSyntax &GetDataSetTransferSyntax() const { return DataSetTS; }
00054   MediaStorage GetMediaStorage() const;
00055   std::string GetMediaStorageAsString() const;
00056
00057   // FIXME: no virtual function means: duplicate code...
00058   void Insert(const DataElement& de) {
00059     if( de.GetTag().GetGroup() == 0x0002 )
00060       {
00061       InsertDataElement( de );
00062       }
00063     else
00064       {
00065       gdcmErrorMacro( "Cannot add element with group != 0x0002 in the file meta header: " « de );
00066       }
00067   }
00068   void Replace(const DataElement& de) {
00069     Remove(de.GetTag());
00070     Insert(de);
00071   }
00072
00074   std::istream &Read(std::istream &is);
00075   std::istream &ReadCompat(std::istream &is);
00076
00078   std::ostream &Write(std::ostream &os) const;
00079
00081   void FillFromDataSet(DataSet const &ds);
00082
00084   const Preamble &GetPreamble() const { return P; }
00085   Preamble &GetPreamble() { return P; }
00086   void SetPreamble(const Preamble &p) { P = p; }
00087
00089   static void SetImplementationClassUID(const char * imp);
00090   static void AppendImplementationClassUID(const char * imp);
00091   static const char *GetImplementationClassUID();
00092   static void SetImplementationVersionName(const char * version);
00093   static const char *GetImplementationVersionName();
00094   static void SetSourceApplicationEntityTitle(const char * title);
```

```
  static const char *GetSourceApplicationEntityTitle();

  FileMetaInformation(FileMetaInformation const& fmi) = default;
  FileMetaInformation& operator=(const FileMetaInformation& fmi) = default;

    VL GetFullLength() const {
      return P.GetLength() + DataSet::GetLength<ExplicitDataElement>();
    }

protected:
  void ComputeDataSetTransferSyntax(); // FIXME

  template <typename TSwap>
  std::istream &ReadCompatInternal(std::istream &is);

  void Default();
  void ComputeDataSetMediaStorageSOPClass();

  TransferSyntax DataSetTS;
  TransferSyntax::NegociatedType MetaInformationTS;
  MediaStorage::MSType DataSetMS;

protected:
  static const char * GetFileMetaInformationVersion();
  static const char * GetGDCMImplementationClassUID();
  static const char * GetGDCMImplementationVersionName();
  static const char * GetGDCMSourceApplicationEntityTitle();

private:
  Preamble P;

//static stuff:
  static const char GDCM_FILE_META_INFORMATION_VERSION[];
  static const char GDCM_IMPLEMENTATION_CLASS_UID[];
  static const char GDCM_IMPLEMENTATION_VERSION_NAME[];
  static const char GDCM_SOURCE_APPLICATION_ENTITY_TITLE[];
  static std::string ImplementationClassUID;
  static std::string ImplementationVersionName;
  static std::string SourceApplicationEntityTitle;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const FileMetaInformation &val)
{
  os « val.GetPreamble() « std::endl;
  val.Print( os );
  return os;
}

} // end namespace gdcm_ns

#endif //GDCMFILEMETAINFORMATION_H
```

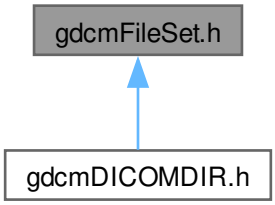
13.143 gdcmFileSet.h File Reference

```
#include "gdcmFile.h"
#include <vector>
```

Include dependency graph for gdcmFileSet.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::FileSet

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)

13.144 gdcmFileSet.h

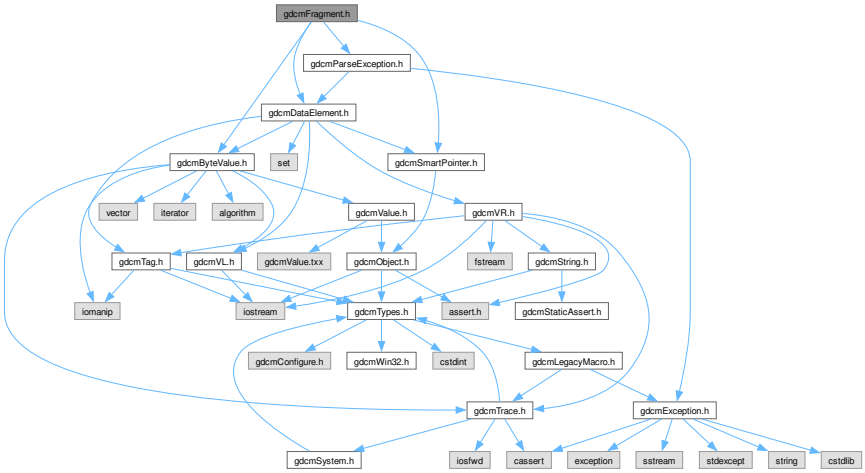
Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMFILESET_H
00015 #define GDCMFILESET_H
00016
00017 #include "gdcmFile.h"
00018 #include <vector>
00019
00020 namespace gdcm
00021 {
00026 class GDCM_EXPORT FileSet
00027 {
00028   friend std::ostream& operator«(std::ostream &_os, const FileSet &d);
00029 public:
00030   FileSet():Files() {}
00031   typedef std::string FileType;
00032   typedef std::vector<FileType> FilesType;
00033
00035   void AddFile(File const & ) {}
00036
00039   bool AddFile(const char *filename);
00040
00041   void SetFiles(FilesType const &files);
00042   FilesType const &GetFiles() const {
00043     return Files;
00044   }
00045
00046 private:
00047   FilesType Files;
00048 };
00049 //-----------------------------------------------------------------------------
00050 inline std::ostream& operator«(std::ostream &os, const FileSet &f)
00051 {
00052   (void)f; // FIXME
00053   return os;
00054 }
00055
00056 } // end namespace gdcm
00057
00058 #endif //GDCMFILESET_H
```

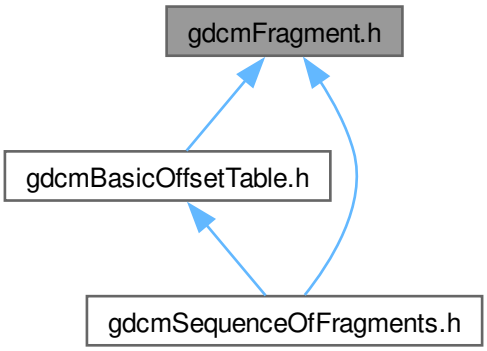
13.145 gdcmFragment.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include "gdcmParseException.h"
```

Include dependency graph for gdcmFragment.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Fragment

  Class to represent a Fragment.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Fragment &val)

13.146 gdcmFragment.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFRAGMENT_H
#define GDCMFRAGMENT_H

#include "gdcmDataElement.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
#include "gdcmParseException.h"

namespace gdcm_ns
{

// Implementation detail:
// I think Fragment should be a protected subclass of DataElement:
// looking somewhat like this:
/*
class GDCM_EXPORT Fragment : protected DataElement
{
public:
  using DataElement::GetTag;
  using DataElement::GetVL;
  using DataElement::SetByteValue;
  using DataElement::GetByteValue;
  using DataElement::GetValue;
*/
// Instead I am only hiding the SetTag member...

class GDCM_EXPORT Fragment : public DataElement
{
//protected:
//  void SetTag(const Tag &t);
public:
  Fragment() : DataElement(Tag(0xfffe, 0xe000), 0) {}
  friend std::ostream &operator«(std::ostream &os, const Fragment &val);

  VL GetLength() const;

  VL ComputeLength() const;

  template <typename TSwap>
  std::istream &Read(std::istream &is)
    {
    ReadPreValue<TSwap>(is);
    return ReadValue<TSwap>(is);
    }

  template <typename TSwap>
  std::istream &ReadPreValue(std::istream &is)
    {
    TagField.Read<TSwap>(is);
    if( !is )
      {
      //  BogusItemStartItemEnd.dcm
      throw Exception( "Problem #1" );
```

```
      }
    if( !ValueLengthField.Read<TSwap>(is) )
      {
      // GENESIS_SIGNA-JPEG-CorruptFrag.dcm
      // JPEG fragment is declared to have 61902, but in fact really is only 61901
      // so we end up reading 0xddff,0x00e0, and VL = 0x0 (1 byte)
      throw Exception( "Problem #2" );
      }
#ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
    const Tag itemStart(0xfffe, 0xe000);
    const Tag seqDelItem(0xfffe,0xe0dd);
    if( TagField != itemStart && TagField != seqDelItem )
      {
      throw Exception( "Problem #3" );
      }
#endif
    return is;
    }

  template <typename TSwap>
  std::istream &ReadValue(std::istream &is)
    {
    // Self
    SmartPointer<ByteValue> bv = new ByteValue;
    const std::streampos cur = is.tellg();
    if( cur != std::streampos(-1) )
      {
      is.seekg(0, std::ios::end);
      const std::streampos end = is.tellg();
      is.seekg(cur);
      if( end != std::streampos(-1) && is.good()
        && static_cast<uint64_t>(end - cur) < static_cast<uint32_t>(ValueLengthField) )
        {
        throw Exception( "Fragment Value Length exceeds remaining stream size" );
        }
      }
    bv->SetLength(ValueLengthField);
    if( !bv->Read<TSwap>(is) )
      {
      // Fragment is incomplete, but is a itemStart, let's try to push it anyway...
      gdcmWarningMacro( "Fragment could not be read" );
      //bv->SetLength(is.gcount());
      ValueField = bv;
      ParseException pe;
      pe.SetLastElement( *this );
      throw pe;
      }
    ValueField = bv;
    return is;
    }

  template <typename TSwap>
  std::istream &ReadBacktrack(std::istream &is)
    {
    const Tag itemStart(0xfffe, 0xe000);
    const Tag seqDelItem(0xfffe,0xe0dd);

    bool cont = true;
    const std::streampos start = is.tellg();
    const int max = 10;
    int offset = 0;
    while( cont )
      {
      TagField.Read<TSwap>(is);
      assert( is );
      if( TagField != itemStart && TagField != seqDelItem )
        {
        ++offset;
        is.seekg( (std::streampos)((size_t)start - offset) );
        gdcmWarningMacro( "Fuzzy Search, backtrack: " « (start - is.tellg()) « " Offset: " « is.tellg() );
        if( offset > max )
          {
          gdcmErrorMacro( "Giving up" );
          throw "Impossible to backtrack";
          }
        }
      else
        {
        cont = false;
        }
      }
```

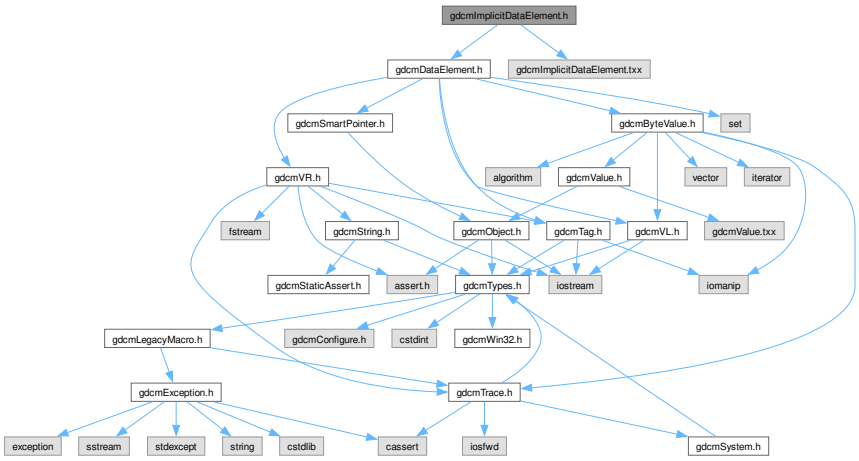
```
    assert( TagField == itemStart || TagField == seqDelItem );
    if( !ValueLengthField.Read<TSwap>(is) )
      {
      return is;
      }

    // Self
    SmartPointer<ByteValue> bv = new ByteValue;
    const std::streampos cur2 = is.tellg();
    if( cur2 != std::streampos(-1) )
      {
      is.seekg(0, std::ios::end);
      const std::streampos end2 = is.tellg();
      is.seekg(cur2);
      if( end2 != std::streampos(-1) && is.good()
        && static_cast<uint64_t>(end2 - cur2) < static_cast<uint32_t>(ValueLengthField) )
        {
        throw Exception( "Fragment Value Length exceeds remaining stream size" );
        }
      }
    bv->SetLength(ValueLengthField);
    if( !bv->Read<TSwap>(is) )
      {
      // Fragment is incomplete, but is a itemStart, let's try to push it anyway...
      gdcmWarningMacro( "Fragment could not be read" );
      //bv->SetLength(is.gcount());
      ValueField = bv;
      ParseException pe;
      pe.SetLastElement( *this );
      throw pe;
      }
    ValueField = bv;
    return is;
    }


  template <typename TSwap>
  std::ostream &Write(std::ostream &os) const {
    const Tag itemStart(0xfffe, 0xe000);
    const Tag seqDelItem(0xfffe,0xe0dd);
    if( !TagField.Write<TSwap>(os) )
      {
      assert(0 && "Should not happen");
      return os;
      }
    assert( TagField == itemStart
         || TagField == seqDelItem );
    const ByteValue *bv = GetByteValue();
    // VL
    // The following piece of code is hard to read in order to support such broken file as:
    // CompressedLossy.dcm
    if( IsEmpty() )
      {
      //assert( bv );
      VL zero = 0;
      if( !zero.Write<TSwap>(os) )
        {
        assert(0 && "Should not happen");
        return os;
        }
      }
    else
      {
      assert( ValueLengthField );
      assert( !ValueLengthField.IsUndefined() );
      const VL actuallen = bv->ComputeLength();
      assert( actuallen == ValueLengthField || actuallen == ValueLengthField + 1 );
      if( !actuallen.Write<TSwap>(os) )
        {
        assert(0 && "Should not happen");
        return os;
        }
      }
    // Value
    if( ValueLengthField && bv )
      {
      // Self
      assert( bv );
      assert( bv->GetLength() == ValueLengthField );
      if( !bv->Write<TSwap>(os) )
        {
```

```
        assert(0 && "Should not happen");
        return os;
        }
      }
    return os;
    }
};
//-----------------------------------------------------------------------------
inline std::ostream &operator«(std::ostream &os, const Fragment &val)
{
  os « "Tag: " « val.TagField;
  os « "\tVL: " « val.ValueLengthField;
  if( val.ValueField )
    {
    os « "\t" « *(val.ValueField);
    }

  return os;
}

} // end namespace gdcm_ns

#endif //GDCMFRAGMENT_H
```

13.147 gdcmImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmImplicitDataElement.txx"
```

Include dependency graph for gdcmImplicitDataElement.h:



Classes

- class gdcm::ImplicitDataElement

  Class to represent an Implicit VR Data Element.

Namespaces

- namespace gdcm

13.148 gdcmImplicitDataElement.h

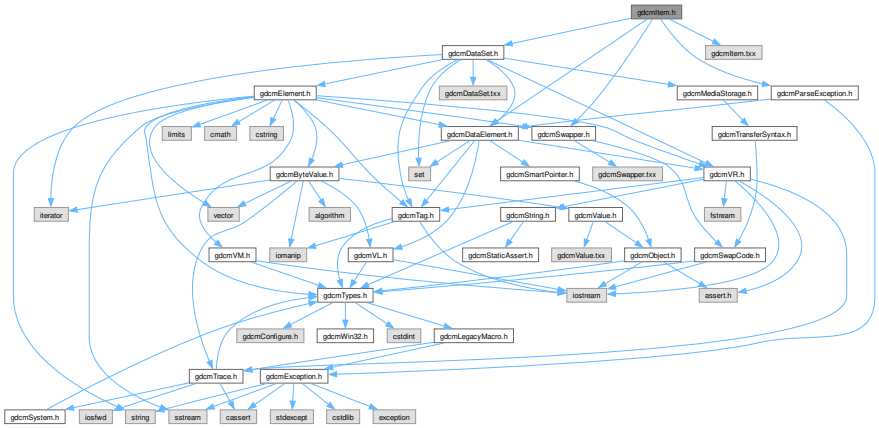
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMPLICITDATAELEMENT_H
#define GDCMIMPLICITDATAELEMENT_H

#include "gdcmDataElement.h"

namespace gdcm_ns
{

class GDCM_EXPORT ImplicitDataElement : public DataElement
{
public:
  VL GetLength() const;

  template <typename TSwap>
  std::istream &Read(std::istream& is);

  template <typename TSwap>
  std::istream &ReadPreValue(std::istream& is);

  template <typename TSwap>
  std::istream &ReadValue(std::istream& is, bool readvalues = true);

  template <typename TSwap>
  std::istream &ReadWithLength(std::istream& is, VL & length, bool readvalues = true);

  template <typename TSwap>
  std::istream &ReadValueWithLength(std::istream& is, VL & length, bool readvalues = true);

  template <typename TSwap>
  const std::ostream &Write(std::ostream& os) const;
};

} // end namespace gdcm_ns

#include "gdcmImplicitDataElement.txx"

#endif //GDCMIMPLICITDATAELEMENT_H
```

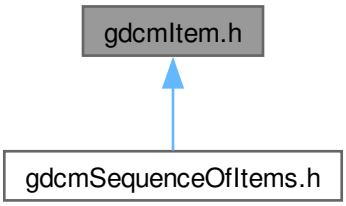
13.149 gdcmItem.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"
#include "gdcmItem.txx"
```

Include dependency graph for gdcmItem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Item

  Class to represent an Item.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)

13.150 gdcmItem.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMITEM_H
#define GDCMITEM_H

#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"

#ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
#include "gdcmByteSwapFilter.h"
#endif

namespace gdcm_ns
{

class DataSet;
class GDCM_EXPORT Item : public DataElement
{
public:
  Item() : DataElement(Tag(0xfffe, 0xe000), 0xFFFFFFFF), NestedDataSet() {}
  friend std::ostream& operator« (std::ostream &os, const Item &val);

  void Clear() {
    this->DataElement::Clear();
    NestedDataSet.Clear();
    }

  template <typename TDE>
  VL GetLength() const;

  void InsertDataElement(const DataElement & de) {
    NestedDataSet.Insert(de);
    // Update the length
    if( !IsUndefinedLength() )
      {
      assert( 0 && "InsertDataElement" );
      //ValueLengthField += de.GetLength();
      }
    }
  const DataElement& GetDataElement(const Tag& t) const
    {
    return NestedDataSet.GetDataElement(t);
    }

  // Completely defines it with the nested dataset
  // destroy anything present
  void SetNestedDataSet(const DataSet& nested)
    {
    NestedDataSet = nested;
    }
  // Return a const ref to the Nested Data Set
  const DataSet &GetNestedDataSet() const
    {
    return NestedDataSet;
    }
  DataSet &GetNestedDataSet()
    {
    return NestedDataSet;
    }

  //Value const & GetValue() const { return *NestedDataSet; }
```

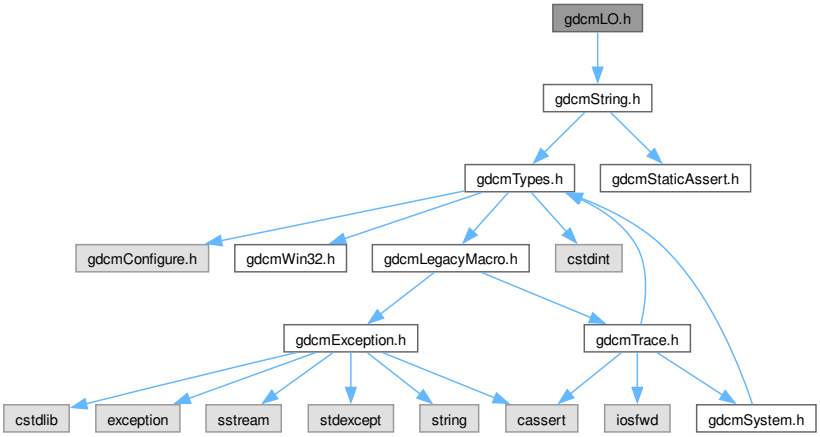
```
  Item(Item const &val):DataElement(val)
    {
    NestedDataSet = val.NestedDataSet;
    }

  template <typename TDE, typename TSwap>
  std::istream &Read(std::istream &is) {
    // Superclass
    {
        DataSet &nested = NestedDataSet;
        nested.Clear();
        assert( nested.IsEmpty() );
    }
    if( !TagField.Read<TSwap>(is) )
      {
      throw Exception("Should not happen (item)");
      return is;
      }
#ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
    // MR_Philips_Intera_SwitchIndianess_noLgtSQItem_in_trueLgtSeq.dcm
    if( TagField == Tag(0xfeff, 0x00e0)
      || TagField == Tag(0xfeff, 0xdde0) )
      {
      gdcmWarningMacro( "ByteSwaping Private SQ: " « TagField );
      // Invert previously read TagField since wrong endianness:
      TagField = Tag( SwapperDoOp::Swap( TagField.GetGroup() ), SwapperDoOp::Swap( TagField.GetElement() ) );
      assert ( TagField == Tag(0xfffe, 0xe000)
        || TagField == Tag(0xfffe, 0xe0dd) );

      if( !ValueLengthField.Read<SwapperDoOp>(is) )
        {
        assert(0 && "Should not happen");
        return is;
        }
      // Self
      // Some file written by GDCM 1.0 we write 0xFFFFFFFF instead of 0x0
      if( TagField == Tag(0xfffe,0xe0dd) )
        {
        if( ValueLengthField )
          {
          gdcmErrorMacro( "ValueLengthField is not 0" );
          }
        }
      //else if( ValueLengthField == 0 )
      //  {
      //  //assert( TagField == Tag( 0xfffe, 0xe0dd) );
      //  if( TagField != Tag( 0xfffe, 0xe0dd) )
      //    {
      //    gdcmErrorMacro( "SQ: " « TagField « " has a length of 0" );
      //    }
      //  }
      else if( ValueLengthField.IsUndefined() )
        {
        DataSet &nested = NestedDataSet;
        nested.Clear();
        assert( nested.IsEmpty() );
        std::streampos start = is.tellg();
        try
          {
          nested.template ReadNested<TDE,SwapperDoOp>(is);
          ByteSwapFilter bsf(nested);
          bsf.ByteSwap();
          }
        catch(ParseException &pe)
          {
          (void)pe;
          // MR_Philips_Intera_PrivateSequenceExplicitVR_in_SQ_2001_e05f_item_wrong_lgt_use_NOSHADOWSEQ.dcm
          // You have to byteswap the length but not the tag...sigh
          gdcmWarningMacro( "Attempt to read nested Item without byteswapping the Value Length." );
          start -= is.tellg();
          assert( start < 0 );
          is.seekg( start, std::ios::cur );
          nested.Clear();
          nested.template ReadNested<TDE,SwapperNoOp>(is);
          ByteSwapFilter bsf(nested);
          // Tag are read in big endian, need to byteswap them back...
          bsf.SetByteSwapTag(true);
          bsf.ByteSwap();
```

```
          }
        catch(Exception &e)
          {
          // MR_Philips_Intera_No_PrivateSequenceImplicitVR.dcm
          throw e;
          }
        catch(...)
          {
          assert(0);
          }
        }
      else /* if( ValueLengthField.IsUndefined() ) */
        {
        DataSet &nested = NestedDataSet;
        nested.Clear();
        assert( nested.IsEmpty() );
        nested.template ReadWithLength<TDE,SwapperDoOp>(is, ValueLengthField);
        ByteSwapFilter bsf(nested);
        bsf.ByteSwap();
        }
      return is;
      }
    // http://groups.google.com/group/comp.protocols.dicom/msg/c07efcf5e759fc83
    // Bug_Philips_ItemTag_3F3F.dcm
    if( TagField == Tag(0x3f3f, 0x3f00) )
      {
      //TagField = Tag(0xfffe, 0xe000);
      }
#endif
    if( TagField != Tag(0xfffe, 0xe000) && TagField != Tag(0xfffe, 0xe0dd) )
      {
      gdcmDebugMacro( "Invalid Item, found tag: " « TagField);
      throw Exception( "Not a valid Item" );
      }
    assert( TagField == Tag(0xfffe, 0xe000) || TagField == Tag(0xfffe, 0xe0dd) );

    if( !ValueLengthField.Read<TSwap>(is) )
      {
      assert(0 && "Should not happen");
      return is;
      }
    // Self
    if( TagField == Tag(0xfffe,0xe0dd) )
      {
      // Some file written by GDCM 1.0 were written with 0xFFFFFFFF instead of 0x0
      if( ValueLengthField )
        {
        gdcmDebugMacro( "ValueLengthField is not 0 but " « ValueLengthField );
        }
      }
    else if( ValueLengthField.IsUndefined() )
      {
      DataSet &nested = NestedDataSet;
      nested.Clear();
      assert( nested.IsEmpty() );
      nested.template ReadNested<TDE,TSwap>(is);
      }
    else /* if( ValueLengthField.IsUndefined() ) */
      {
      assert( !ValueLengthField.IsUndefined() );
      DataSet &nested = NestedDataSet;
      nested.Clear();
      assert( nested.IsEmpty() );
      nested.template ReadWithLength<TDE,TSwap>(is, ValueLengthField);
      }

    return is;
  }

  template <typename TDE, typename TSwap>
  const std::ostream &Write(std::ostream &os) const {
#ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
    if( TagField == Tag(0x3f3f,0x3f00) && false )
      {
      Tag t(0xfffe, 0xe000);
      t.Write<TSwap>(os);
      }
    else
#endif
      {
      assert ( TagField == Tag(0xfffe, 0xe000)
```

```
        || TagField == Tag(0xfffe, 0xe0dd) );
      // Not sure how this happen
      if( TagField == Tag(0xfffe, 0xe0dd) )
        {
        gdcmWarningMacro( "SeqDelItem found in defined length Sequence" );
        assert( ValueLengthField == 0 );
        assert( NestedDataSet.Size() == 0 );
        }
      if( !TagField.Write<TSwap>(os) )
        {
        assert(0 && "Should not happen");
        return os;
        }
      }
    if( ValueLengthField.IsUndefined() )
      {
      if( !ValueLengthField.Write<TSwap>(os) )
        {
        assert(0 && "Should not happen");
        return os;
        }
      }
    else
      {
      const VL dummy = NestedDataSet.GetLength<TDE>();
      assert( dummy % 2 == 0 );
      //assert( ValueLengthField == dummy );
      if( !dummy.Write<TSwap>(os) )
        {
        assert(0 && "Should not happen");
        return os;
        }
      }
    // Self
    NestedDataSet.Write<TDE,TSwap>(os);
    if( ValueLengthField.IsUndefined() )
      {
      const Tag itemDelItem(0xfffe,0xe00d);
      itemDelItem.Write<TSwap>(os);
      VL zero = 0;
      zero.Write<TSwap>(os);
      }

    return os;
  }

/*
There are three special SQ related Data Elements that are not ruled by the VR encoding rules conveyed
by the Transfer Syntax. They shall be encoded as Implicit VR. These special Data Elements are Item
(FFFE,E000), Item Delimitation Item (FFFE,E00D), and Sequence Delimitation Item (FFFE,E0DD).
However, the Data Set within the Value Field of the Data Element Item (FFFE,E000) shall be encoded
according to the rules conveyed by the Transfer Syntax.
*/
  bool FindDataElement(const Tag &t) const {
    return NestedDataSet.FindDataElement( t );
  }

private:
  /* NESTED DATA SET  a Data Set contained within a Data Element of an other Data Set.
   * May be nested recursively.
   * Only Data Elements with VR = SQ  may, themselves, contain Data Sets
   */
  DataSet NestedDataSet;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& os, const Item &val)
{
  os « val.TagField;
  os « "\t" « val.ValueLengthField « "\n";
  val.NestedDataSet.Print( os, "\t" );

  return os;
}


} // end namespace gdcm_ns

#include "gdcmItem.txx"

#endif //GDCMITEM_H
```

13.151 gdcmLO.h File Reference

`#include "gdcmString.h"`
Include dependency graph for gdcmLO.h:



Classes

- class gdcm::LO
  LO.

Namespaces

- namespace gdcm

13.152 gdcmLO.h

Go to the documentation of this file.

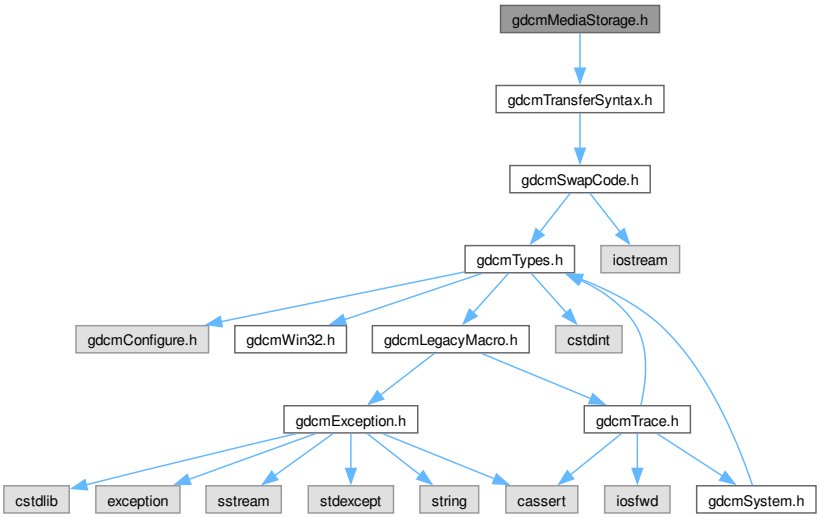
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMLO_H
#define GDCMLO_H

#include "gdcmString.h"
```

```
namespace gdcm
{

class /*GDCM_EXPORT*/ LO : public String<'\\',64> /* PLEASE do not export me */
{
public:
  // typedef are not inherited:
  typedef String<'\\',64> Superclass;
  typedef Superclass::value_type             value_type;
  typedef Superclass::pointer                pointer;
  typedef Superclass::reference              reference;
  typedef Superclass::const_reference        const_reference;
  typedef Superclass::size_type              size_type;
  typedef Superclass::difference_type        difference_type;
  typedef Superclass::iterator               iterator;
  typedef Superclass::const_iterator         const_iterator;
  typedef Superclass::reverse_iterator       reverse_iterator;
  typedef Superclass::const_reverse_iterator const_reverse_iterator;

  // LO constructors.
  LO(): Superclass() {}
  LO(const value_type* s): Superclass(s) {}
  LO(const value_type* s, size_type n): Superclass(s, n) {}
  LO(const Superclass& s, size_type pos=0, size_type n=npos):
    Superclass(s, pos, n) {}

  bool IsValid() const {
    if( !Superclass::IsValid() ) return false;
    // Implementation specific:
    return true;
  }
};

} // end namespace gdcm

#endif //GDCMLO_H
```

13.153 gdcmMediaStorage.h File Reference

`#include "gdcmTransferSyntax.h"`

Include dependency graph for gdcmMediaStorage.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::MediaStorage

  MediaStorage.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)

13.154 gdcmMediaStorage.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMMEDIASTORAGE_H
00015 #define GDCMMEDIASTORAGE_H
00016
00017 #include "gdcmTransferSyntax.h"
00018
00019 namespace gdcm { class Tag; }
00020 namespace gdcm_ns
00021 {
00022 #if !defined(SWIGPYTHON) && !defined(SWIGCSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00023 using namespace gdcm;
00024 #endif
00025 class DataSet;
00026 class FileMetaInformation;
00027 class File;
00028
00029 // WARNING: This class will be deprecated in the future. There is no reason to extend this class.
00030 // Please check the gdcm::UIDs class if adding new well known UID.
00031
00043 class GDCM_EXPORT MediaStorage
00044 {
00045 public:
00046   typedef enum {
00047     MediaStorageDirectoryStorage = 0,
00048     ComputedRadiographyImageStorage,
00049     DigitalXRayImageStorageForPresentation,
```

```
    DigitalXRayImageStorageForProcessing,
    DigitalMammographyImageStorageForPresentation,
    DigitalMammographyImageStorageForProcessing,
    DigitalIntraoralXrayImageStorageForPresentation,
    DigitalIntraoralXRayImageStorageForProcessing,
    CTImageStorage,
    EnhancedCTImageStorage,
    UltrasoundImageStorageRetired,
    UltrasoundImageStorage,
    UltrasoundMultiFrameImageStorageRetired,
    UltrasoundMultiFrameImageStorage,
    MRImageStorage,
    EnhancedMRImageStorage,
    MRSpectroscopyStorage,
    NuclearMedicineImageStorageRetired,
    SecondaryCaptureImageStorage,
    MultiframeSingleBitSecondaryCaptureImageStorage,
    MultiframeGrayscaleByteSecondaryCaptureImageStorage,
    MultiframeGrayscaleWordSecondaryCaptureImageStorage,
    MultiframeTrueColorSecondaryCaptureImageStorage,
    StandaloneOverlayStorage,
    StandaloneCurveStorage,
    LeadECGWaveformStorage, // 12-
    GeneralECGWaveformStorage,
    AmbulatoryECGWaveformStorage,
    HemodynamicWaveformStorage,
    CardiacElectrophysiologyWaveformStorage,
    BasicVoiceAudioWaveformStorage,
    StandaloneModalityLUTStorage,
    StandaloneVOILUTStorage,
    GrayscaleSoftcopyPresentationStateStorageSOPClass,
    XRayAngiographicImageStorage,
    XRayRadiofluoroscopingImageStorage,
    XRayAngiographicBiPlaneImageStorageRetired,
    NuclearMedicineImageStorage,
    RawDataStorage,
    SpacialRegistrationStorage, // Spatial
    SpacialFiducialsStorage, // Spatial..
    PETImageStorage,
    RTImageStorage,
    RTDoseStorage,
    RTStructureSetStorage,
    RTPlanStorage,
    CSANonImageStorage,
    Philips3D,
    EnhancedSR,
    BasicTextSR,
    HardcopyGrayscaleImageStorage,
    ComprehensiveSR,
    DetachedStudyManagementSOPClass,
    EncapsulatedPDFStorage,
    EncapsulatedCDAStorage,
    StudyComponentManagementSOPClass,
    DetachedVisitManagementSOPClass,
    DetachedPatientManagementSOPClass,
    VideoEndoscopicImageStorage,
    GeneralElectricMagneticResonanceImageStorage,
    GEPrivate3DModelStorage,
    ToshibaPrivateDataStorage,
    MammographyCADSR,
    KeyObjectSelectionDocument,
    HangingProtocolStorage,
    ModalityPerformedProcedureStepSOPClass,
    PhilipsPrivateMRSyntheticImageStorage,
    VLPhotographicImageStorage,
    SegmentationStorage, // "1.2.840.10008.5.1.4.1.1.66.4"
    RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
    XRay3DAngiographicImageStorage, // 1.2.840.10008.5.1.4.1.1.13.1.1
    EnhancedXAImageStorage,
    RTIonBeamsTreatmentRecordStorage, // 1.2.840.10008.5.1.4.1.1.481.9
    SurfaceSegmentationStorage, // "1.2.840.10008.5.1.4.1.1.66.5"
    VLWholeSlideMicroscopyImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.6
    RTTreatmentSummaryRecordStorage, // 1.2.840.10008.5.1.4.1.1.481.7
    EnhancedUSVolumeStorage, // 1.2.840.10008.5.1.4.1.1.6.2
    XRayRadiationDoseSR, // 1.2.840.10008.5.1.4.1.1.88.67
    VLEndoscopicImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.1
    BreastTomosynthesisImageStorage, // 1.2.840.10008.5.1.4.1.1.13.1.3
    FujiPrivateCRImageStorage, // 1.2.392.200036.9125.1.1.2
    OphthalmicPhotography8BitImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.5.1
    OphthalmicTomographyImageStorage, // 1.2.840.10008.5.1.4.1.1.77.1.5.4
    VLMicroscopicImageStorage,
```

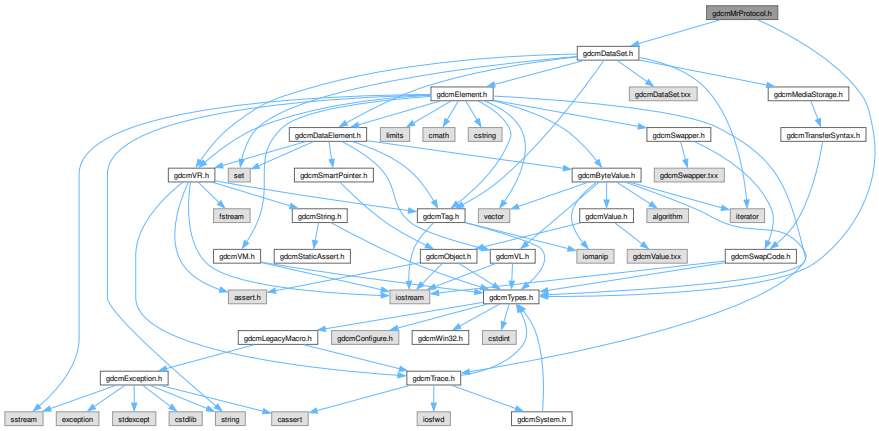
```
    EnhancedPETImageStorage,
    VideoPhotographicImageStorage,
    XRay3DCraniofacialImageStorage,
    IVOCTForPresentation,
    IVOCTForProcessing,
    LegacyConvertedEnhancedCTImageStorage,
    LegacyConvertedEnhancedMRImageStorage,
    LegacyConvertedEnhancedPETImageStorage,
    BreastProjectionXRayImageStorageForPresentation,
    BreastProjectionXRayImageStorageForProcessing,
    HardcopyColorImageStorage,
    EnhancedMRColorImageStorage,
    FujiPrivateMammoCRImageStorage,
    OphthalmicPhotography16BitImageStorage,
    VideoMicroscopicImageStorage,
    MS_END
  } MSType; // Media Storage Type

typedef enum {
    NoObject = 0, // DICOMDIR
    Video, // Most common, include image, video and volume
    Waveform, // Isn't it simply a 1D video ?
    Audio, // ???
    PDF,
    URI, // URL...
    Segmentation, // TODO
    ObjectEnd
  } ObjectType;

  static const char* GetMSString(MSType ts);

  const char* GetString() const;
  static MSType GetMSType(const char *str);

  MediaStorage(MSType type = MS_END):MSField(type) {}

  static bool IsImage(MSType ts);

  operator MSType () const { return MSField; }

  const char *GetModality() const;
  unsigned int GetModalityDimension() const;

  static unsigned int GetNumberOfMSType();
  static unsigned int GetNumberOfMSString();
  static unsigned int GetNumberOfModality();


  bool SetFromFile(File const &file);

  bool SetFromDataSet(DataSet const &ds); // Will get the SOP Class UID
  bool SetFromHeader(FileMetaInformation const &fmi); // Will get the Media Storage SOP Class UID
  bool SetFromModality(DataSet const &ds);
  void GuessFromModality(const char *modality, unsigned int dimension = 2);

  friend std::ostream &operator«(std::ostream &os, const MediaStorage &ms);

  bool IsUndefined() const { return MSField == MS_END; }

protected:
  void SetFromSourceImageSequence(DataSet const &ds);

private:
  bool SetFromDataSetOrHeader(DataSet const &ds, const Tag & tag);

  std::string GetFromDataSetOrHeader(DataSet const &ds, const Tag & tag);
  std::string GetFromHeader(FileMetaInformation const &fmi);
  std::string GetFromDataSet(DataSet const &ds);

private:
  MSType MSField;
};
//-----------------------------------------------------------------------------
inline std::ostream &operator«(std::ostream &_os, const MediaStorage &ms)
{
  const char *msstring = MediaStorage::GetMSString(ms);
  _os « (msstring ? msstring : "INVALID MEDIA STORAGE");
  return _os;

}

```

```
} // end namespace gdcm_ns

#endif // GDCMMEDIASTORAGE_H
```

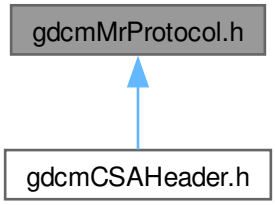
13.155 gdcmMrProtocol.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmMrProtocol.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::MrProtocol

  Class for MrProtocol.
- struct gdcm::MrProtocol::Slice
- struct gdcm::MrProtocol::SliceArray
- struct gdcm::MrProtocol::Vector3

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const MrProtocol &d)

13.156 gdcmMrProtocol.h

Go to the documentation of this file.

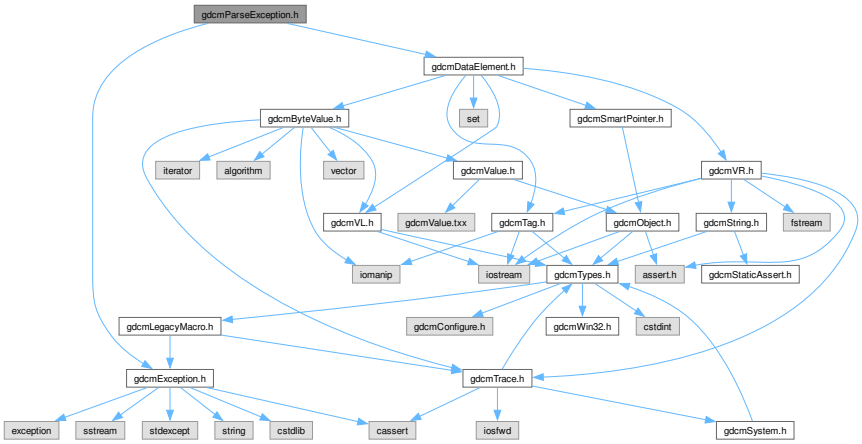
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMRPROTOCOL_H
#define GDCMMRPROTOCOL_H

#include "gdcmTypes.h"
#include "gdcmDataSet.h"

namespace gdcm
{
class ByteValue;
/*
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 */
//-----------------------------------------------------------------------------

class DataElement;
class GDCM_EXPORT MrProtocol
{
  friend std::ostream& operator«(std::ostream &_os, const MrProtocol &d);
public :
  MrProtocol();
  ~MrProtocol();

  bool Load( const ByteValue * bv, const char * str, int version );
  void Print(std::ostream &os) const;

  int GetVersion() const;

  const char * GetMrProtocolByName(const char *name) const;

  bool FindMrProtocolByName(const char *name) const;

  struct Vector3
  {
    double dSag;
    double dCor;
    double dTra;
  };
  struct Slice
  {
    Vector3 Normal;
    Vector3 Position;
  };
```

```
  struct SliceArray
  {
    std::vector< Slice > Slices;
  };
  bool GetSliceArray( MrProtocol::SliceArray & sa ) const;

private:
  struct Element;
  struct Internals;
  Internals *Pimpl;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const MrProtocol &d)
{
  d.Print( os );
  return os;
}

} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMMRPROTOCOL_H
```

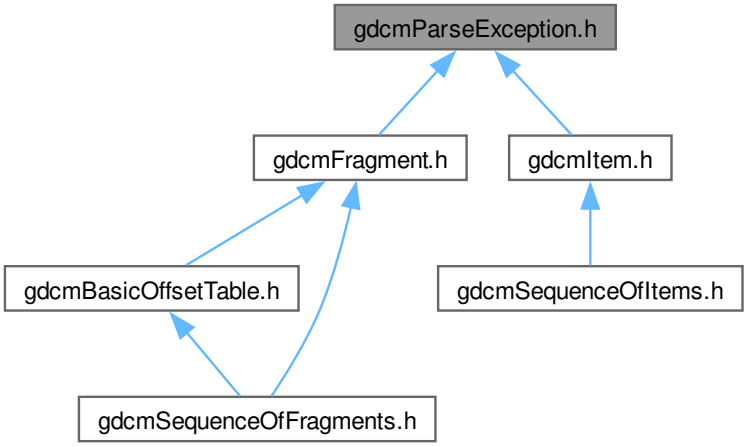
13.157 gdcmParseException.h File Reference

```
#include "gdcmException.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmParseException.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ParseException

  ParseException Standard exception handling object.

Namespaces

- namespace gdcm

13.158 gdcmParseException.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPARSEEXCEPTION_H
#define GDCMPARSEEXCEPTION_H

#include "gdcmException.h"
#include "gdcmDataElement.h"

```

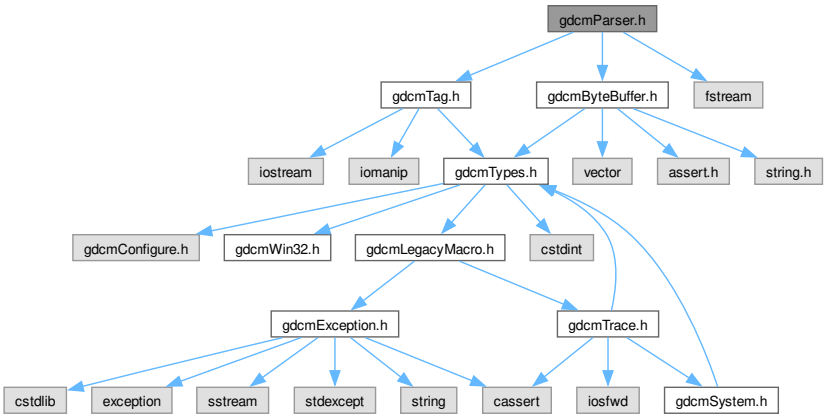
```
// Disable clang warning "dynamic exception specifications are deprecated".
// We need to be C++03 and C++11 compatible, and if we remove the 'throw()'
// specifier we'll get an error in C++03 by not matching the superclass.
#if defined(__clang__) && defined(__has_warning)
# if __has_warning("-Wdeprecated")
#  pragma clang diagnostic push
#  pragma clang diagnostic ignored "-Wdeprecated"
# endif
#endif

namespace gdcm_ns
{
class ParseException : public Exception
{
public:
  ParseException() = default;
  ~ParseException() throw() override {}

  ParseException &operator= ( const ParseException &orig )
    {
    LastElement = orig.LastElement;
    return *this;
    }
  ParseException(const ParseException& orig):Exception(orig)
    {
    LastElement = orig.LastElement;
    }

/*  virtual bool operator==( const ParseException &orig )
  {
    return true;
  }*/

/*
  // Multiple calls to what ??
  const char* what() const throw()
    {
    static std::string strwhat;
    std::ostringstream oswhat;
    oswhat « File « ":" « Line « ":\n";
    oswhat « Description;
    strwhat = oswhat.str();
    return strwhat.c_str();
    }
*/
  void SetLastElement(DataElement& de)
    {
    LastElement = de;
    }
  const DataElement& GetLastElement() const { return LastElement; }

private:
  // Store last parsed element before error:
  DataElement LastElement;
};

} // end namespace gdcm_ns

// Undo warning suppression.
#if defined(__clang__) && defined(__has_warning)
# if __has_warning("-Wdeprecated")
#  pragma clang diagnostic pop
# endif
#endif

#endif
```

13.159 gdcmParser.h File Reference

```
#include "gdcmTag.h"
#include "gdcmByteBuffer.h"
```

`#include <fstream>`

Include dependency graph for gdcmParser.h:



Classes

- class gdcm::Parser

  Parser ala XML_Parser from expat (SAX).

Namespaces

- namespace gdcm

13.160 gdcmParser.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMPARSER_H
#define GDCMPARSER_H

#include "gdcmTag.h"
#error do not use
#include "gdcmByteBuffer.h"

#include <fstream> // std::ifstream

```

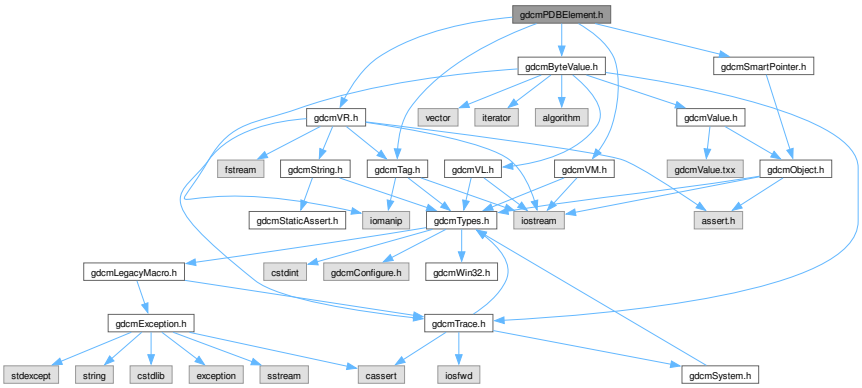
```
namespace gdcm
{
class GDCM_EXPORT Parser /*: private IStream*/
{
public:
  typedef enum {
    NoError,
    NoMemoryError,
    SyntaxError,
    NoElementsError,
    TagMismatchError,
    DuplicateAttributeError,
    JunkAfterDocElementError,
    UndefinedEntityError,
    UnexpectedStateError
  } ErrorType;

  Parser() : UserData(0),Buffer(),ErrorCode(NoError) {}
  ~Parser() {}

  // Parse some more of the document. The string s is a buffer containing
  // part (or perhaps all) of the document. The number of bytes of s that
  // are part of the document is indicated by len. This means that s
  // doesn't have to be null terminated. It also means that if len is
  // larger than the number of bytes in the block of memory that s points
  // at, then a memory fault is likely. The isFinal parameter informs the
  // parser that this is the last piece of the document. Frequently, the
  // last piece is empty (i.e. len is zero.) If a parse error occurred,
  // it returns 0. Otherwise it returns a non-zero value.
  bool Parse(const char* s, int len, bool isFinal);

  // Set handlers for start and end tags. Attributes are passed to the
  // start handler as a pointer to a vector of char pointers. Each
  // attribute seen in a start (or empty) tag occupies 2 consecutive places
  // in this vector: the attribute name followed by the attribute value.
  // These pairs are terminated by a null pointer.
  typedef void (*StartElementHandler) (void *userData,
                                       const Tag &tag,
                                       const char *atts[]);
  typedef void (*EndElementHandler) (void *userData, const Tag &name);
  void SetElementHandler(StartElementHandler start, EndElementHandler end);

  // Return what type of error has occurred.
  ErrorType GetErrorCode() const;

  // Return a string describing the error corresponding to code.
  // The code should be one of the enums that can be returned from
  // GetErrorCode.
  static const char *GetErrorString(ErrorType const &err);

  // Return the byte offset of the position.
  unsigned long GetCurrentByteIndex() const;

  // Miscellaneous functions

  // The functions in this section either obtain state information from
  // the parser or can be used to dynamically set parser options.

  // This sets the user data pointer that gets passed to handlers.
  void SetUserData(void *userData);

  // This returns the user data pointer that gets passed to handlers.
  void * GetUserData() const;

protected:

  // This is just like Parse, except in this case expat provides the buffer.
  // By obtaining the buffer from expat with the GetBuffer function,
  // the application can avoid double copying of the input.
  bool ParseBuffer(int len, bool isFinal);

  // Obtain a buffer of size len to read a piece of the document into.
  // A NULL value is returned if expat can't allocate enough memory for
  // this buffer. This has to be called prior to every call to ParseBuffer.
  char *GetBuffer(int len);

  ErrorType Process();

private:
  std::ifstream Stream;
  void* UserData;
```

```
00111   ByteBuffer Buffer;
00112   ErrorType ErrorCode;
00113
00114   StartElementHandler StartElement;
00115   EndElementHandler EndElement;
00116 };
00117
00118 } // end namespace gdcm
00119
00120 #endif //GDCMPARSER_H
```

13.161 gdcmPDBElement.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmPDBElement.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::PDBElement

  Class to represent a PDB Element.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBElement &val)

13.162 gdcmPDBElement.h

Go to the documentation of this file.

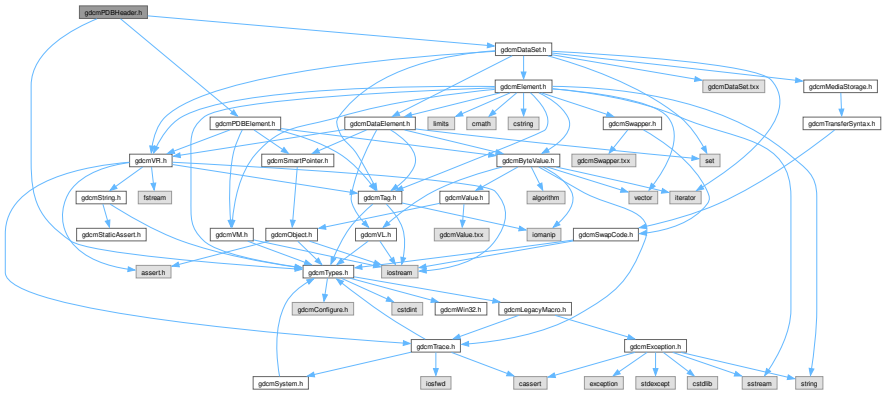
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPDBELEMENT_H
#define GDCMPDBELEMENT_H

#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

namespace gdcm
{
class GDCM_EXPORT PDBElement
{
public:
  PDBElement() = default;

  friend std::ostream& operator«(std::ostream &os, const PDBElement &val);

  const char *GetName() const { return NameField.c_str(); }
  void SetName(const char *name) { NameField = name; }

  const char *GetValue() const { return ValueField.c_str(); }
  void SetValue(const char *value) { ValueField = value; }

  bool operator==(const PDBElement &de) const
    {
    return ValueField == de.ValueField
      && NameField == de.NameField;
    }

protected:
  std::string NameField;
  std::string ValueField;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const PDBElement &val)
```

```
00056 {
00057   os « val.NameField;
00058   os « " \"";
00059   os « val.ValueField;
00060   os « "\"";
00061
00062   return os;
00063 }
00064
00065 } // end namespace gdcm
00066
00067 #endif //GDCMPDBELEMENT_H
```

13.163 gdcmPDBHeader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmPDBElement.h"
```

Include dependency graph for gdcmPDBHeader.h:



Classes

- class gdcm::PDBHeader

  Class for PDBHeader.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)

13.164 gdcmPDBHeader.h

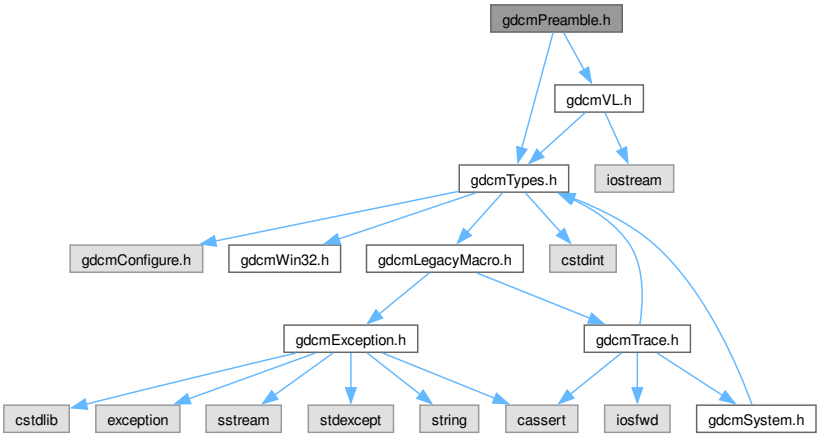
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPDBHEADER_H
#define GDCMPDBHEADER_H

#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmPDBElement.h"

namespace gdcm
{

/*
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 */
//-----------------------------------------------------------------------------

class DataElement;
class PrivateTag;
class GDCM_EXPORT PDBHeader
{
  friend std::ostream& operator«(std::ostream &_os, const PDBHeader &d);
public :
  PDBHeader() = default;
  ~PDBHeader() = default;

  bool LoadFromDataElement(DataElement const &de);

  void Print(std::ostream &os) const;

  static const PrivateTag & GetPDBInfoTag();

  const PDBElement &GetPDBElementByName(const char *name);

  bool FindPDBElementByName(const char *name);

protected:
  const PDBElement& GetPDBEEnd() const;

private:
  int readprotocoldatablock(const char *input, size_t inputlen, bool verbose);
  std::vector<PDBElement> InternalPDBDataSet;
  static PDBElement PDBEEnd;
  bool IsXML;
  std::string xmltxt;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const PDBHeader &d)
{
  d.Print( os );
  return os;
}

} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMPDBHEADER_H
```

13.165 gdcmPreamble.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVL.h"
```

Include dependency graph for gdcmPreamble.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Preamble

  DICOM Preamble (Part 10).

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)

13.166 gdcmPreamble.h

Go to the documentation of this file.

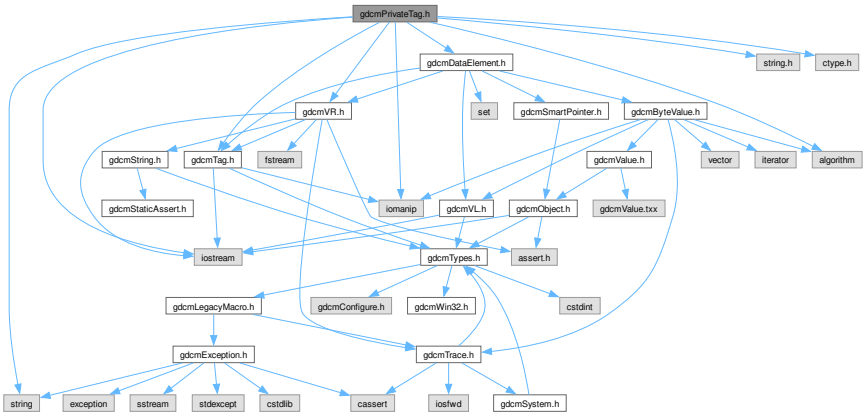
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPREAMBLE_H
#define GDCMPREAMBLE_H

#include "gdcmTypes.h"
#include "gdcmVL.h"

namespace gdcm
{

class GDCM_EXPORT Preamble
{
public:
  Preamble();
  ~Preamble();

  friend std::ostream &operator«(std::ostream &_os, const Preamble &_val);

  void Clear();

  void Valid();
  void Create();
  void Remove();

  std::istream &Read(std::istream &is);

  std::ostream const &Write(std::ostream &os) const;

  void Print(std::ostream &os) const;

  const char *GetInternal() const { return Internal; }

  bool IsEmpty() const { return !Internal; }

  VL GetLength() const { return 128 + 4; }

  Preamble(Preamble const &):Internal(nullptr)
    {
    Create();
    }
  Preamble& operator=(Preamble const &)
    {
    Create();
    return *this;
    }
protected:
  //
  bool IsValid() const {
    // is (IsValid == true) => Internal was read
    return true;
  }

private:
  char *Internal;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Preamble &val)
{
  os « val.Internal;
  return os;
}

} // end namespace gdcm
```

```
00087
00088 #endif //GDCMPREAMBLE_H
```

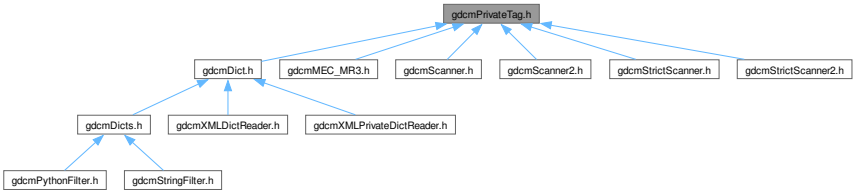
13.167 gdcmPrivateTag.h File Reference

```
#include "gdcmTag.h"
#include "gdcmVR.h"
#include "gdcmDataElement.h"
#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>
#include <string.h>
#include <ctype.h>
```

Include dependency graph for gdcmPrivateTag.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PrivateTag

  Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner).

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)

13.168 gdcmPrivateTag.h

Go to the documentation of this file.

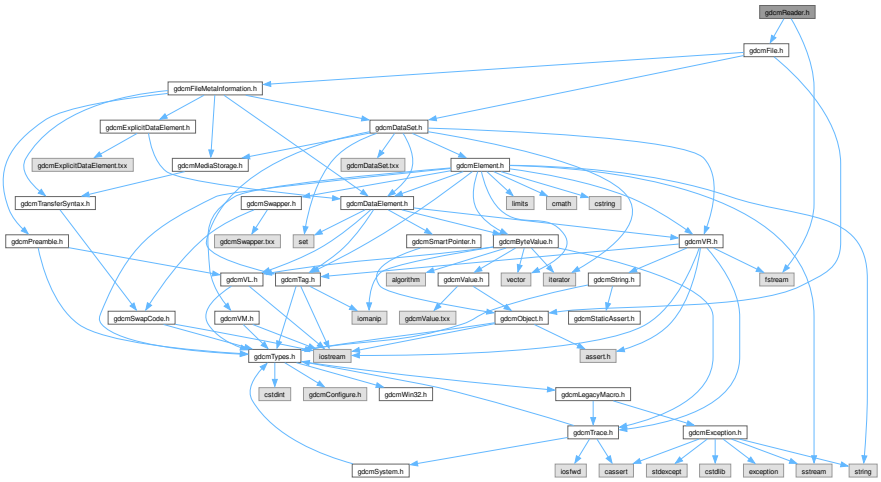
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPRIVATETAG_H
#define GDCMPRIVATETAG_H

#include "gdcmTag.h"
#include "gdcmVR.h"
#include "gdcmDataElement.h"

#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>

#include <string.h> // strlen
#include <ctype.h> // tolower

namespace gdcm_ns
{


// TODO: We could save some space since we only store 8bits for element
class GDCM_EXPORT PrivateTag : public Tag
{
  friend std::ostream& operator«(std::ostream &_os, const PrivateTag &_val);
public:
  PrivateTag(uint16_t group = 0, uint16_t element = 0, const char *owner =
"):Tag(group,element),Owner(owner ? LOComp::Trim(owner) : "") {
    // truncate the high bits
    SetElement( (uint8_t)element );
  }
  PrivateTag( Tag const & t, const char *owner = ""):Tag(t),Owner(owner ? LOComp::Trim(owner) : "") {
    // truncate the high bits
    SetElement( (uint8_t)t.GetElement());
  }

  const char *GetOwner() const { return Owner.c_str(); }
  void SetOwner(const char *owner) { if(owner) Owner = LOComp::Trim(owner); }

  PrivateTag &operator=(const PrivateTag &_val)
    {
    SetElementTag( _val.GetElementTag() );
    Owner = _val.Owner;
    return *this;
    }

  bool operator==(const Tag &_val) const
    {
```

```
    return GetElementTag() == _val.GetElementTag();
    }
  bool operator==(const PrivateTag &_val) const
    {
    return GetElementTag() == _val.GetElementTag() && Owner == _val.Owner;
    }
  bool operator!=(const Tag &_val) const
    {
    return GetElementTag() != _val.GetElementTag();
    }
  bool operator!=(const PrivateTag &_val) const
    {
    return GetElementTag() != _val.GetElementTag() || Owner != _val.Owner;
    }

  bool operator<(const PrivateTag &_val) const;

  bool ReadFromCommaSeparatedString(const char *str);

  DataElement GetAsDataElement() const;

private:
  // SIEMENS MED, GEMS_PETD_01 ...
  std::string Owner;
};

inline std::ostream& operator«(std::ostream &os, const PrivateTag &val)
{
  //assert( !val.Owner.empty() );
  os.setf( std::ios::right );
  os « std::hex « '(' « std::setw( 4 ) « std::setfill( '0' )
    « val[0] « ',' « std::setw( 2 ) « std::setfill( '0' )
    « val[1] « ',';
  os « val.Owner;
  os « ')' « std::setfill( ' ' ) « std::dec;
  return os;
}

} // end namespace gdcm_ns

#endif //GDCMPRIVATETAG_H
```

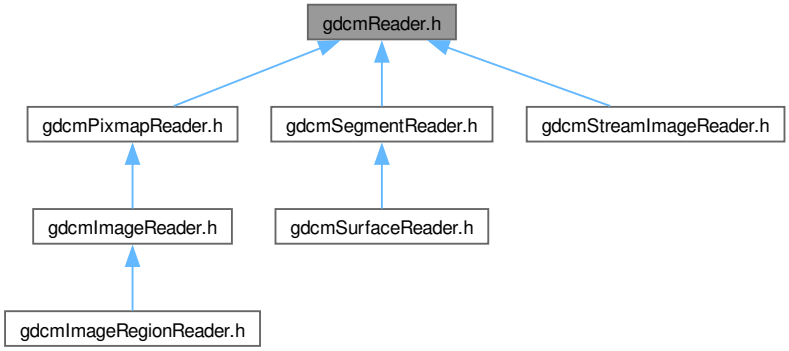
13.169 gdcmReader.h File Reference

```
#include "gdcmFile.h"
#include <fstream>
```

Include dependency graph for gdcmReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Reader

  Reader ala DOM (Document Object Model).

Namespaces

- namespace gdcm

13.170 gdcmReader.h

Go to the documentation of this file.

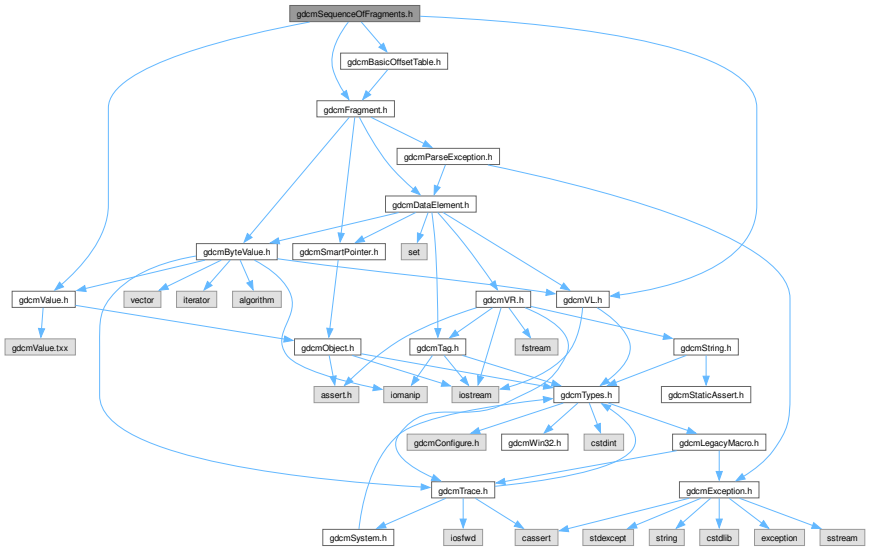
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMREADER_H
00015 #define GDCMREADER_H
00016
00017 #include "gdcmFile.h"
00018
00019 #include <fstream>
00020
00021 namespace gdcm_ns
00022 {
00023   class StreamImageReader;
00053 class GDCM_EXPORT Reader
00054 {
00055 public:
```

```
  Reader();
  virtual ~Reader();

  virtual bool Read(); // Execute()

  void SetFileName(const char *filename_native);

  void SetStream(std::istream &input_stream) {
    Stream = &input_stream;
  }

  const File &GetFile() const { return *F; }

  File &GetFile() { return *F; }

  void SetFile(File& file) { F = &file; }

  bool ReadUpToTag(const Tag & tag, std::set<Tag> const & skiptags = std::set<Tag>() );

  bool ReadSelectedTags(std::set<Tag> const & tags, bool readvalues = true);

  bool ReadSelectedPrivateTags(std::set<PrivateTag> const & ptags, bool readvalues = true);

  bool CanRead() const;

  size_t GetStreamCurrentPosition() const;

protected:
  bool ReadPreamble();
  bool ReadMetaInformation();
  bool ReadDataSet();

  SmartPointer<File> F;

  friend class StreamImageReader; //need to be friended to be able to grab the GetStreamPtr

  //this function is added for the StreamImageReader, which needs to read
  //up to the pixel data and then stops right before reading the pixel data.
  //it's used to get that position, so that reading can continue
  //apace once the read function is called.
  //so, this function gets the stream directly, and then allows for position information
  //from the tellg function, and allows for stream/pointer manip in order
  //to read the pixel data.  Note, of course, that reading pixel elements
  //will still have to be subject to endianness swaps, if necessary.
  std::istream* GetStreamPtr() const { return Stream; }

private:
  template <typename T_Caller>
  bool InternalReadCommon(const T_Caller &caller);
  TransferSyntax GuessTransferSyntax();
  std::istream *Stream;
  std::ifstream *Ifstream;

  // prevent copy/move to avoid 2 ifstream leak
  Reader(const Reader &) = delete;
  Reader &operator=(const Reader &) = delete;
  Reader(const Reader &&) = delete;
  Reader &operator=(const Reader &&) = delete;
};


} // end namespace gdcm_ns


#endif //GDCMREADER_H
```

13.171 gdcmSequenceOfFragments.h File Reference

```
#include "gdcmValue.h"
#include "gdcmVL.h"
#include "gdcmFragment.h"
```

`#include "gdcmBasicOffsetTable.h"`
Include dependency graph for gdcmSequenceOfFragments.h:



Classes

- class gdcm::SequenceOfFragments

  Class to represent a Sequence Of Fragments.

Namespaces

- namespace gdcm

13.172 gdcmSequenceOfFragments.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSEQUENCEOFFRAGMENTS_H
#define GDCMSEQUENCEOFFRAGMENTS_H

#include "gdcmValue.h"
#include "gdcmVL.h"
#include "gdcmFragment.h"
```

```
#include "gdcmBasicOffsetTable.h"

namespace gdcm_ns
{

  // FIXME gdcmSequenceOfItems and gdcmSequenceOfFragments
  // should be rethink (duplicate code)
class GDCM_EXPORT SequenceOfFragments : public Value
{
public:
  // Typdefs:
  typedef std::vector<Fragment> FragmentVector;
  typedef FragmentVector::size_type SizeType;
  typedef FragmentVector::iterator Iterator;
  typedef FragmentVector::const_iterator ConstIterator;
  Iterator Begin() { return Fragments.begin(); }
  Iterator End() { return Fragments.end(); }
  ConstIterator Begin() const { return Fragments.begin(); }
  ConstIterator End() const { return Fragments.end(); }

  SequenceOfFragments():Table(),SequenceLengthField(0xFFFFFFFF) { }

  VL GetLength() const override {
    return SequenceLengthField;
  }

  void SetLength(VL length) override {
    SequenceLengthField = length;
  }

  void Clear() override;

  void AddFragment(Fragment const &item);

  // Compute the length of all fragments (and fragments only!).
  // Basically the size of the PixelData as stored (in bytes).
  unsigned long ComputeByteLength() const;

  // Compute the length of fragments (in bytes)+ length of tag...
  // to be used for computation of Group Length
  VL ComputeLength() const;

  // Get the buffer
  bool GetBuffer(char *buffer, unsigned long length) const;
  bool GetFragBuffer(unsigned int fragNb, char *buffer, unsigned long &length) const;

  SizeType GetNumberOfFragments() const;
  const Fragment& GetFragment(SizeType num) const;

  // Write the buffer of each fragment (call WriteBuffer on all Fragments, which are
  // ByteValue). No Table information is written.
  bool WriteBuffer(std::ostream &os) const;

  const BasicOffsetTable &GetTable() const { return Table; }
  BasicOffsetTable &GetTable() { return Table; }

template <typename TSwap>
std::istream& Read(std::istream &is, bool readvalues = true)
{
  assert( SequenceLengthField.IsUndefined() );
  ReadPreValue<TSwap>(is);
  return ReadValue<TSwap>(is, readvalues);
}

template <typename TSwap>
std::istream& ReadPreValue(std::istream &is)
{
  // First item is the basic offset table:
#if 0
  try
    {
    Table.Read<TSwap>(is);
    gdcmDebugMacro( "Table: " « Table );
    }
  catch(...)
    {
    // throw "SIEMENS Icon thingy";
    // Bug_Siemens_PrivateIconNoItem.dcm
    // First thing first let's rewind
    is.seekg(-4, std::ios::cur);
    // FF D8 <=> Start of Image (SOI) marker
```

```
    // FF E0 <=> APP0 Reserved for Application Use
    if ( Table.GetTag() == Tag(0xd8ff,0xe0ff) )
      {
      Table = BasicOffsetTable(); // clear up stuff
      //Table.SetByteValue( "", 0 );
      Fragment frag;
      if( FillFragmentWithJPEG( frag, is ) )
        {
        Fragments.push_back( frag );
        }
      return is;
      }
    else
      {
      throw "Catch me if you can";
      //assert(0);
      }
    }
#else
  Table.Read<TSwap>(is);
  gdcmDebugMacro( "Table: " « Table );
#endif
  return is;
}

template <typename TSwap>
std::istream& ReadValue(std::istream &is, bool /*readvalues*/)
{
  const Tag seqDelItem(0xfffe,0xe0dd);
  // not used for now...
  Fragment frag;
  try
    {
    while( frag.Read<TSwap>(is) && frag.GetTag() != seqDelItem )
      {
      //gdcmDebugMacro( "Frag: " « frag );
      Fragments.push_back( frag );
      }
    assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
    }
  catch(Exception &ex)
    {
    (void)ex;
#ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
    // that's ok ! In all cases the whole file was read, because
    // Fragment::Read only fail on eof() reached 1.
    // SIEMENS-JPEG-CorruptFrag.dcm is more difficult to deal with, we have a
    // partial fragment, read we decide to add it anyway to the stack of
    // fragments (eof was reached so we need to clear error bit)
    if( frag.GetTag() == Tag(0xfffe,0xe000)  )
      {
      gdcmWarningMacro( "Pixel Data Fragment could be corrupted. Use file at own risk" );
      Fragments.push_back( frag );
      is.clear(); // clear the error bit
      }
    // 2. GENESIS_SIGNA-JPEG-CorruptFrag.dcm
    else if ( frag.GetTag() == Tag(0xddff,0x00e0) )
      {
      assert( Fragments.size() == 1 );
      const ByteValue *bv = Fragments[0].GetByteValue();
      assert( bv->GetLength() >= 1 && (unsigned char)bv->GetPointer()[ bv->GetLength() - 1 ] == 0xfe );
      // Yes this is an extra copy, this is a bug anyway, go fix YOUR code
      Fragments[0].SetByteValue( bv->GetPointer(), bv->GetLength() - 1 );
      gdcmWarningMacro( "JPEG Fragment length was declared with an extra byte"
        " at the end: stripped !" );
      is.clear(); // clear the error bit
      }
    // 3. LEICA/WSI
    else if ( (frag.GetTag().GetGroup() == 0x00ff)
      && ((frag.GetTag().GetElement() & 0x00ff) == 0xe0) )
      {
      // Looks like there is a mess with offset and odd byte array
      // We are going first to backtrack one byte back, and then use a
      // ReadBacktrack function which in turn may backtrack up to 10 bytes
      // backward. This appears to be working on a set of DICOM/WSI files from
      // LEICA
      gdcmWarningMacro( "Trying to fix the even-but-odd value length bug #1" );
      assert( Fragments.size() );
      const size_t lastf = Fragments.size() - 1;
      const ByteValue *bv = Fragments[ lastf ].GetByteValue();
      const char *a = bv->GetPointer();
```

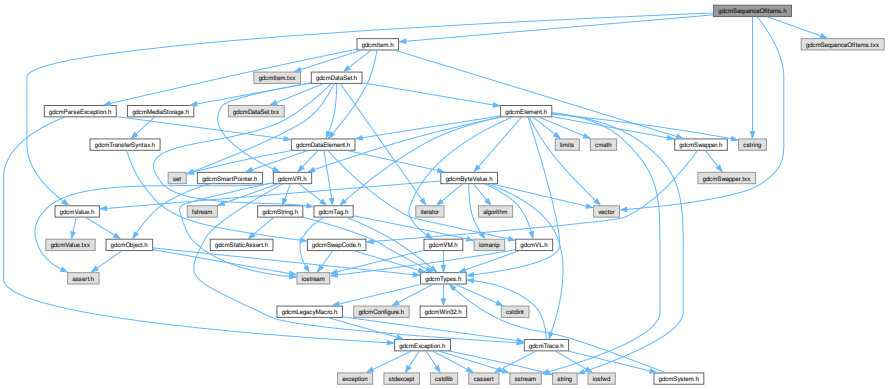
```
      gdcmAssertAlwaysMacro( bv->GetLength() >= 1 && (unsigned char)a[ bv->GetLength() - 1 ] == 0xfe );
      Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 1 );
      is.seekg( -9, std::ios::cur );
      assert( is.good() );
      while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
        {
        gdcmDebugMacro( "Frag: " « frag );
        Fragments.push_back( frag );
        }
      assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
      }
    // 4. LEICA/WSI (bis)
    else if ( frag.GetTag().GetGroup() == 0xe000 )
      {
      // Looks like there is a mess with offset and odd byte array
      // We are going first to backtrack one byte back, and then use a
      // ReadBacktrack function which in turn may backtrack up to 10 bytes
      // backward. This appears to be working on a set of DICOM/WSI files from
      // LEICA
      gdcmWarningMacro( "Trying to fix the even-but-odd value length bug #2" );
      assert( Fragments.size() );
      const size_t lastf = Fragments.size() - 1;
      const ByteValue *bv = Fragments[ lastf ].GetByteValue();
      const char *a = bv->GetPointer();
      gdcmAssertAlwaysMacro( bv->GetLength() >= 2 && (unsigned char)a[ bv->GetLength() - 2 ] == 0xfe );
      Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 2 );
      is.seekg( -10, std::ios::cur );
      assert( is.good() );
      while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
        {
        gdcmDebugMacro( "Frag: " « frag );
        Fragments.push_back( frag );
        }
      assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
      }
    // 5. LEICA/WSI (ter)
    else if ( (frag.GetTag().GetGroup() & 0x00ff) == 0x00e0
    && (frag.GetTag().GetElement() & 0xff00) == 0x0000 )
      {
      // Looks like there is a mess with offset and odd byte array
      // We are going first to backtrack one byte back, and then use a
      // ReadBacktrack function which in turn may backtrack up to 10 bytes
      // backward. This appears to be working on a set of DICOM/WSI files from
      // LEICA
      gdcmWarningMacro( "Trying to fix the even-but-odd value length bug #3" );
      assert( Fragments.size() );
      const size_t lastf = Fragments.size() - 1;
      const ByteValue *bv = Fragments[ lastf ].GetByteValue();
      const char *a = bv->GetPointer();
      gdcmAssertAlwaysMacro( bv->GetLength() >= 3 && (unsigned char)a[ bv->GetLength() - 3 ] == 0xfe );
      Fragments[ lastf ].SetByteValue( bv->GetPointer(), bv->GetLength() - 3 );
      is.seekg( -11, std::ios::cur );
      assert( is.good() );
      while( frag.ReadBacktrack<TSwap>(is) && frag.GetTag() != seqDelItem )
        {
        gdcmDebugMacro( "Frag: " « frag );
        Fragments.push_back( frag );
        }
      assert( frag.GetTag() == seqDelItem && frag.GetVL() == 0 );
      }
    else
      {
      // 3. gdcm-JPEG-LossLess3a.dcm: easy case, an extra tag was found
      // instead of terminator (eof is the next char)
      gdcmWarningMacro( "Reading failed at Tag:" « frag.GetTag() « " Index #"
        « Fragments.size() « " Offset " « is.tellg() « ". Use file at own risk."
        « ex.what() );
      }
#endif /* GDCM_SUPPORT_BROKEN_IMPLEMENTATION */
    }

  return is;
}

template <typename TSwap>
std::ostream const &Write(std::ostream &os) const
{
  if( !Table.Write<TSwap>(os) )
    {
    assert(0 && "Should not happen");
    return os;
```

```
    }
  for(ConstIterator it = Begin();it != End(); ++it)
    {
    it->Write<TSwap>(os);
    }
  // seq del item is not stored, write it !
  const Tag seqDelItem(0xfffe,0xe0dd);
  seqDelItem.Write<TSwap>(os);
  VL zero = 0;
  zero.Write<TSwap>(os);

  return os;
}

//#if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA)
  // For now leave it there, this does not make sense in the C++ layer
  // Create a new object
  static SmartPointer<SequenceOfFragments> New()
  {
     return new SequenceOfFragments();
  }
//#endif

protected:
public:
  void Print(std::ostream &os) const override {
    os « "SQ L= " « SequenceLengthField « "\n";
    os « "Table:" « Table « "\n";
    for(ConstIterator it = Begin();it != End(); ++it)
      {
      os « "  " « *it « "\n";
      }
    assert( SequenceLengthField.IsUndefined() );
      {
      const Tag seqDelItem(0xfffe,0xe0dd);
      VL zero = 0;
      os « seqDelItem;
      os « "\t" « zero;
      }
  }
  bool operator==(const Value &val) const override
    {
    const SequenceOfFragments &sqf = dynamic_cast<const SequenceOfFragments&>(val);
    return Table == sqf.Table &&
      SequenceLengthField == sqf.SequenceLengthField &&
      Fragments == sqf.Fragments;
    }

private:
  BasicOffsetTable Table;
  VL SequenceLengthField;
  FragmentVector Fragments;

private:
  bool FillFragmentWithJPEG( Fragment & frag, std::istream & is );
};


} // end namespace gdcm_ns

#endif //GDCMSEQUENCEOFFRAGMENTS_H
```

13.173 gdcmSequenceOfItems.h File Reference

```
#include "gdcmValue.h"
#include "gdcmItem.h"
#include <vector>
#include <cstring>
#include "gdcmSequenceOfItems.txx"
```

Include dependency graph for gdcmSequenceOfItems.h:



Classes

- class gdcm::SequenceOfItems

  Class to represent a Sequence Of Items.

Namespaces

- namespace gdcm

13.174 gdcmSequenceOfItems.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMSEQUENCEOFITEMS_H
#define GDCMSEQUENCEOFITEMS_H

#include "gdcmValue.h"
#include "gdcmItem.h"

#include <vector>
#include <cstring> // strcmp

namespace gdcm_ns
{

class GDCM_EXPORT SequenceOfItems : public Value
{
public:
```

```
// Typdefs:
typedef std::vector< Item > ItemVector;
typedef ItemVector::size_type SizeType;
typedef ItemVector::iterator Iterator;
typedef ItemVector::const_iterator ConstIterator;
Iterator Begin() { return Items.begin(); }
Iterator End() { return Items.end(); }
ConstIterator Begin() const { return Items.begin(); }
ConstIterator End() const { return Items.end(); }

SequenceOfItems():SequenceLengthField(0xFFFFFFFF) { }
//SequenceOfItems(VL const &vl = 0xFFFFFFFF):SequenceLengthField(vl),NType(type) { }

VL GetLength() const override { return SequenceLengthField; }
void SetLength(VL length) override {
  SequenceLengthField = length;
}
void SetLengthToUndefined();
bool IsUndefinedLength() const {
  return SequenceLengthField.IsUndefined();
}

template <typename TDE>
VL ComputeLength() const;

void Clear() override;

void AddItem(Item const &item);

Item & AddNewUndefinedLengthItem();

bool RemoveItemByIndex( const SizeType index );

bool IsEmpty() const { return Items.empty(); }
SizeType GetNumberOfItems() const {  return Items.size(); }
void SetNumberOfItems(SizeType n) {  Items.resize(n); }

/* WARNING: first item is #1 (see DICOM standard)
 *  Each Item shall be implicitly assigned an ordinal position starting with the value 1 for the
 * first Item in the Sequence, and incremented by 1 with each subsequent Item. The last Item in the
 * Sequence shall have an ordinal position equal to the number of Items in the Sequence.
 */
const Item &GetItem(SizeType position) const;
Item &GetItem(SizeType position);

SequenceOfItems &operator=(const SequenceOfItems &val) {
  SequenceLengthField = val.SequenceLengthField;
  Items = val.Items;
  return *this;
  }

template <typename TDE, typename TSwap>
std::istream &Read(std::istream &is, bool readvalues = true)
  {
  (void)readvalues;
  const Tag seqDelItem(0xfffe,0xe0dd);
  if( SequenceLengthField.IsUndefined() )
    {
    Item item;
    while( item.Read<TDE,TSwap>(is) && item.GetTag() != seqDelItem )
      {
      //gdcmDebugMacro( "Item: " « item );
      assert( item.GetTag() != seqDelItem );
      Items.push_back( item );
      item.Clear();
      }
    //assert( item.GetTag() == seqDelItem && item.GetVL() == 0 );
    }
  else
    {
    Item item;
    VL l = 0;
    //is.seekg( SequenceLengthField, std::ios::cur ); return is;
    while( l != SequenceLengthField )
      {
      try
        {
        item.Read<TDE,TSwap>(is);
        }
      catch( Exception &ex )
        {
```

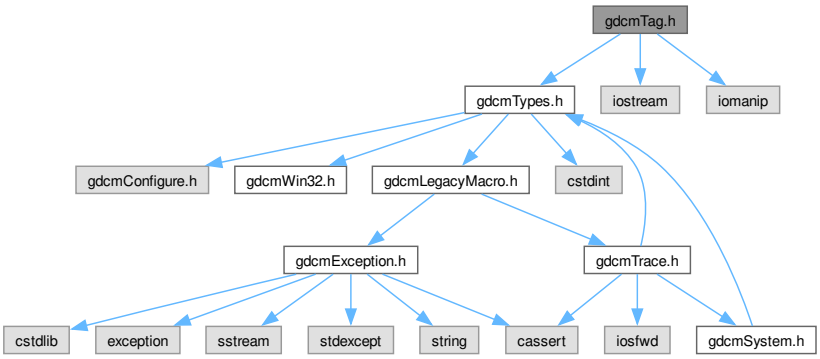
```
          if( strcmp( ex.GetDescription(), "Changed Length" ) == 0 )
            {
            VL newlength = l + item.template GetLength<TDE>();
            if( newlength > SequenceLengthField )
              {
              // BogugsItemAndSequenceLength.dcm
              gdcmWarningMacro( "SQ length is wrong" );
              SequenceLengthField = newlength;
              }
            }
          else
            {
            throw ex;
            }
          }
#ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
        if( item.GetTag() == seqDelItem )
          {
          gdcmWarningMacro( "SeqDelItem found in defined length Sequence. Skipping" );
          assert( item.GetVL() == 0 );
          assert( item.GetNestedDataSet().Size() == 0 );
          // we need to pay attention that the length of the Sequence of Items will be wrong
          // this way. Indeed by not adding this item we are changing the size of this sqi
          }
        else // Not a seq del item marker
#endif
          {
          // By design we never load them. If we were to load those attribute
          // as normal item it would become very complex to convert a sequence
          // from defined length to undefined length with the risk to write two
          // seq del marker
          Items.push_back( item );
          }
        l += item.template GetLength<TDE>();
        if( l > SequenceLengthField )
          {
          gdcmDebugMacro( "Found: Length of Item larger than expected" );
          throw "Length of Item larger than expected";
          }
        assert( l <= SequenceLengthField );
        //std::cerr « "sqi debug len: " « is.tellg() « " " «  l « " " «  SequenceLengthField « std::endl;
#ifdef GDCM_SUPPORT_BROKEN_IMPLEMENTATION
        // MR_Philips_Intera_No_PrivateSequenceImplicitVR.dcm
        // (0x2005, 0x1080): for some reason computation of length fails...
        if( SequenceLengthField == 778 && l == 774 )
          {
          gdcmWarningMacro( "PMS: Super bad hack" );
          SequenceLengthField = l;
          throw Exception( "Wrong Length" );
          //l = SequenceLengthField;
          }
        // Bug_Philips_ItemTag_3F3F
        // (0x2005, 0x1080): Because we do not handle fully the bug at the item
        // level we need to check here too
        else if ( SequenceLengthField == 444 && l == 3*71 )
          {
          // This one is a double bug. Item length is wrong and impact SQ length
          gdcmWarningMacro( "PMS: Super bad hack" );
          l = SequenceLengthField;
          }
#endif
        }
      assert( l == SequenceLengthField );
      }
    return is;
    }

  template <typename TDE,typename TSwap>
  std::ostream const &Write(std::ostream &os) const
    {
    typename ItemVector::const_iterator it = Items.begin();
    for(;it != Items.end(); ++it)
      {
      it->Write<TDE,TSwap>(os);
      }
    if( SequenceLengthField.IsUndefined() )
      {
      // seq del item is not stored, write it !
      const Tag seqDelItem(0xfffe,0xe0dd);
      seqDelItem.Write<TSwap>(os);
      VL zero = 0;
```

```
      zero.Write<TSwap>(os);
      }

    return os;
    }

//protected:
  void Print(std::ostream &os) const override {
    os « "\t(" « SequenceLengthField « ")\n";
    ItemVector::const_iterator it =
      Items.begin();
    for(;it != Items.end(); ++it)
      {
      os « "  " « *it;
      }
    if( SequenceLengthField.IsUndefined() )
      {
      const Tag seqDelItem(0xfffe,0xe0dd);
      VL zero = 0;
      os « seqDelItem;
      os « "\t" « zero;
      }
  }

  static SmartPointer<SequenceOfItems> New()
  {
     return new SequenceOfItems;
  }
  bool FindDataElement(const Tag &t) const;

  bool operator==(const Value &val) const override
    {
    const SequenceOfItems &sqi = dynamic_cast<const SequenceOfItems&>(val);
    return SequenceLengthField == sqi.SequenceLengthField &&
      Items == sqi.Items;
    }

private:
public:
  VL SequenceLengthField;
  ItemVector Items;
};

} // end namespace gdcm_ns

#include "gdcmSequenceOfItems.txx"

#endif //GDCMSEQUENCEOFITEMS_H
```

13.175 gdcmTag.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
#include <iomanip>
```

Include dependency graph for gdcmTag.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Tag

  Class to represent a DICOM Data Element (Attribute) Tag (Group, Element).

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)
- std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)

13.176 gdcmTag.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTAG_H
#define GDCMTAG_H

#include "gdcmTypes.h"

#include <iostream>
#include <iomanip>

namespace gdcm
{

class GDCM_EXPORT Tag
{
public:
  Tag(uint16_t group, uint16_t element) {
    ElementTag.tags[0] = group; ElementTag.tags[1] = element;
  }
  Tag(uint32_t tag = 0) {
    SetElementTag(tag);
  }

  friend std::ostream& operator«(std::ostream &_os, const Tag &_val);
  friend std::istream& operator»(std::istream &_is, Tag &_val);

  uint16_t GetGroup() const { return ElementTag.tags[0]; }
  uint16_t GetElement() const { return ElementTag.tags[1]; }
  void SetGroup(uint16_t group) { ElementTag.tags[0] = group; }
  void SetElement(uint16_t element) { ElementTag.tags[1] = element; }
  void SetElementTag(uint16_t group, uint16_t element) {
    ElementTag.tags[0] = group; ElementTag.tags[1] = element;
  }

  uint32_t GetElementTag() const {
#ifndef GDCM_WORDS_BIGENDIAN
    return (ElementTag.tag«16) | (ElementTag.tag»16);
#else
    return ElementTag.tag;
#endif
  }

  void SetElementTag(uint32_t tag) {
#ifndef GDCM_WORDS_BIGENDIAN
    tag = ( (tag«16) | (tag»16) );
#endif
    ElementTag.tag = tag;
  }

  const uint16_t &operator[](const unsigned int &_id) const
    {
    assert(_id<2);
    return ElementTag.tags[_id];
    }
  uint16_t &operator[](const unsigned int &_id)
    {
    assert(_id<2);
    return ElementTag.tags[_id];
    }

  Tag &operator=(const Tag &_val)
    {
    ElementTag.tag = _val.ElementTag.tag;
    return *this;
```

```
    }

  bool operator==(const Tag &_val) const
    {
    return ElementTag.tag == _val.ElementTag.tag;
    }
  bool operator!=(const Tag &_val) const
    {
    return ElementTag.tag != _val.ElementTag.tag;
    }

  // FIXME FIXME FIXME TODO
  // the following is pretty dumb. Since we have control over who is group
  // and who is element, we should reverse them in little endian and big endian case
  // since what we really want is fast comparison and not guarantee that group is in #0
  // ...
  bool operator<(const Tag &_val) const
    {
#ifndef GDCM_WORDS_BIGENDIAN
    if( ElementTag.tags[0] < _val.ElementTag.tags[0] )
      return true;
    if( ElementTag.tags[0] == _val.ElementTag.tags[0]
      && ElementTag.tags[1] <  _val.ElementTag.tags[1] )
      return true;
    return false;
#else
    // Plain comparison is enough!
    return ( ElementTag.tag < _val.ElementTag.tag );
#endif
    }
  bool operator<=(const Tag &t2) const
    {
    const Tag &t1 = *this;
    return t1 == t2 || t1 < t2;
    }

  Tag(const Tag &_val)
    {
    ElementTag.tag = _val.ElementTag.tag;
    }
  uint32_t GetLength() const { return 4; }

  bool IsPublic() const { return !(ElementTag.tags[0] % 2); }

  bool IsPrivate() const { return !IsPublic(); }

  //-----------------------------------------------------------------------------
  template <typename TSwap>
  std::istream &Read(std::istream &is)
    {
    if( is.read(ElementTag.bytes, 4) )
      TSwap::SwapArray(ElementTag.tags, 2);
    return is;
    }

  template <typename TSwap>
  const std::ostream &Write(std::ostream &os) const
    {
    uint16_t copy[2];
    copy[0]= ElementTag.tags[0];
    copy[1]= ElementTag.tags[1];
    TSwap::SwapArray(copy, 2);
    return os.write((char*)(&copy), 4);
    }

  Tag GetPrivateCreator() const
    {
    // See PS 3.5 - 7.8.1 PRIVATE DATA ELEMENT TAGS
    // eg: 0x0123,0x1425 -> 0x0123,0x0014
    if( IsPrivate() && !IsPrivateCreator() )
      {
      Tag r = *this;
      r.SetElement( (uint16_t)(GetElement() » 8) );
      return r;
      }
    if( IsPrivateCreator() ) return *this;
    return Tag(0x0,0x0);
    }
  void SetPrivateCreator(Tag const &t)
    {
    // See PS 3.5 - 7.8.1 PRIVATE DATA ELEMENT TAGS
```

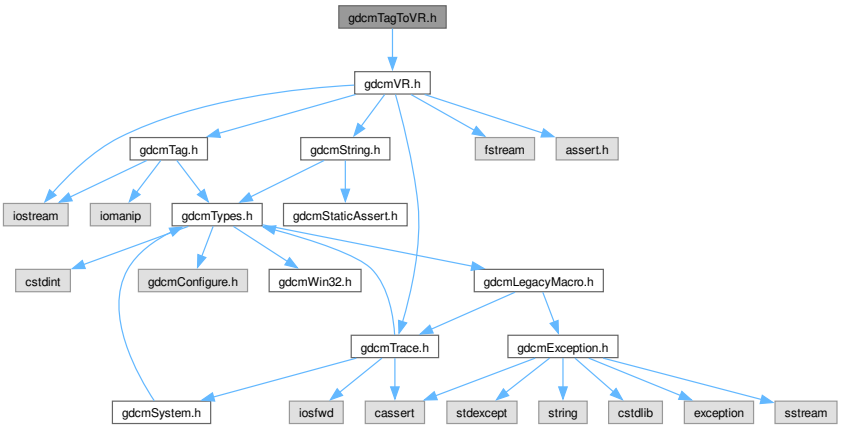
```
    // eg: 0x0123,0x0045 -> 0x0123,0x4567
    assert( t.IsPrivate() /*&& t.IsPrivateCreator()*/ );
    const uint16_t element = (uint16_t)(t.GetElement() « 8);
    const uint16_t base = (uint16_t)(GetElement() « 8);
    SetElement( (uint16_t)((base » 8) + element) );
    SetGroup( t.GetGroup() );
    }

  bool IsPrivateCreator() const
    {
    return IsPrivate() && (GetElement() <= 0xFF && GetElement() >= 0x10);
    }

  bool IsIllegal() const
    {
    // DICOM reserved those groups:
    return GetGroup() == 0x0001 || GetGroup() == 0x0003 || GetGroup() == 0x0005 || GetGroup() == 0x0007
    // This is a very special case, in private group, one cannot use element [0x01,0x09] ...
//      || (IsPrivate() && !IsPrivateCreator() && !IsGroupLength());
      || (IsPrivate() && GetElement() > 0x0 && GetElement() < 0x10 );
    }

  bool IsGroupLength() const
    {
    return GetElement() == 0x0;
    }

  bool IsGroupXX(const Tag &t) const
    {
    if( t.GetElement() == GetElement() )
      {
      if( t.IsPrivate() ) return false;
      uint16_t group = (uint16_t)((GetGroup() » 8 ) « 8);
      return group == t.GetGroup();
      }
    return false;
    }

  bool ReadFromCommaSeparatedString(const char *str);

  bool ReadFromContinuousString(const char *str);

  std::string PrintAsContinuousString() const;

  std::string PrintAsContinuousUpperCaseString() const;

  bool ReadFromPipeSeparatedString(const char *str);

  std::string PrintAsPipeSeparatedString() const;

private:
  union { uint32_t tag; uint16_t tags[2]; char bytes[4]; } ElementTag;
};
//-----------------------------------------------------------------------------
inline std::istream& operator»(std::istream &_is, Tag &_val)
{
  char c;
  _is » c;
  uint16_t a, b;
  _is » std::hex » a;
  //_is » std::hex » _val[0];
  //_is » std::hex » _val.ElementTag.tags[0];
  _is » c;
  //_is » _val[1];
  //_is » std::hex » _val.ElementTag.tags[1];
  _is » std::hex » b;
  _is » c;
  _val.SetGroup( a );
  _val.SetElement( b );
  return _is;
}

inline std::ostream& operator«(std::ostream &_os, const Tag &_val)
{
  _os.setf( std::ios::right);
  _os « std::hex « '(' « std::setw( 4 ) « std::setfill( '0' )
    « _val[0] « ',' « std::setw( 4 ) « std::setfill( '0' )
    « _val[1] « ')' « std::setfill( ' ' ) « std::dec;
  return _os;
}
```

```
} // end namespace gdcm

#endif //GDCMTAG_H
```

13.177 gdcmTagToVR.h File Reference

`#include "gdcmVR.h"`

Include dependency graph for gdcmTagToVR.h:



Namespaces

- namespace gdcm

Functions

- VR::VRType gdcm::GetVRFromTag (Tag const &tag)

13.178 gdcmTagToVR.h

Go to the documentation of this file.

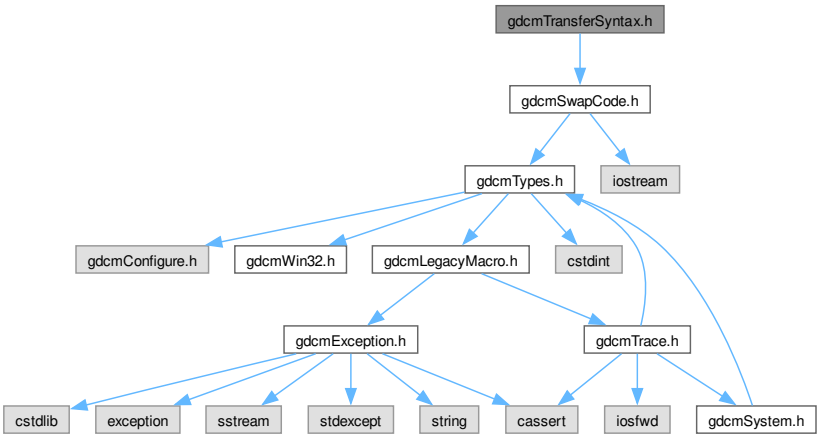
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
```

```
#ifndef GDCMTAGTOVR_H
#define GDCMTAGTOVR_H

#include "gdcmVR.h"

namespace gdcm
{
  class Tag;
  VR::VRType GetVRFromTag( Tag const & tag );
}

#endif // GDCMTAGTOVR_H
```

13.179 gdcmTransferSyntax.h File Reference

`#include "gdcmSwapCode.h"`

Include dependency graph for gdcmTransferSyntax.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::TransferSyntax

  Class to manipulate Transfer Syntax.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)

13.180 gdcmTransferSyntax.h

Go to the documentation of this file.

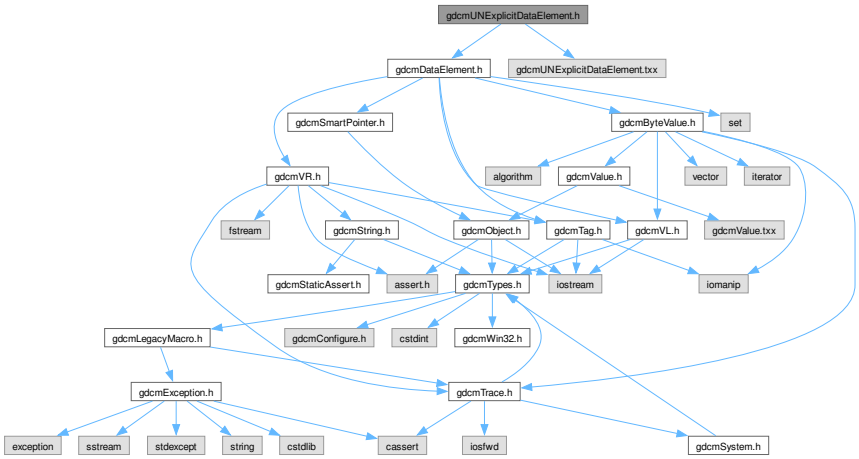
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTRANSFERSYNTAX_H
#define GDCMTRANSFERSYNTAX_H

#include "gdcmSwapCode.h"

namespace gdcm
{

class GDCM_EXPORT TransferSyntax
{
public:
  typedef enum {
    Unknown = 0,
    Explicit,
    Implicit
  } NegociatedType;

#if 0
  //NOT FLEXIBLE, since forces user to update lib every time new module
  //comes out...
  // TODO
  typedef enum {
    NoSpacing = 0,
    PixelSpacing,
    ImagerPixelSpacing,
    PixelAspectRatio
  } ImageSpacingType;
  ImageSpacingType GetImageSpacing();
#endif

  typedef enum {
    ImplicitVRLittleEndian = 0,
    ImplicitVRBigEndianPrivateGE,
    ExplicitVRLittleEndian,
    DeflatedExplicitVRLittleEndian,
    ExplicitVRBigEndian,
    JPEGBaselineProcess1,
    JPEGExtendedProcess2_4,
    JPEGExtendedProcess3_5,
    JPEGSpectralSelectionProcess6_8,
    JPEGFullProgressionProcess10_12,
    JPEGLosslessProcess14,
    JPEGLosslessProcess14_1,
    JPEGLSLossless,
    JPEGLSNearLossless,
    JPEG2000Lossless,
    JPEG2000,
    JPEG2000Part2Lossless,
    JPEG2000Part2,
    RLELossless,
    MPEG2MainProfile,
    ImplicitVRBigEndianACRNEMA,
    WeirdPapryus,
```

```
    CT_private_ELE,
    JPIPReferenced,
    MPEG2MainProfileHighLevel,
    MPEG4AVCH264HighProfileLevel4_1,
    MPEG4AVCH264BDcompatibleHighProfileLevel4_1,
    TS_END
  } TSType;

  // Return the string as written in the official DICOM dict from
  // a custom enum type
  static const char* GetTSString(TSType ts);
  static TSType GetTSType(const char *str);

  NegociatedType GetNegociatedType() const;

  SwapCode GetSwapCode() const;

  bool IsValid() const { return TSField != TS_END; }

  operator TSType () const { return TSField; }

  // FIXME: ImplicitVRLittleEndian used to be the default, but nowadays
  // this is rather the ExplicitVRLittleEndian instead...should be change the default ?
  TransferSyntax(TSType type = ImplicitVRLittleEndian):TSField(type) {}

  // return if dataset is encoded or not (Deflate Explicit VR)
  bool IsEncoded() const;

  bool IsImplicit() const;
  bool IsExplicit() const;

  bool IsEncapsulated() const;

  bool IsLossy() const;
  bool IsLossless() const;
  bool CanStoreLossy() const;

  const char *GetString() const { return TransferSyntax::GetTSString(TSField); }

  friend std::ostream &operator«(std::ostream &os, const TransferSyntax &ts);
private:
  // DO NOT EXPOSE the following. Internal details of TransferSyntax
bool IsImplicit(TSType ts) const;
bool IsExplicit(TSType ts) const;
bool IsLittleEndian(TSType ts) const;
bool IsBigEndian(TSType ts) const;

  TSType TSField;
};
//-----------------------------------------------------------------------------
inline std::ostream &operator«(std::ostream &_os, const TransferSyntax &ts)
{
  _os « TransferSyntax::GetTSString(ts);
  return _os;

}

} // end namespace gdcm

#endif //GDCMTRANSFERSYNTAX_H
```

13.181 gdcmUNExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmUNExplicitDataElement.txx"
```

Include dependency graph for gdcmUNExplicitDataElement.h:



Classes

- class gdcm::UNExplicitDataElement

  Class to read/write a DataElement as UNExplicit Data Element.

Namespaces

- namespace gdcm

13.182 gdcmUNExplicitDataElement.h

Go to the documentation of this file.

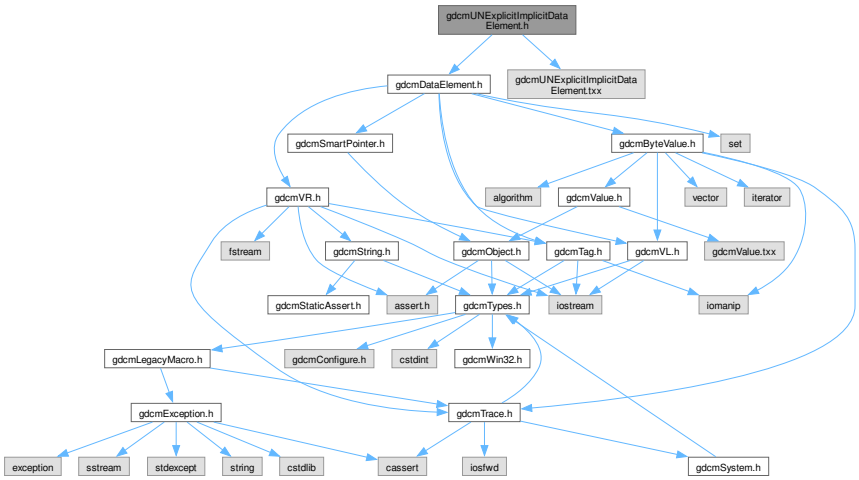
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMUNEXPLICITDATAELEMENT_H
#define GDCMUNEXPLICITDATAELEMENT_H

#include "gdcmDataElement.h"

namespace gdcm
{
// Data Element (UNExplicit)
class GDCM_EXPORT UNExplicitDataElement : public DataElement
{
public:
```

```
  VL GetLength() const;

  template <typename TSwap>
  std::istream &Read(std::istream &is);

  template <typename TSwap>
  std::istream &ReadPreValue(std::istream &is);

  template <typename TSwap>
  std::istream &ReadValue(std::istream &is, bool readvalues = true);

  template <typename TSwap>
  std::istream &ReadWithLength(std::istream &is, VL & length);

  // PURPOSELY do not provide an implementation for writing !
  //template <typename TSwap>
  //const std::ostream &Write(std::ostream &os) const;
};

} // end namespace gdcm

#include "gdcmUNExplicitDataElement.txx"

#endif //GDCMUNEXPLICITDATAELEMENT_H
```

13.183 gdcmUNExplicitImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmUNExplicitImplicitDataElement.txx"
```

Include dependency graph for gdcmUNExplicitImplicitDataElement.h:



Classes

- class gdcm::UNExplicitImplicitDataElement

  Class to read/write a DataElement as ExplicitImplicit Data Element.

Namespaces

- namespace gdcm

13.184 gdcmUNExplicitImplicitDataElement.h

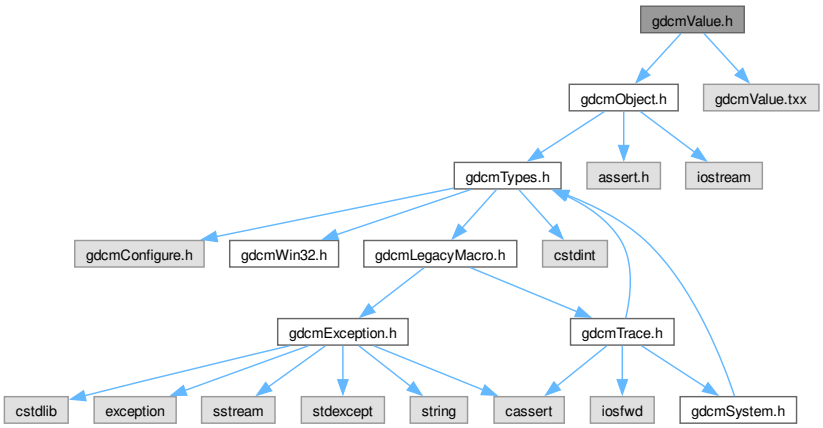
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMUNEXPLICITIMPLICITDATAELEMENT_H
#define GDCMUNEXPLICITIMPLICITDATAELEMENT_H

#include "gdcmDataElement.h"

namespace gdcm
{
// Data Element (ExplicitImplicit)
class GDCM_EXPORT UNExplicitImplicitDataElement : public DataElement
{
public:
  VL GetLength() const;

  template <typename TSwap>
  std::istream &Read(std::istream &is);

  template <typename TSwap>
  std::istream &ReadPreValue(std::istream &is);

  template <typename TSwap>
  std::istream &ReadValue(std::istream &is);

  // PURPOSELY do not provide an implementation for writing !
  //template <typename TSwap>
  //const std::ostream &Write(std::ostream &os) const;
};

} // end namespace gdcm

#include "gdcmUNExplicitImplicitDataElement.txx"

#endif //GDCMUNEXPLICITIMPLICITDATAELEMENT_H
```

13.185 gdcmValue.h File Reference

```
#include "gdcmObject.h"
#include "gdcmValue.txx"
```

Include dependency graph for gdcmValue.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::Value

  Class to represent the value of a Data Element.

Namespaces

- namespace gdcm

13.186 gdcmValue.h

Go to the documentation of this file.

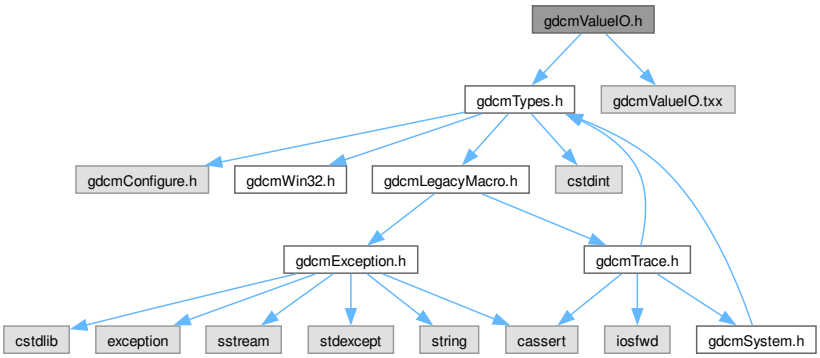
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.
```

```
00012
00013 ==========================================================================*/
00014 #ifndef GDCMVALUE_H
00015 #define GDCMVALUE_H
00016
00017 #include "gdcmObject.h"
00018
00019 namespace gdcm { class VL; }
00020 namespace gdcm_ns
00021 {
00022 #if !defined(SWIGPYTHON) && !defined(SWIGCSHARP) && !defined(SWIGJAVA) && !defined(SWIGPHP)
00023 using namespace gdcm;
00024 #endif
00031 class GDCM_EXPORT Value : public Object
00032 {
00033 public:
00034   Value() = default;
00035   ~Value() override = default;
00036
00037   virtual VL GetLength() const = 0;
00038   virtual void SetLength(VL l) = 0;
00039
00040   virtual void Clear() = 0;
00041
00042   virtual bool operator==(const Value &val) const = 0;
00043
00044 protected:
00045   friend class DataElement;
00046   virtual void SetLengthOnly(VL l);
00047 };
00048
00049 } // end namespace gdcm_ns
00050
00051 #include "gdcmValue.txx"
00052
00053 #endif //GDCMVALUE_H
```

13.187 gdcmValueIO.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmValueIO.txx"
```

Include dependency graph for gdcmValueIO.h:



Classes

- class gdcm::ValueIO< TDE, TSwap, TType >

  Class to dispatch template calls.

Namespaces

- namespace gdcm

13.188 gdcmValueIO.h

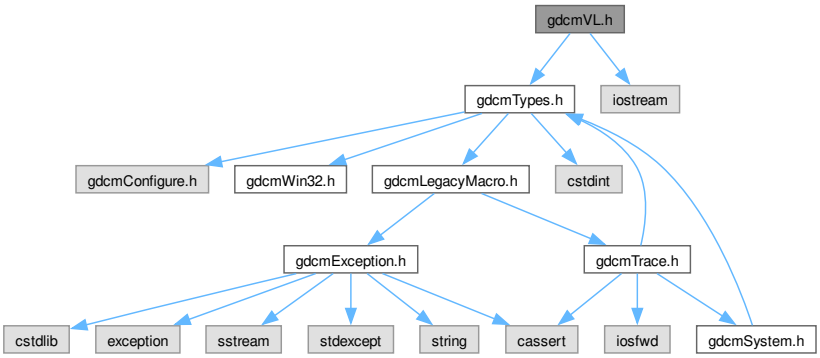
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMVALUEIO_H
#define GDCMVALUEIO_H

#include "gdcmTypes.h"

namespace gdcm_ns
{
template <typename TDE, typename TSwap, typename TType=uint8_t>
class /*GDCM_EXPORT*/ ValueIO
{
public:
  static std::istream &Read(std::istream &is, Value& v, bool readvalues);

  static const std::ostream &Write(std::ostream &os, const Value& v);
};

} // end namespace gdcm_ns

#include "gdcmValueIO.txx"

#endif //GDCMVALUEIO_H
```

13.189 gdcmVL.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmVL.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::VL

  Value Length.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)

13.190 gdcmVL.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
```

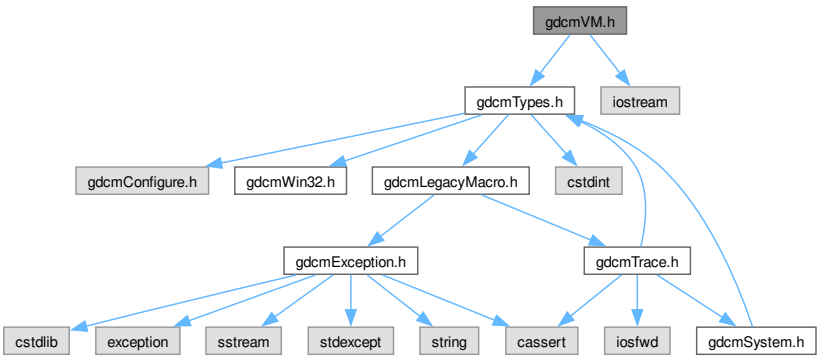
```
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMVL_H
#define GDCMVL_H

#include "gdcmTypes.h"

#include <iostream>

namespace gdcm
{

class GDCM_EXPORT VL
{
public:
  typedef uint32_t Type;
  VL(uint32_t vl = 0) : ValueLength(vl) { }

  // FIXME: ugly
  static uint32_t GetVL32Max() { return 0xFFFFFFFF; }
  static uint16_t GetVL16Max() { return 0xFFFF; }

  bool IsUndefined() const {
    return ValueLength == 0xFFFFFFFF;
  }
  void SetToUndefined() {
    ValueLength = 0xFFFFFFFF;
  }

  bool IsOdd() const {
    return !IsUndefined() && ValueLength % 2;
  }

  VL& operator+=(VL const &vl) {
    ValueLength += vl.ValueLength;
    return *this;
  }
  VL& operator++() {
    ++ValueLength;
    return *this;
  }
  VL operator++(int) {
    uint32_t tmp(ValueLength);
    ++ValueLength;
    return tmp;
  }

  operator uint32_t () const { return ValueLength; }

  VL GetLength() const {
    // VL cannot know it's length...well in implicit yes...
    // TODO: need to check we cannot call this function from an Explicit element
    return 4;
  }

  friend std::ostream& operator«(std::ostream& os, const VL& vl);

  // PURPOSELY not implemented (could not differentiate 16bits vs 32bits VL)
  //friend std::istream& operator»(std::istream& is, VL& n);

  template <typename TSwap>
  std::istream &Read(std::istream &is)
    {
    is.read((char*)(&ValueLength), sizeof(uint32_t));
    TSwap::SwapArray(&ValueLength,1);
    return is;
    }

  template <typename TSwap>
  std::istream &Read16(std::istream &is)
    {
    uint16_t copy;
    is.read((char*)(&copy), sizeof(uint16_t));
    TSwap::SwapArray(&copy,1);
    ValueLength = copy;
    assert( ValueLength <=  65535 /*UINT16_MAX*/ ); // ?? doh !
```

```
    return is;
    }

  template <typename TSwap>
  const std::ostream &Write(std::ostream &os) const
    {
    uint32_t copy = ValueLength;
    if( IsOdd() )
      {
      ++copy;
      }
    TSwap::SwapArray(&copy,1);
    return os.write((char*)(&copy), sizeof(uint32_t));
    }

  template <typename TSwap>
  const std::ostream &Write16(std::ostream &os) const
    {
    assert( ValueLength <=   65535 /*UINT16_MAX*/ );
    uint16_t copy = (uint16_t)ValueLength;
    if( IsOdd() )
      {
      ++copy;
      }
    TSwap::SwapArray(&copy,1);
    return os.write((char*)(&copy), sizeof(uint16_t));
    }

private:
  uint32_t ValueLength;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& os, const VL& val)
{
  os « /*std::hex «*/ val.ValueLength;
  return os;
}

} // end namespace gdcm

#endif //GDCMVL_H
```

13.191 gdcmVM.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmVM.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::VM

  Value Multiplicity Looking at the DICOMV3 dict only there is very few cases: 1 2 3 4 5 6 8 16 24 1-2 1-3 1-8 1-32 1-99 1-n 2-2n 2-n 3-3n 3-n.

Namespaces

- namespace gdcm

Macros

- #define TYPETOLENGTH(type, length)

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VM &_val)

13.191.1 Macro Definition Documentation

13.191.1.1 TYPETOLENGTH

```
#define TYPETOLENGTH(
            type,
            length)
```

Value:

```
template<> struct VMToLength<VM::type> \
{ enum { Length = length }; };
```

13.192 gdcmVM.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMVM_H
#define GDCMVM_H

#include "gdcmTypes.h"
#include <iostream>

namespace gdcm
{

class GDCM_EXPORT VM
{
public:
  typedef enum {
    VM0 = 0, // aka the invalid VM
    VM1 = 1,
    VM2 = 2,
    VM3 = 4,
    VM4 = 8,
    VM5 = 16,
    VM6 = 32,
    VM8 = 64,
    VM9 = 128,
    VM10 = 256,
    VM12 = 512, //1024,
    VM16 = 1024, //2048,
    VM18 = 2048, //4096,
    VM24 = 4096, //8192,
    VM28 = 8192, //16384,
    VM32 = 16384, //32768,
    VM35 = 32768, //65536,
    VM99 = 65536, //131072,
    VM256 = 131072, //262144,
    VM1_2  = VM1 | VM2,
    VM1_3  = VM1 | VM2 | VM3,
    VM1_4  = VM1 | VM2 | VM3 | VM4,
    VM1_5  = VM1 | VM2 | VM3 | VM4 | VM5,
    VM1_8  = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8,
// The following need some work:
    VM1_32 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32,
    VM1_99 = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99,
    VM1_n  = VM1 | VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
    VM2_2n =       VM2       | VM4       | VM6 | VM8       | VM16 | VM24 | VM32        | VM256,
    VM2_n  =       VM2 | VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
    VM3_4  =             VM3 | VM4,
    VM3_3n =             VM3 |             VM6       | VM9        | VM24        | VM99 | VM256,
    VM3_n  =             VM3 | VM4 | VM5 | VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
    VM4_4n =                   VM4                           | VM16 | VM24 | VM32        | VM256,
    VM6_6n =                                 VM6             | VM12 | VM18 | VM24               ,
    VM6_n  =                                 VM6 | VM8 | VM9 | VM16 | VM24 | VM32 | VM99 | VM256,
    VM7_7n,
    VM30_30n,
    VM47_47n,
    VM_END = VM1_n + 1  // Custom tag to count number of entry
  } VMType;

  static const char* GetVMString(VMType vm);
  static VMType GetVMType(const char *vm);

  static bool IsValid(int vm1, VMType vm2);
  //bool IsValid() { return VMField != VM0 && VMField < VM_END; }

  bool Compatible(VM const &vm) const;
```

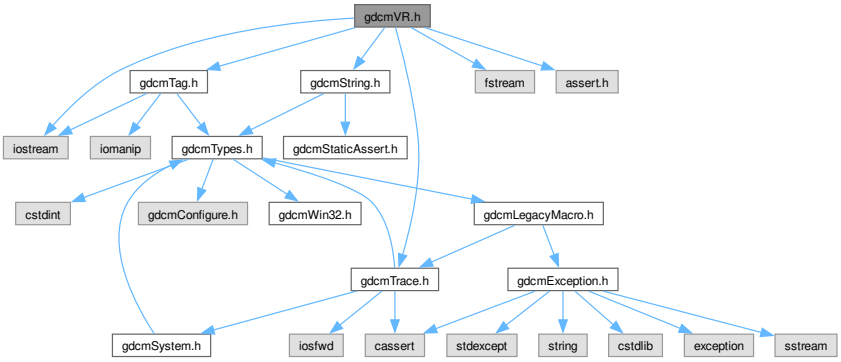
```
  static VMType GetVMTypeFromLength(size_t length, unsigned int size);
  static size_t GetNumberOfElementsFromArray(const char *array, size_t length);

  VM(VMType type = VM0):VMField(type) {}
  operator VMType () const { return VMField; }
  unsigned int GetLength() const;

  friend std::ostream &operator«(std::ostream &os, const VM &vm);
protected:
  static unsigned int GetIndex(VMType vm);

private:
  VMType VMField;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const VM &_val)
{
  assert( VM::GetVMString(_val) );
  _os « VM::GetVMString(_val);
  return _os;
}

//template <int TVM> struct LengthToVM;
//template <> struct LengthToVM<1>
//{ enum { TVM = VM::VM1 }; };

template<int T> struct VMToLength;
#define TYPETOLENGTH(type,length) \
  template<> struct VMToLength<VM::type> \
  { enum { Length = length }; };
// TODO: Could be generated from XML file
//TYPETOLENGTH(VM0,1)
TYPETOLENGTH(VM1,1)
TYPETOLENGTH(VM2,2)
TYPETOLENGTH(VM3,3)
TYPETOLENGTH(VM4,4)
TYPETOLENGTH(VM5,5)
TYPETOLENGTH(VM6,6)
TYPETOLENGTH(VM8,8)
TYPETOLENGTH(VM9,9)
TYPETOLENGTH(VM10,10)
TYPETOLENGTH(VM12,12)
TYPETOLENGTH(VM16,16)
TYPETOLENGTH(VM18,18)
TYPETOLENGTH(VM24,24)
TYPETOLENGTH(VM28,28)
TYPETOLENGTH(VM32,32)
TYPETOLENGTH(VM35,35)
TYPETOLENGTH(VM99,99)
TYPETOLENGTH(VM256,256)
//TYPETOLENGTH(VM1_2,2)
//TYPETOLENGTH(VM1_3,3)
//TYPETOLENGTH(VM1_8,8)
//TYPETOLENGTH(VM1_32,32)
//TYPETOLENGTH(VM1_99,99)
//TYPETOLENGTH(VM1_n,
//TYPETOLENGTH(VM2_2n,
//TYPETOLENGTH(VM2_n,
//TYPETOLENGTH(VM3_3n,
//TYPETOLENGTH(VM3_n,

} // end namespace gdcm

#endif //GDCMVM_H
```

13.193 gdcmVR.h File Reference

```
#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmString.h"
#include <iostream>
#include <fstream>
```

```
#include <assert.h>
```

Include dependency graph for gdcmVR.h:



This graph shows which files directly or indirectly include this file:

Classes

- struct gdcm::UI
- class gdcm::VR

  VR class.

Namespaces

- namespace gdcm

Macros

- #define TYPETOENCODING(type, rep, rtype)
- #define VRTypeTemplateCase(type)

Typedefs

- typedef String<'\\', 16 > gdcm::AEComp
- typedef String<'\\', 64 > gdcm::ASComp
- typedef String<'\\', 16 > gdcm::CSComp
- typedef String<'\\', 64 > gdcm::DAComp
- typedef String<'\\', 64 > gdcm::DTComp
- typedef String<'\\', 64 > gdcm::LOComp
- typedef String<'\\', 64 > gdcm::LTComp
- typedef String<'\\', 64 > gdcm::PNComp
- typedef String<'\\', 64 > gdcm::SHComp
- typedef String<'\\', 64 > gdcm::STComp
- typedef String<'\\', 16 > gdcm::TMComp
- typedef String<'\\', 4294967294 > gdcm::UCComp
- typedef String<'\\', 64, 0 > gdcm::UIComp
- typedef String<'\\', 4294967294 > gdcm::URComp
- typedef String<'\\', 64 > gdcm::UTComp

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const UI &_val)
- std::ostream & gdcm::operator<< (std::ostream &_os, const VR &val)

13.193.1 Macro Definition Documentation

13.193.1.1 TYPETOENCODING

```
#define TYPETOENCODING(
            type,
            rep,
            rtype)
```

Value:

```
template<> struct VRToEncoding<VR::type>    \
{ enum:long long { Mode = VR::rep }; };                \
template<> struct VRToType<VR::type>        \
{ typedef rtype Type; };
```

13.193.1.2 VRTypeTemplateCase

```
#define VRTypeTemplateCase(
            type)
```

Value:

```
case VR::type: \
  return sizeof ( VRToType<VR::type>::Type );
```

Referenced by gdcm::VR::GetSize().

13.194 gdcmVR.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMVR_H
#define GDCMVR_H

#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmString.h"

#include <iostream>
#include <fstream>
#include <assert.h>

//these defines are here to ensure compilation on sunos gcc
#if defined (CS)
# undef CS
#endif
#if defined (DS)
# undef DS
#endif
#if defined (SS)
# undef SS
#endif


namespace gdcm
{

class GDCM_EXPORT VR
{
public:
  enum VRType : long long {
    // Warning: Do not write if ( vr & VR::INVALID ) but if ( vr == VR::INVALID )
    INVALID = 0, // For Item/(Seq) Item Delimitation Item
    AE = 1,
    AS = 2,
    AT = 4,
    CS = 8,
    DA = 16,
    DS = 32,
    DT = 64,
    FD = 128,
    FL = 256,
    IS = 512,
    LO = 1024,
    LT = 2048,
    OB = 4096,
    OD = 134217728, // 2^27
    OF = 8192,
    OL = 268435456, // 2^28
    OV = 2147483648, // 2^31
    OW = 16384,
    PN = 32768,
    SH = 65536,
    SL = 131072,
    SQ = 262144,
    SS = 524288,
    ST = 1048576,
    SV = 4294967296, // 2^32
    TM = 2097152,
    UC = 536870912, // 2^29
    UI = 4194304,
    UL = 8388608,
    UN = 16777216,
```

```
    UR = 1073741824, // 2^30
    US = 33554432,
    UT = 67108864,
    UV = 8589934592, // 2^33
    OB_OW = OB | OW,
    US_SS = US | SS,
    US_SS_OW = US | SS | OW,
    US_OW = US | OW,
    // The following do not have a VRString equivalent (ie cannot be found in PS 3.6)
    VL16 = AE | AS | AT | CS | DA | DS | DT | FD | FL | IS | LO | LT | PN | SH | SL | SS | ST | TM | UI | UL | US, // if( VR & VL16 ) => VR has its VL coded over 16bits
    VL32 = OB | OW | OD | OF | OL | OV | SQ | SV | UC | UN | UR | UT | UV, // if( VR & VL32 ) => VR has its VL coded over 32bits
    VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UC | UI | UR | UT,
    VRBINARY = AT | FL | FD | OB | OD | OF | OL | OV | OW | SL | SQ | SS | SV | UL | UN | US | UV, // FIXME: UN ?
    // PS 3.5:
    // Data Elements with a VR of SQ, OD, OF, OL, OW, OB or UN shall always have a Value Multiplicity of one.
    // GDCM is adding a couple more: AS, LT, ST, UT
    VR_VM1 = AS | LT | ST | UT | SQ | OF | OL | OV | OD | OW | OB | UN, // All those VR have a VM1
    VRALL = VRASCII | VRBINARY,
    VR_END = UV+1  // Invalid VR, need to be max(VRType)+1
  };

  static const char *GetVRString(VRType vr);

  // This function will only look at the very first two chars nothing else
  static VRType GetVRTypeFromFile(const char *vr);

  // You need to make sure end of string is \0
  static VRType GetVRType(const char *vr);
  static const char *GetVRStringFromFile(VRType vr);

  static bool IsValid(const char *vr);
  // Check if vr1 is valid against vr2,
  // Typically vr1 is read from the file and vr2 is taken from the dict
  static bool IsValid(const char *vr1, VRType vr2);
  //static bool IsValid(const VRType &vr1, const VRType &vr2);
  // Find out if the string read is byte swapped
  static bool IsSwap(const char *vr);

  // Size read on disk
  // FIXME: int ?
  int GetLength() const {
    return VR::GetLength(VRField);
  }
  unsigned int GetSizeof() const;
  static uint32_t GetLength(VRType vr) {
    //if( vr == VR::INVALID ) return 4;
    if( vr & VL32 )
      {
      return 4;
      }
    else
      return 2;
  }

  // Some use of template metaprograming with ugly macro
  static bool IsBinary(VRType vr);
  static bool IsASCII(VRType vr);
  // TODO: REMOVE ME
  static bool CanDisplay(VRType vr);
  // TODO: REMOVE ME
  static bool IsBinary2(VRType vr);
  // TODO: REMOVE ME
  static bool IsASCII2(VRType vr);

  VR(VRType vr = INVALID):VRField(vr) { }
  //VR(VR const &vr):VRField(vr.VRField) { }
  std::istream &Read(std::istream &is)
    {
    char vr[2];
    is.read(vr, 2);
    VRField = GetVRTypeFromFile(vr);
    assert( VRField != VR::VR_END );
    if( VRField == VR::INVALID )
    {
      // \0\2 Data/TheralysGDCM120Bug.dcm
      // \0\0 Data/MR_Philips_Intera_PrivateSequenceExplicitVR_in_SQ_2001_e05f_item_wrong_lgt_use_NOSHADOWSEQ.dcm
```

```
      // \0\4 Data/BugGDCM2_UndefItemWrongVL.dcm
      // \44\0 Data/gdcm-MR-PHILIPS-16-Multi-Seq.dcm
      // \0\20 Data/ExplicitVRforPublicElementsImplicitVRforShadowElements.dcm
      // \0\3 Data/DMCPACS_ExplicitImplicit_BogusIOP.dcm
      // \0\4 Data/THERALYS-12-MONO2-Uncompressed-Even_Length_Tag.dcm
      // \0\4 Data/PrivateGEImplicitVRBigEndianTransferSyntax16Bits.dcm
      // \0\4 Data/GE_DLX-8-MONO2-PrivateSyntax.dcm
      throw Exception( "INVALID VR" );
    }
    if( VRField & VL32 )
      {
#if 0
      // For some reason this seems slower on my linux box...
      is.seekg(2, std::ios::cur );
#else
      char dumb[2];
      is.read(dumb, 2);
      if( !(dumb[0] == 0 && dumb[1] == 0 ))
        {
        // JDDICOM_Sample4.dcm
        gdcmDebugMacro( "32bits VR contains non zero bytes. Skipped" );
        }
#endif
      }
    return is;
    }

  const std::ostream &Write(std::ostream &os) const
    {
    VRType vrfield = VRField;
    gdcmAssertAlwaysMacro( !IsDual() );
    if( vrfield == VR::INVALID )
      {
      //vrfield = VR::UN;
      }
    const char *vr = GetVRString(vrfield);
    //assert( strlen( vr ) == 2 );
    assert( vr[0] && vr[1] && vr[2] == 0 );
    os.write(vr, 2);
    // See PS 3.5, Data Element Structure With Explicit VR
    if( vrfield & VL32 )
      {
      const char dumb[2] = {0, 0};
      os.write(dumb,2);
      }
    return os;
    }
  friend std::ostream &operator«(std::ostream &os, const VR &vr);

  operator VRType () const { return VRField; }

  unsigned int GetSize() const;

  bool Compatible(VR const &vr) const;

  bool IsVRFile() const;

  bool IsDual() const;

private:
  // Internal function that map a VRType to an index in the VRStrings table
  static unsigned int GetIndex(VRType vr);
  VRType VRField;
};
//-----------------------------------------------------------------------------
inline std::ostream &operator«(std::ostream &_os, const VR &val)
{
  //_os « VR::GetVRStringFromFile(val.VRField);
  _os « VR::GetVRString(val.VRField);
  return _os;
}

// Apparently SWIG is not happy with something, somewhere below...
#ifndef SWIG

// Tells whether VR Type is ASCII or Binary
template<long long T> struct VRToEncoding;
// Convert from VR Type to real underlying type
template<long long T> struct VRToType;
#define TYPETOENCODING(type,rep, rtype)          \
  template<> struct VRToEncoding<VR::type>    \
```

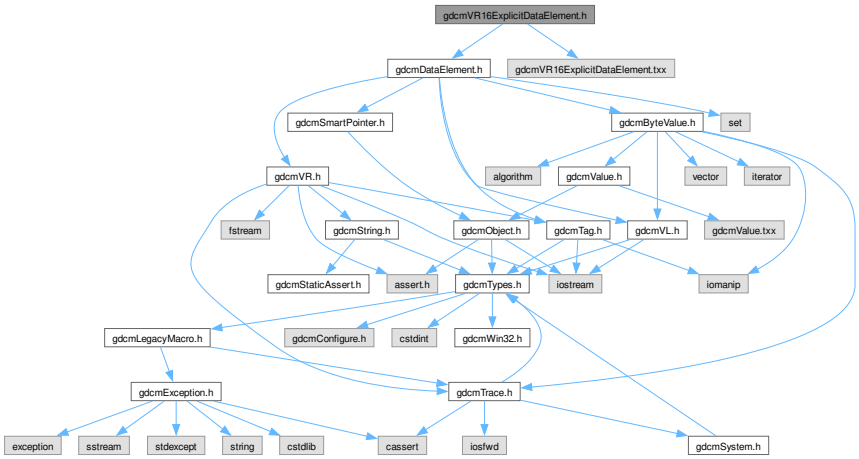
```
  { enum:long long { Mode = VR::rep }; };                    \
  template<> struct VRToType<VR::type>        \
  { typedef rtype Type; };


// Do not use me
struct UI { char Internal[64+1];
  friend std::ostream& operator«(std::ostream &_os, const UI &_val);
};
inline std::ostream& operator«(std::ostream &_os, const UI &_val)
{
  _os « _val.Internal;
  return _os;
}

typedef String<'\\',16> AEComp;
typedef String<'\\',64> ASComp;
typedef String<'\\',16> CSComp;
typedef String<'\\',64> DAComp;
typedef String<'\\',64> DTComp;
typedef String<'\\',64> LOComp;
typedef String<'\\',64> LTComp;
typedef String<'\\',64> PNComp;
typedef String<'\\',64> SHComp;
typedef String<'\\',64> STComp;
typedef String<'\\',4294967294> UCComp;
typedef String<'\\',4294967294> URComp;
typedef String<'\\',16> TMComp;
typedef String<'\\',64,0> UIComp;
typedef String<'\\',64> UTComp;


// TODO: Could be generated from XML file
TYPETOENCODING(AE,VRASCII ,AEComp)
TYPETOENCODING(AS,VRASCII ,ASComp)
TYPETOENCODING(AT,VRBINARY,Tag)
TYPETOENCODING(CS,VRASCII ,CSComp)
TYPETOENCODING(DA,VRASCII ,DAComp)
TYPETOENCODING(DS,VRASCII ,double)
TYPETOENCODING(DT,VRASCII ,DTComp)
TYPETOENCODING(FL,VRBINARY,float)
TYPETOENCODING(FD,VRBINARY,double)
TYPETOENCODING(IS,VRASCII ,int32_t)
TYPETOENCODING(LO,VRASCII ,LOComp)
TYPETOENCODING(LT,VRASCII ,LTComp)
TYPETOENCODING(OB,VRBINARY,uint8_t)
TYPETOENCODING(OD,VRBINARY,double)
TYPETOENCODING(OF,VRBINARY,float)
TYPETOENCODING(OL,VRBINARY,uint32_t)
TYPETOENCODING(OV,VRBINARY,uint64_t)
TYPETOENCODING(OW,VRBINARY,uint16_t)
TYPETOENCODING(PN,VRASCII ,PNComp)
TYPETOENCODING(SH,VRASCII ,SHComp)
TYPETOENCODING(SL,VRBINARY,int32_t)
TYPETOENCODING(SQ,VRBINARY,unsigned char) // FIXME
TYPETOENCODING(SS,VRBINARY,int16_t)
TYPETOENCODING(ST,VRASCII ,STComp)
TYPETOENCODING(SV,VRBINARY,int64_t)
TYPETOENCODING(TM,VRASCII ,TMComp)
TYPETOENCODING(UC,VRASCII ,UCComp)
TYPETOENCODING(UI,VRASCII ,UIComp)
TYPETOENCODING(UL,VRBINARY,uint32_t)
TYPETOENCODING(UN,VRBINARY,uint8_t) // FIXME ?
TYPETOENCODING(UR,VRASCII,URComp)
TYPETOENCODING(US,VRBINARY,uint16_t)
TYPETOENCODING(UT,VRASCII ,UTComp)
TYPETOENCODING(UV,VRBINARY,uint64_t)

#define VRTypeTemplateCase(type) \
  case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );

inline unsigned int VR::GetSize() const
{
  switch(VRField)
  {
    VRTypeTemplateCase(AE)
    VRTypeTemplateCase(AS)
    VRTypeTemplateCase(AT)
    VRTypeTemplateCase(CS)
    VRTypeTemplateCase(DA)
```

```
    VRTypeTemplateCase(DS)
    VRTypeTemplateCase(DT)
    VRTypeTemplateCase(FL)
    VRTypeTemplateCase(FD)
    VRTypeTemplateCase(IS)
    VRTypeTemplateCase(LO)
    VRTypeTemplateCase(LT)
    VRTypeTemplateCase(OB)
    VRTypeTemplateCase(OD)
    VRTypeTemplateCase(OF)
    VRTypeTemplateCase(OL)
    VRTypeTemplateCase(OV)
    VRTypeTemplateCase(OW)
    VRTypeTemplateCase(PN)
    VRTypeTemplateCase(SH)
    VRTypeTemplateCase(SL)
    VRTypeTemplateCase(SQ)
    VRTypeTemplateCase(SS)
    VRTypeTemplateCase(ST)
    VRTypeTemplateCase(SV)
    VRTypeTemplateCase(TM)
    VRTypeTemplateCase(UC)
    VRTypeTemplateCase(UI)
    VRTypeTemplateCase(UL)
    VRTypeTemplateCase(UN)
    VRTypeTemplateCase(UR)
    VRTypeTemplateCase(US)
    VRTypeTemplateCase(UT)
    VRTypeTemplateCase(UV)
    case VR::US_SS:
      return 2;

    case VR::INVALID:
    case VR::OB_OW:
    case VR::US_SS_OW:
    case VR::US_OW:
    case VR::VL16:
    case VR::VL32:
    case VR::VRASCII:
    case VR::VRBINARY:
    case VR::VR_VM1:
    case VR::VRALL:
    case VR::VR_END:
    default:
       assert( 0 && "should not" );
  }
  return 0;
}
#endif // SWIG


} // end namespace gdcm

#endif //GDCMVR_H
```

13.195 gdcmVR16ExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmVR16ExplicitDataElement.txx"
```

Include dependency graph for gdcmVR16ExplicitDataElement.h:



Classes

- class gdcm::VR16ExplicitDataElement

  Class to read/write a DataElement as Explicit Data Element.

Namespaces

- namespace gdcm

13.196 gdcmVR16ExplicitDataElement.h

Go to the documentation of this file.

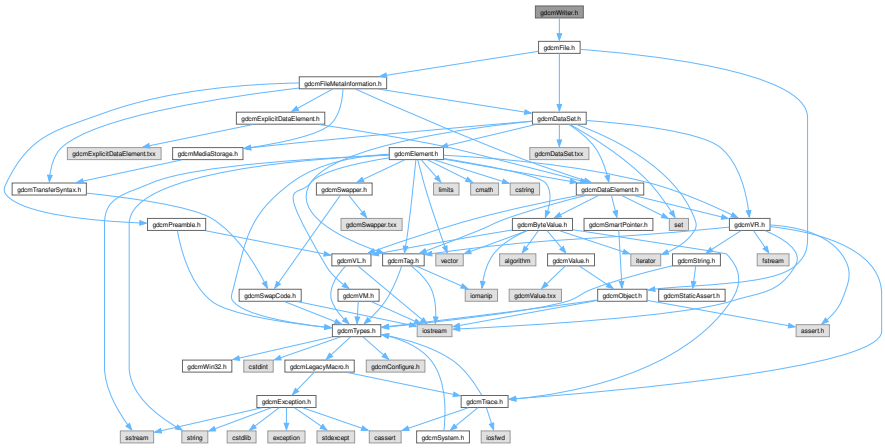
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMVR16EXPLICITDATAELEMENT_H
#define GDCMVR16EXPLICITDATAELEMENT_H

#include "gdcmDataElement.h"

namespace gdcm
{
// Data Element (Explicit)
class GDCM_EXPORT VR16ExplicitDataElement : public DataElement
{
public:
```

```
  VL GetLength() const;

  template <typename TSwap>
  std::istream &Read(std::istream &is);

  template <typename TSwap>
  std::istream &ReadPreValue(std::istream &is);

  template <typename TSwap>
  std::istream &ReadValue(std::istream &is, bool readvalues = true);

  template <typename TSwap>
  std::istream &ReadWithLength(std::istream &is, VL & length);

  // PURPOSELY do not provide an implementation for writing !
  //template <typename TSwap>
  //const std::ostream &Write(std::ostream &os) const;
};

} // end namespace gdcm

#include "gdcmVR16ExplicitDataElement.txx"

#endif //GDCMVR16EXPLICITDATAELEMENT_H
```

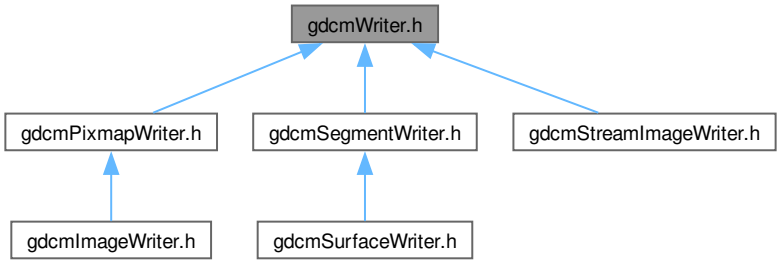
13.197 gdcmWriter.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Writer

  Writer ala DOM (Document Object Model).

Namespaces

- namespace gdcm

13.198 gdcmWriter.h

Go to the documentation of this file.

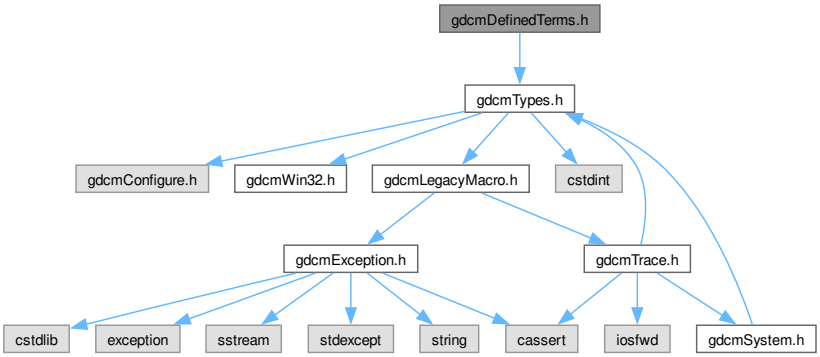
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMWRITER_H
#define GDCMWRITER_H

#include "gdcmFile.h"

namespace gdcm
{

class FileMetaInformation;
class GDCM_EXPORT Writer
{
public:
  Writer();
  virtual ~Writer();

```

```
  virtual bool Write(); // Execute()

  void SetFileName(const char *filename_native);

  void SetStream(std::ostream &output_stream) {
    Stream = &output_stream;
  }

  void SetFile(const File& f) { F = f; }
  File &GetFile() { return *F; }

  void SetCheckFileMetaInformation(bool b) { CheckFileMetaInformation = b; }
  void CheckFileMetaInformationOff() { CheckFileMetaInformation = false; }
  void CheckFileMetaInformationOn() { CheckFileMetaInformation = true; }

protected:
  void SetWriteDataSetOnly(bool b) { WriteDataSetOnly = b; }

protected:
  friend class StreamImageWriter;
  //this function is added for the StreamImageWriter, which needs to write
  //up to the pixel data and then stops right before writing the pixel data.
  //after that, for the raw codec at least, zeros are written for the length of the data
  std::ostream* GetStreamPtr() const { return Stream; }

protected:
  std::ostream *Stream;
  std::ofstream *Ofstream;
  bool GetCheckFileMetaInformation() const { return CheckFileMetaInformation; }

private:
  SmartPointer<File> F;
  bool CheckFileMetaInformation;
  bool WriteDataSetOnly;
};

} // end namespace gdcm

#endif //GDCMWRITER_H
```

13.199 gdcmDefinedTerms.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmDefinedTerms.h:



Classes

- class gdcm::DefinedTerms

Defined Terms are used when the specified explicit Values may be extended by implementors to include additional new Values. These new Values shall be specified in the Conformance Statement (see PS 3.2) and shall not have the same meaning as currently defined Values in this standard. A Data Element with Defined Terms that does not contain a Value equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation Type ID (4008,0210) is an example of a Data Element having Defined Terms. It is defined to have a Value that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data Element has Defined Terms other Interpretation Type IDs may be defined by the implementor.

Namespaces

- namespace gdcm

13.200 gdcmDefinedTerms.h

Go to the documentation of this file.

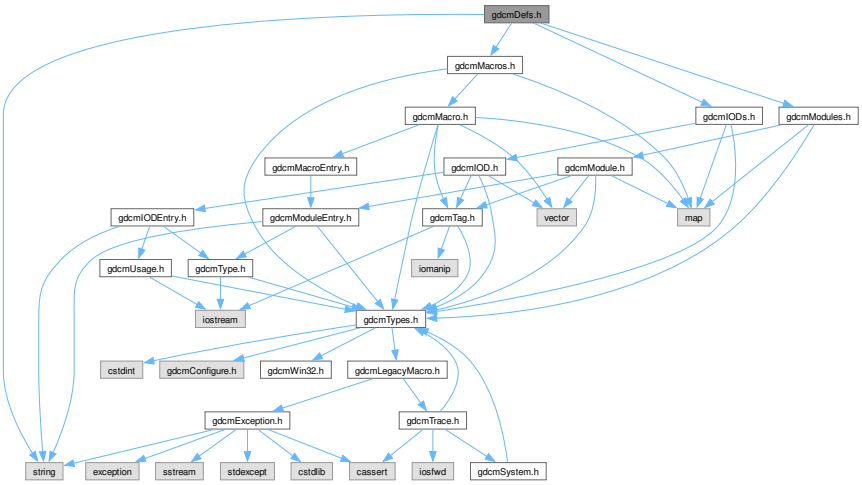
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDEFINEDTERMS_H
#define GDCMDEFINEDTERMS_H

#include "gdcmTypes.h"

namespace gdcm
{
class GDCM_EXPORT DefinedTerms
{
public:
  DefinedTerms() = default;
private:
};

} // end namespace gdcm

#endif //GDCMDEFINEDTERMS_H
```

13.201 gdcmDefs.h File Reference

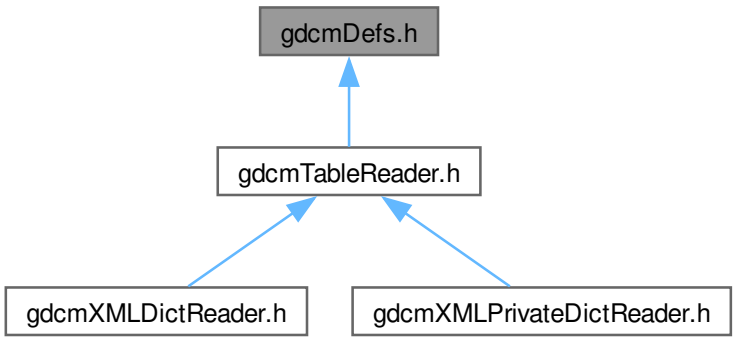
```
#include "gdcmModules.h"
#include "gdcmMacros.h"
#include "gdcmIODs.h"
```

```
#include <string>
```

Include dependency graph for gdcmDefs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Defs

  FIXME I do not like the name 'Defs'.

Namespaces

- namespace gdcm

13.202 gdcmDefs.h

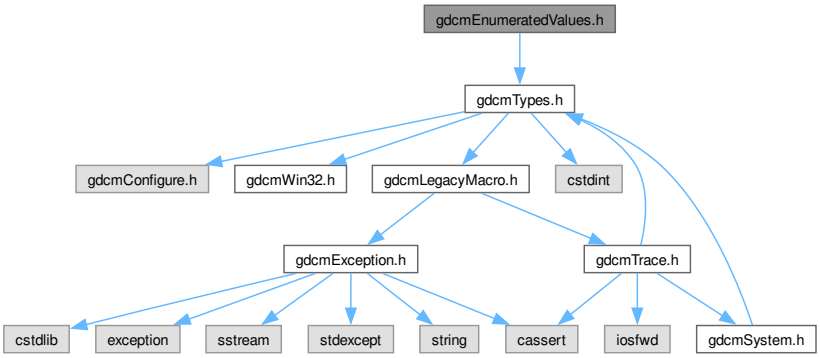
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDEFS_H
#define GDCMDEFS_H

#include "gdcmModules.h"
#include "gdcmMacros.h"
#include "gdcmIODs.h"

#include <string>

namespace gdcm
{
class DataSet;
class File;
class MediaStorage;
class GDCM_EXPORT Defs
{
public:
  Defs();
  ~Defs();
  Defs &operator=(const Defs &val) = delete;
  Defs(const Defs &val) = delete;

  const Modules &GetModules() const { return Part3Modules; }
  Modules &GetModules() { return Part3Modules; }

  const Macros &GetMacros() const { return Part3Macros; }
  Macros &GetMacros() { return Part3Macros; }

  const IODs & GetIODs() const { return Part3IODs; }
  IODs & GetIODs() { return Part3IODs; }

  bool IsEmpty() const { return GetModules().IsEmpty(); }

  bool Verify(const File& file) const;

  // \deprecated DO NOT USE
  bool Verify(const DataSet& ds) const;

  Type GetTypeFromTag(const File& file, const Tag& tag) const;

  static const char *GetIODNameFromMediaStorage(MediaStorage const &ms);

  const IOD& GetIODFromFile(const File& file) const;

protected:
  friend class Global;
  void LoadDefaults();
  void LoadFromFile(const char *filename);

private:
  // Part 3 stuff:
  Macros Part3Macros;
  Modules Part3Modules;
  IODs Part3IODs;

};


} // end namespace gdcm

#endif //GDCMDEFS_H
```

13.203 gdcmEnumeratedValues.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmEnumeratedValues.h:



Classes

- class gdcm::EnumeratedValues

  Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:

Namespaces

- namespace gdcm

13.204 gdcmEnumeratedValues.h

Go to the documentation of this file.

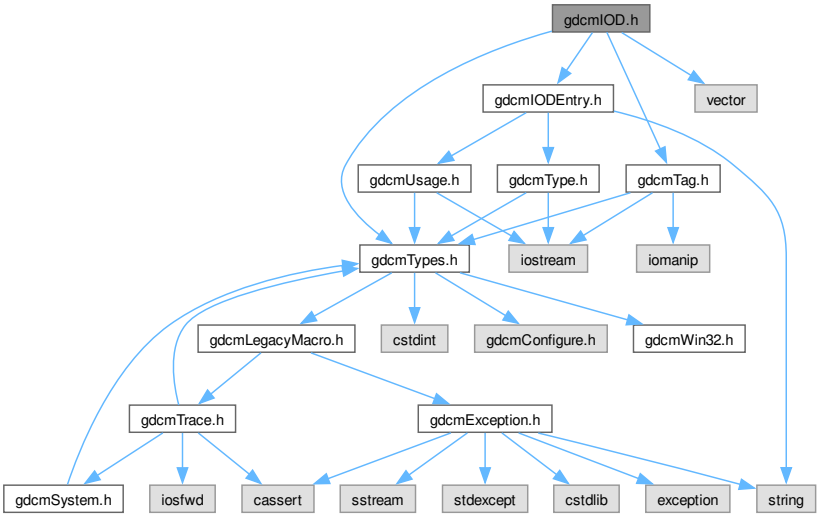
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMENUMERATEDVALUES_H
#define GDCMENUMERATEDVALUES_H

#include "gdcmTypes.h"

namespace gdcm
{
```

```
class GDCM_EXPORT EnumeratedValues
{
public:
  EnumeratedValues() = default;
private:
};

} // end namespace gdcm

#endif //GDCMENUMERATEDVALUES_H
```

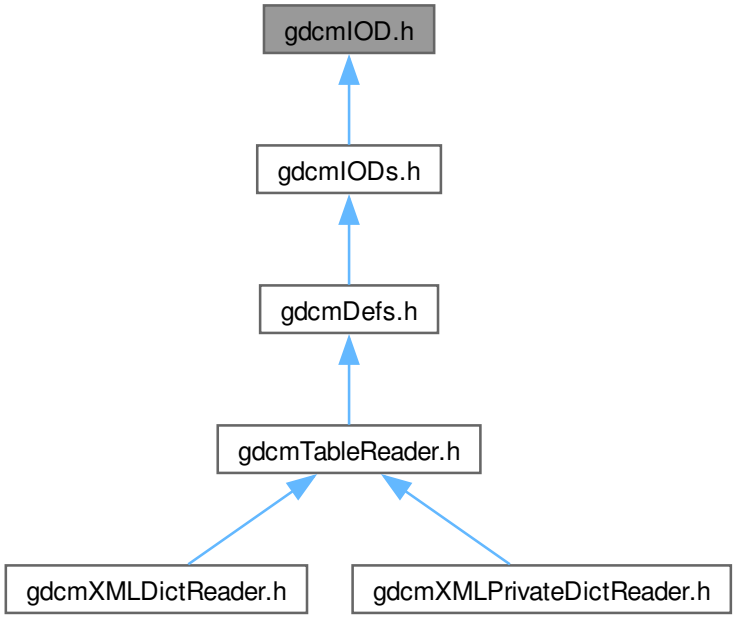
13.205 gdcmIOD.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmIODEntry.h"
#include <vector>
```

Include dependency graph for gdcmIOD.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::IOD

  Class for representing a IOD.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)

13.206 gdcmIOD.h

Go to the documentation of this file.

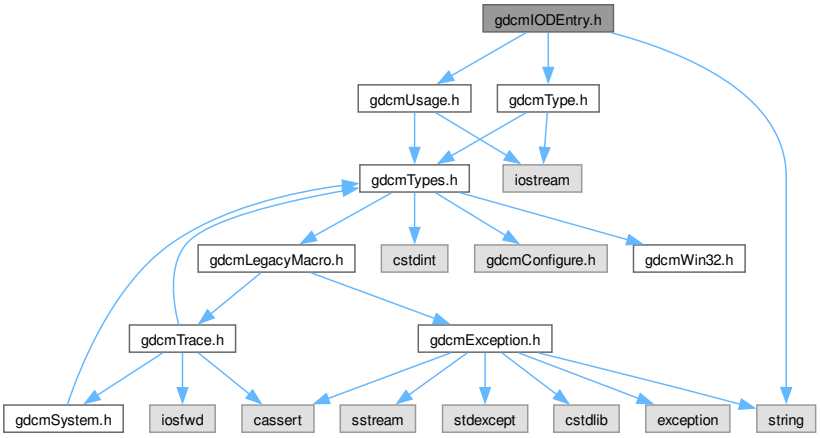
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIOD_H
#define GDCMIOD_H

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmIODEntry.h"

#include <vector>

namespace gdcm
{
class DataSet;
class Defs;

class GDCM_EXPORT IOD
{
public:
  typedef std::vector<IODEntry> MapIODEntry;
  typedef MapIODEntry::size_type SizeType;

  IOD() = default;
  friend std::ostream& operator«(std::ostream& _os, const IOD &_val);

  void Clear() { IODInternal.clear(); }

  void AddIODEntry(const IODEntry & iode)
    {
    IODInternal.push_back(iode);
    }

  SizeType GetNumberOfIODs() const {
    return IODInternal.size();
  }

  const IODEntry& GetIODEntry(SizeType idx) const
    {
    return IODInternal[idx];
    }

  Type GetTypeFromTag(const Defs &defs, const Tag& tag) const;

private:
  //IOD &operator=(const IOD &_val); // purposely not implemented
  //IOD(const IOD &_val); // purposely not implemented

  MapIODEntry IODInternal;
};
//----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const IOD &_val)
{
  IOD::MapIODEntry::const_iterator it = _val.IODInternal.begin();
  for(;it != _val.IODInternal.end(); ++it)
    {
    _os « *it « '\n';
    }

  return _os;
}

} // end namespace gdcm

#endif //GDCMIOD_H
```

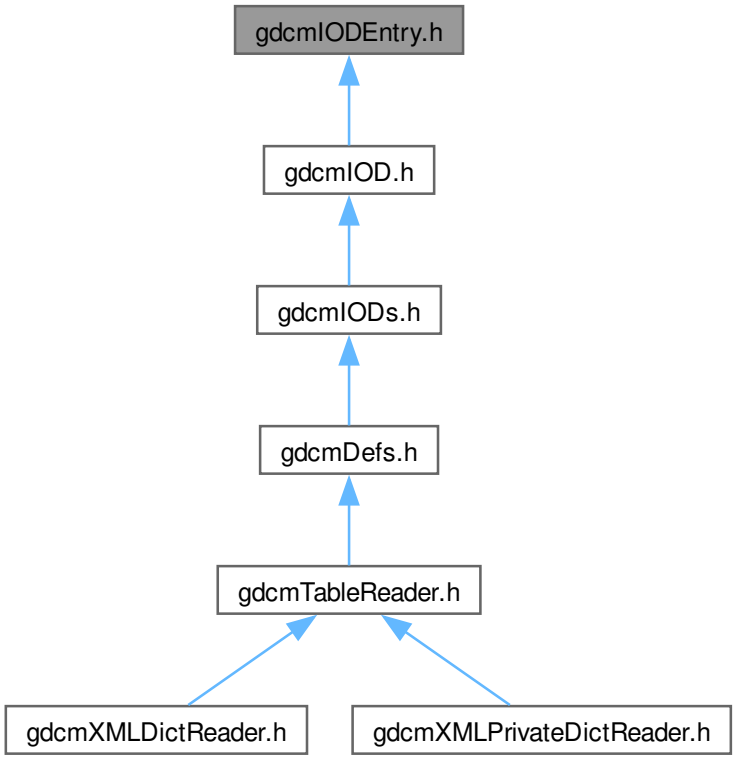
13.207 gdcmIODEntry.h File Reference

```
#include "gdcmUsage.h"
#include "gdcmType.h"
#include <string>
```

Include dependency graph for gdcmIODEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::IODEntry

  Class for representing a IODEntry.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)

13.208 gdcmIODEntry.h

Go to the documentation of this file.

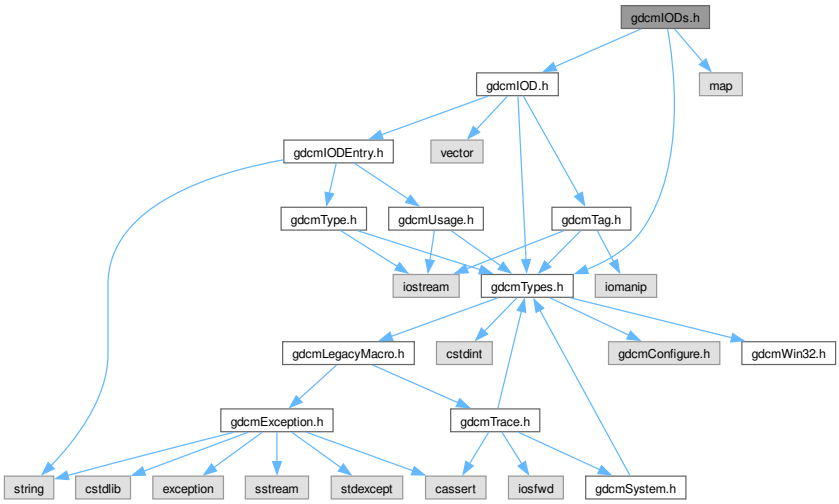
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIODENTRY_H
#define GDCMIODENTRY_H

#include "gdcmUsage.h"
#include "gdcmType.h"

#include <string>

namespace gdcm
{
class GDCM_EXPORT IODEntry
{
public:
  IODEntry(const char *name = "", const char *ref = "", const char *inUsage = ""):Name(name),Ref(ref),usage(inUsage) {
  }
  friend std::ostream& operator«(std::ostream& _os, const IODEntry &_val);

  void SetIE(const char *ie) { IE = ie; }
  const char *GetIE() const { return IE.c_str(); }

  void SetName(const char *name) { Name = name; }
  const char *GetName() const { return Name.c_str(); }

  void SetRef(const char *ref) { Ref = ref; }
  const char *GetRef() const { return Ref.c_str(); }

  void SetUsage(const char *inUsage) { usage = inUsage; }
  const char *GetUsage() const { return usage.c_str(); }
  Usage::UsageType GetUsageType() const;

private:
  std::string IE;

  std::string Name;

  std::string Ref;

  std::string usage;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const IODEntry &_val)
{
  _os « _val.IE « "\t" « _val.Name « "\t" « _val.Ref « "\t" « _val.usage;
  return _os;
}

} // end namespace gdcm

#endif //GDCMIODENTRY_H
```

13.209 gdcmIODs.h File Reference

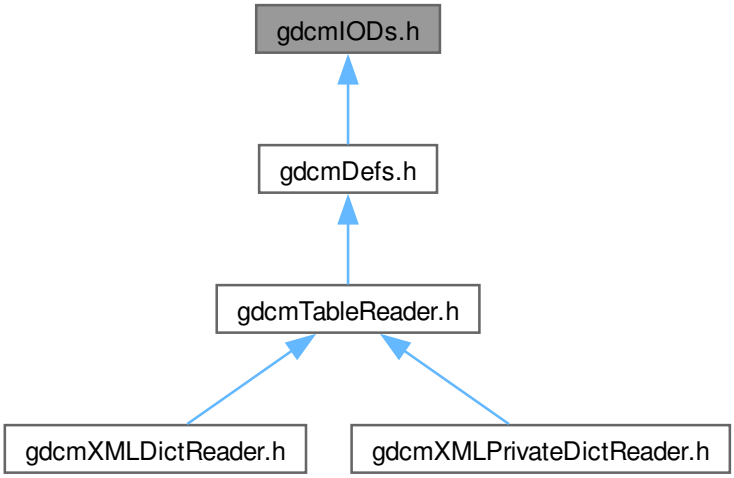
```
#include "gdcmTypes.h"
#include "gdcmIOD.h"
```

```
#include <map>
```

Include dependency graph for gdcmIODs.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::IODs

  Class for representing a IODs.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)

13.210 gdcmIODs.h

Go to the documentation of this file.

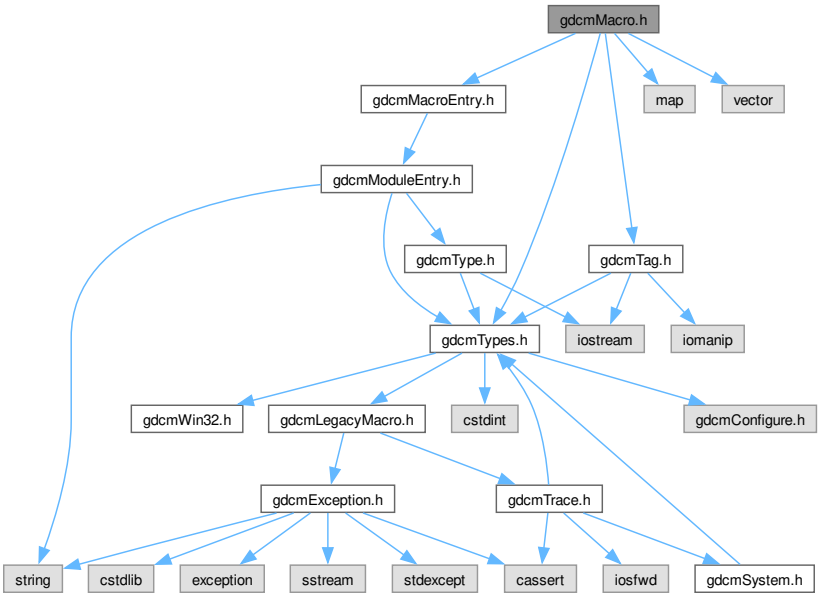
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIODS_H
#define GDCMIODS_H

#include "gdcmTypes.h"
#include "gdcmIOD.h"

#include <map>

namespace gdcm
{
class GDCM_EXPORT IODs
{
public:
  typedef std::string IODName;
  typedef std::map<IODName, IOD> IODMapType;

  IODs() = default;
  friend std::ostream& operator«(std::ostream& _os, const IODs &_val);

  void Clear() { IODsInternal.clear(); }

  void AddIOD(const char *name , const IOD & module )
    {
    IODsInternal.insert(
      IODMapType::value_type(name, module));
    }
  const IOD &GetIOD(const char *name) const
    {
    //return IODsInternal[name];
    IODMapType::const_iterator it = IODsInternal.find( name );
    assert( it != IODsInternal.end() );
    assert( it->first == name );
    return it->second;
    }

  typedef IODMapType::const_iterator IODMapTypeConstIterator;
  IODMapTypeConstIterator Begin() const { return IODsInternal.begin(); }
  IODMapTypeConstIterator End() const { return IODsInternal.end(); }

private:
  IODMapType IODsInternal;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const IODs &_val)
{
  IODs::IODMapType::const_iterator it = _val.IODsInternal.begin();
```

```
  for(;it != _val.IODsInternal.end(); ++it)
    {
    const std::string &name = it->first;
    const IOD &m = it->second;
    _os « name « " " « m « '\n';
    }

  return _os;
}


} // end namespace gdcm

#endif //GDCMIODS_H
```

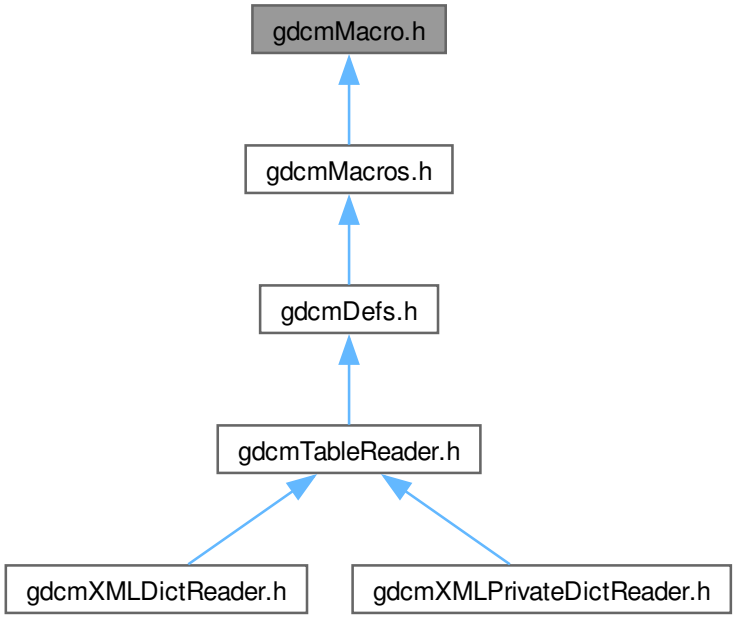
13.211 gdcmMacro.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmMacroEntry.h"
#include <map>
#include <vector>
```

Include dependency graph for gdcmMacro.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Macro

  Class for representing a Macro.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)

13.212 gdcmMacro.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
```

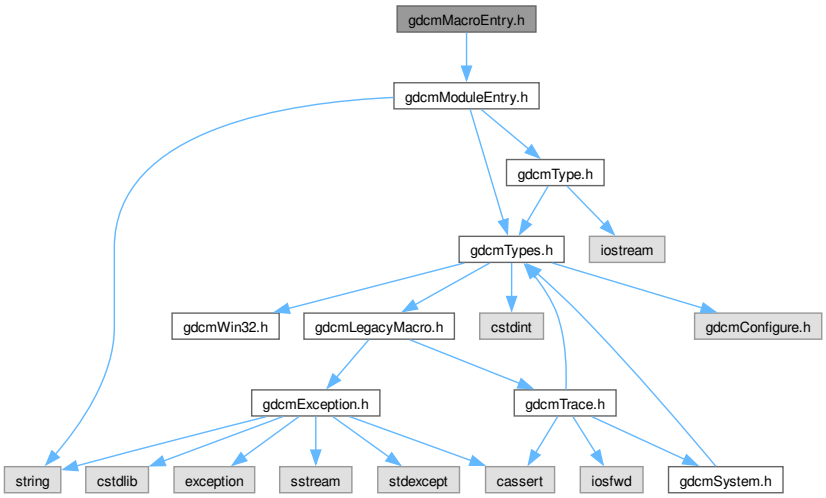
```

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMACRO_H
#define GDCMMACRO_H

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmMacroEntry.h"

#include <map>
#include <vector>

namespace gdcm
{

class DataSet;
class Usage;
class GDCM_EXPORT Macro
{
public:
  typedef std::map<Tag, MacroEntry> MapModuleEntry;
  typedef std::vector<std::string> ArrayIncludeMacrosType;

  //typedef MapModuleEntry::const_iterator ConstIterator;
  //typedef MapModuleEntry::iterator Iterator;
  //ConstIterator Begin() const { return ModuleInternal.begin(); }
  //Iterator Begin() { return ModuleInternal.begin(); }
  //ConstIterator End() const { return ModuleInternal.end(); }
  //Iterator End() { return ModuleInternal.end(); }

  Macro() = default;
  friend std::ostream& operator«(std::ostream& _os, const Macro&_val);

  void Clear() { ModuleInternal.clear(); }

  void AddMacroEntry(const Tag& tag, const MacroEntry & module)
    {
    ModuleInternal.insert(
      MapModuleEntry::value_type(tag, module));
    }

  bool FindMacroEntry(const Tag &tag) const;
  const MacroEntry& GetMacroEntry(const Tag &tag) const;

  void SetName( const char *name) { Name = name; }
  const char *GetName() const { return Name.c_str(); }

  // Verify will print on std::cerr for error
  // Upon success will return true, false otherwise
  bool Verify(const DataSet& ds, Usage const & usage) const;

private:
  //Module &operator=(const Module &_val); // purposely not implemented
  //Module(const Module &_val); // purposely not implemented

  MapModuleEntry ModuleInternal;
  std::string Name;
};
//----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const Macro &_val)
{
  _os « _val.Name « '\n';
  Macro::MapModuleEntry::const_iterator it = _val.ModuleInternal.begin();
  for(;it != _val.ModuleInternal.end(); ++it)
    {
    const Tag &t = it->first;
    const MacroEntry &de = it->second;
    _os « t « " " « de « '\n';
    }

  return _os;
}

```

```
} // end namespace gdcm

#endif //GDCMMACRO_H
```

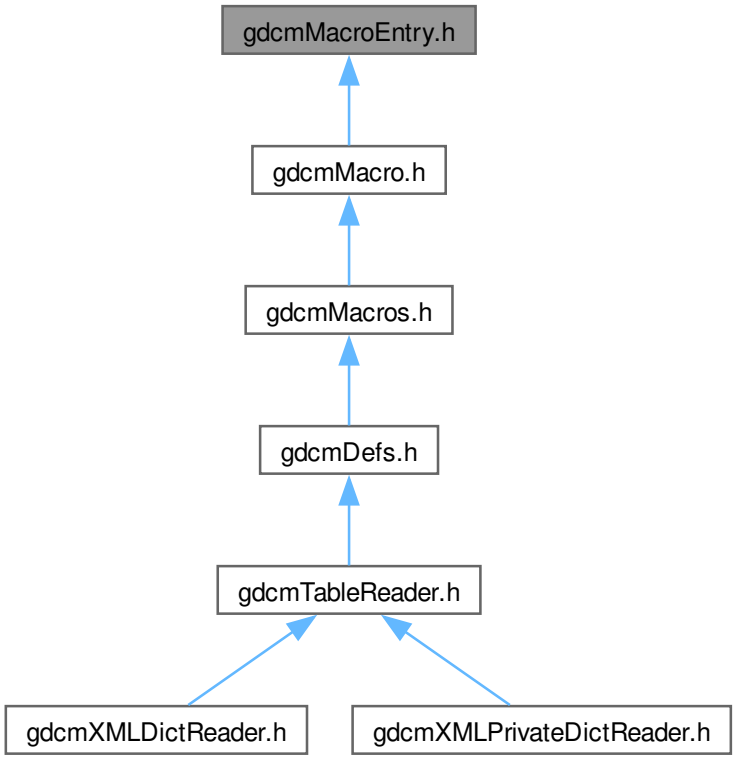
13.213 gdcmMacroEntry.h File Reference

`#include "gdcmModuleEntry.h"`

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define GDCMMACROENTRY_H

13.213.1 Macro Definition Documentation

13.213.1.1 GDCMMACROENTRY_H

```
#define GDCMMACROENTRY_H
```

13.214 gdcmMacroEntry.h

Go to the documentation of this file.

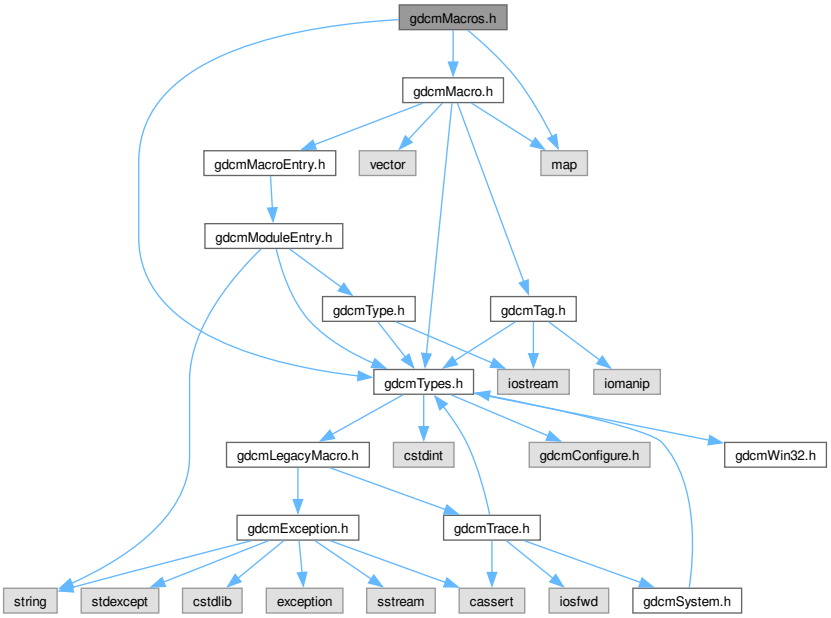
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#if 0
#ifndef GDCMMACROENTRY_H
#define GDCMMACROENTRY_H

#include "gdcmTypes.h"
#include "gdcmType.h"

#include <string>

namespace gdcm
{
class GDCM_EXPORT MacroEntry
{
public:
  MacroEntry(const char *name = "", const char *type = "3", const char *description = ""):Name(name)/*,Type(type)*/,DescriptionField(description) {
    DataElementType = Type::GetTypeType(type);
  }
  virtual ~MacroEntry() {} // important
  friend std::ostream& operator«(std::ostream& _os, const MacroEntry &_val);

  void SetName(const char *name) { Name = name; }
  const char *GetName() const { return Name.c_str(); }

  void SetType(const Type &type) { DataElementType = type; }
  const Type &GetType() const { return DataElementType; }

  /*
   * WARNING: 'Description' is currently a std::string, but it might change in the future
   * do not expect it to remain the same, and always use the ModuleEntry::Description typedef
   * instead.
   */
  typedef std::string Description;
  void SetDescription(const char *d) { DescriptionField = d; }
  const Description & GetDescription() const { return DescriptionField; }

protected:
  // PS 3.3 repeats the name of an attribute, but often contains typos
  // for now we will not use this info, but instead access the DataDict instead
  std::string Name;

  // An attribute, encoded as a Data Element, may or may not be required in a
  // Data Set, depending on that Attribute's Data Element Type.
  Type DataElementType;

  // TODO: for now contains the raw description (with enumerated values, defined terms...)
  Description DescriptionField;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const MacroEntry &_val)
{
  _os « _val.Name « "\t" « _val.DataElementType « "\t" « _val.DescriptionField;
  return _os;
}


} // end namespace gdcm

#endif //GDCMMODULEENTRY_H
#endif

#ifndef GDCMMACROENTRY_H
```

```
#define GDCMMACROENTRY_H
#include "gdcmModuleEntry.h"
#endif
```

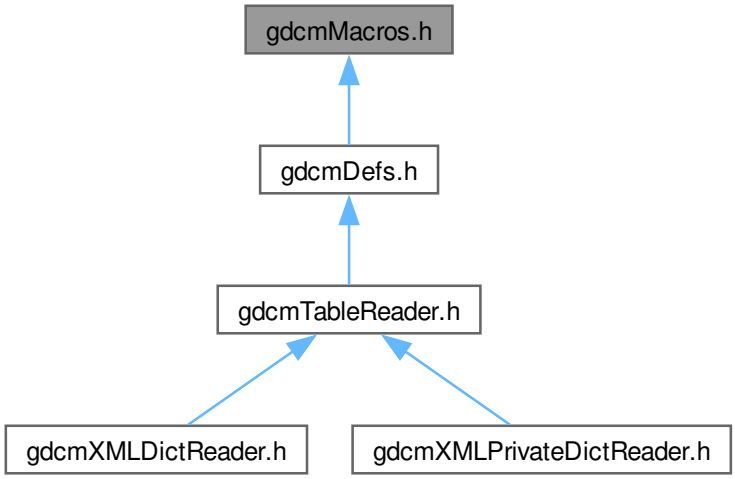
13.215 gdcmMacros.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmMacro.h"
#include <map>
```

Include dependency graph for gdcmMacros.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Macros

  Class for representing a Modules.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)

13.216 gdcmMacros.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
```

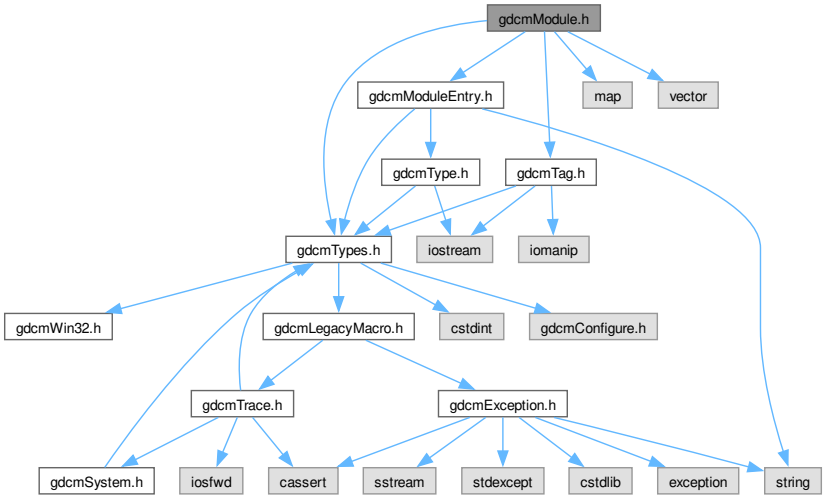
```
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMACROS_H
#define GDCMMACROS_H

#include "gdcmTypes.h"
#include "gdcmMacro.h"

#include <map>

namespace gdcm
{
class GDCM_EXPORT Macros
{
public:
  typedef std::map<std::string, Macro> ModuleMapType;

  Macros() = default;
  friend std::ostream& operator«(std::ostream& _os, const Macros&_val);

  void Clear() { ModulesInternal.clear(); }

  // A Module is inserted based on it's ref
  void AddMacro(const char *ref, const Macro & module )
    {
    assert( ref && *ref );
    assert( ModulesInternal.find( ref ) == ModulesInternal.end() );
    ModulesInternal.insert(
      ModuleMapType::value_type(ref, module));
    }
  const Macro &GetMacro(const char *name) const
    {
    assert( name && *name );
    ModuleMapType::const_iterator it = ModulesInternal.find( name );
    assert( it != ModulesInternal.end() );
    assert( it->first == name );
    return it->second;
    }

  bool IsEmpty() const { return ModulesInternal.empty(); }

private:
  ModuleMapType ModulesInternal;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const Macros &_val)
{
  Macros::ModuleMapType::const_iterator it = _val.ModulesInternal.begin();
  for(;it != _val.ModulesInternal.end(); ++it)
    {
    const std::string &name = it->first;
    const Macro &m = it->second;
    _os « name « " " « m « '\n';
    }

  return _os;
}


} // end namespace gdcm

#endif //GDCMMODULES_H
```

13.217 gdcmModule.h File Reference

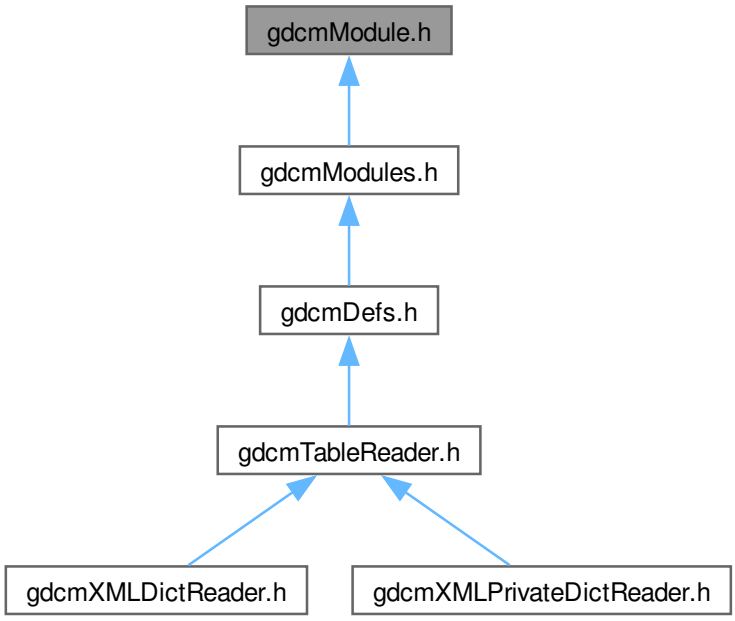
```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmModuleEntry.h"
#include <map>
```

```
#include <vector>
```

Include dependency graph for gdcmModule.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Module

  Class for representing a Module.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)

13.218 gdcmModule.h

Go to the documentation of this file.

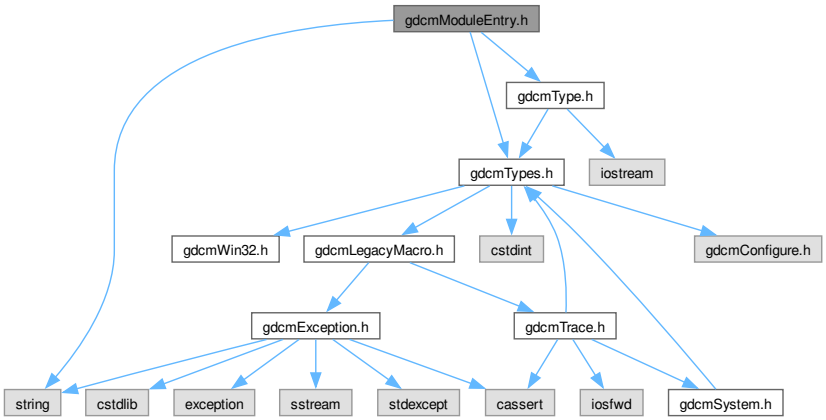
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMODULE_H
#define GDCMMODULE_H

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmModuleEntry.h"

#include <map>
#include <vector>

namespace gdcm
{

class DataSet;
class Usage;
class Macros;
class GDCM_EXPORT Module
{
public:
  typedef std::map<Tag, ModuleEntry> MapModuleEntry;
  typedef std::vector<std::string> ArrayIncludeMacrosType;

  //typedef MapModuleEntry::const_iterator ConstIterator;
  //typedef MapModuleEntry::iterator Iterator;
  //ConstIterator Begin() const { return ModuleInternal.begin(); }
  //Iterator Begin() { return ModuleInternal.begin(); }
  //ConstIterator End() const { return ModuleInternal.end(); }
  //Iterator End() { return ModuleInternal.end(); }

  Module() = default;
  friend std::ostream& operator«(std::ostream& _os, const Module &_val);

  void Clear() { ModuleInternal.clear(); }

  void AddModuleEntry(const Tag& tag, const ModuleEntry & module)
    {
```

```
    ModuleInternal.insert(
      MapModuleEntry::value_type(tag, module));
    }

  void AddMacro(const char *include)
    {
    ArrayIncludeMacros.push_back( include );
    }

  bool FindModuleEntryInMacros(Macros const &macros, const Tag &tag) const;
  const ModuleEntry& GetModuleEntryInMacros(Macros const &macros, const Tag &tag) const;

  void SetName( const char *name) { Name = name; }
  const char *GetName() const { return Name.c_str(); }

  // Verify will print on std::cerr for error
  // Upon success will return true, false otherwise
  bool Verify(const DataSet& ds, Usage const & usage) const;

private:
  //Module &operator=(const Module &_val); // purposely not implemented
  //Module(const Module &_val); // purposely not implemented

  MapModuleEntry ModuleInternal;
  std::string Name;
  ArrayIncludeMacrosType ArrayIncludeMacros;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const Module &_val)
{
  _os « _val.Name « '\n';
  Module::MapModuleEntry::const_iterator it = _val.ModuleInternal.begin();
  for(;it != _val.ModuleInternal.end(); ++it)
    {
    const Tag &t = it->first;
    const ModuleEntry &de = it->second;
    _os « t « " " « de « '\n';
    }

  return _os;
}

} // end namespace gdcm

#endif //GDCMMODULE_H
```

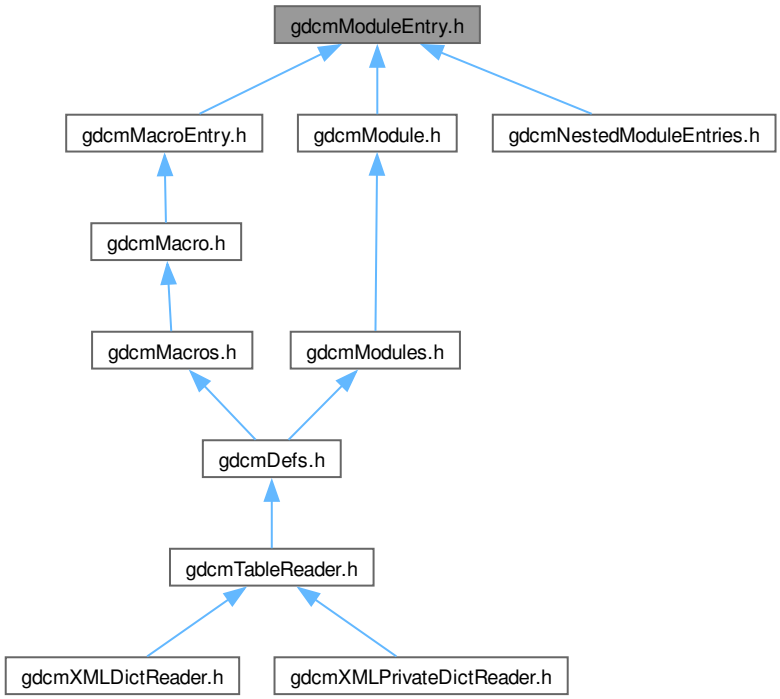
13.219 gdcmModuleEntry.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmType.h"
#include <string>
```

Include dependency graph for gdcmModuleEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ModuleEntry

Class for representing a ModuleEntry.

Namespaces

- namespace gdcm

Typedefs

- typedef ModuleEntry gdcm::MacroEntry

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)

13.220 gdcmModuleEntry.h

Go to the documentation of this file.

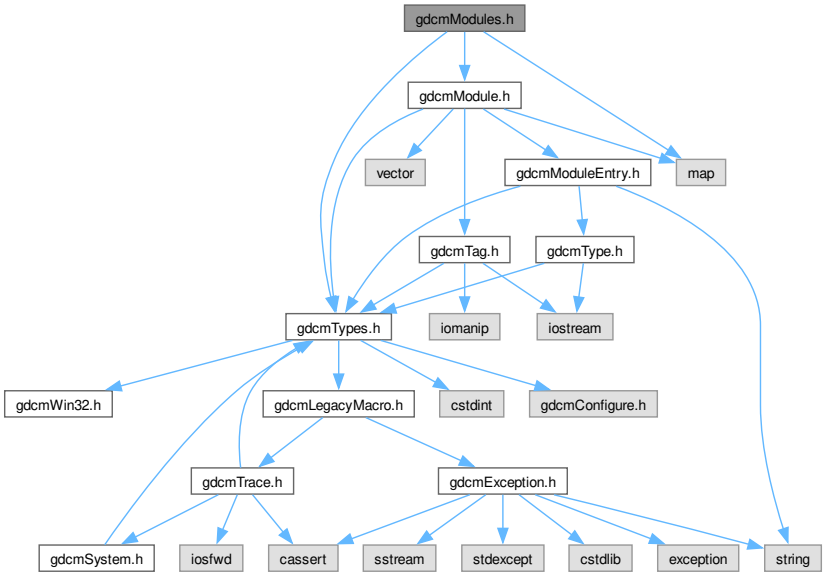
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMODULEENTRY_H
#define GDCMMODULEENTRY_H

#include "gdcmTypes.h"
#include "gdcmType.h"

#include <string>

namespace gdcm
{
class GDCM_EXPORT ModuleEntry
{
public:
  ModuleEntry(const char *name = "", const char *type = "3", const char *description = ""):Name(name)/*,Type(type)*/,DescriptionField(description) {
    DataElementType = Type::GetTypeType(type);
  }
  virtual ~ModuleEntry() = default; // important
  friend std::ostream& operator«(std::ostream& _os, const ModuleEntry &_val);

  void SetName(const char *name) { Name = name; }
  const char *GetName() const { return Name.c_str(); }

  void SetType(const Type &type) { DataElementType = type; }
  const Type &GetType() const { return DataElementType; }

  /*
   * WARNING: 'Description' is currently a std::string, but it might change in the future
   * do not expect it to remain the same, and always use the ModuleEntry::Description typedef
   * instead.
   */
  typedef std::string Description;
  void SetDescription(const char *d) { DescriptionField = d; }
```

```
  const Description & GetDescription() const { return DescriptionField; }

protected:
  // PS 3.3 repeats the name of an attribute, but often contains typos
  // for now we will not use this info, but instead access the DataDict instead
  std::string Name;

  // An attribute, encoded as a Data Element, may or may not be required in a
  // Data Set, depending on that Attribute's Data Element Type.
  Type DataElementType;

  // TODO: for now contains the raw description (with enumerated values, defined terms...)
  Description DescriptionField;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const ModuleEntry &_val)
{
  _os « _val.Name « "\t" « _val.DataElementType « "\t" « _val.DescriptionField;
  return _os;
}

typedef ModuleEntry MacroEntry;


} // end namespace gdcm

#endif //GDCMMODULEENTRY_H
```

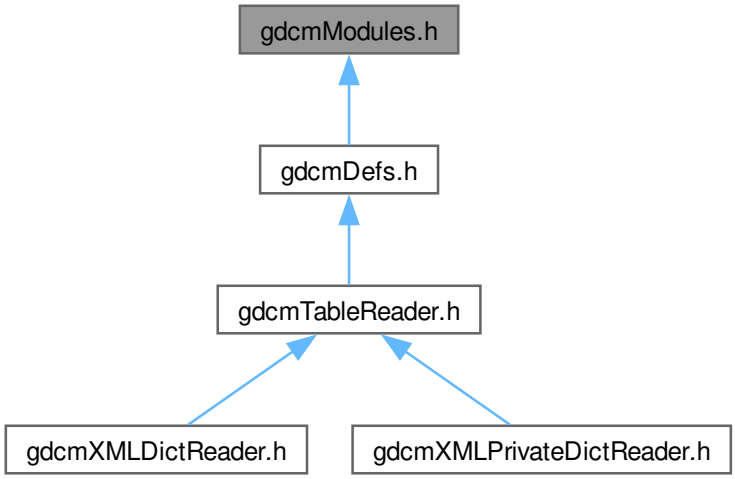
13.221 gdcmModules.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmModule.h"
#include <map>
```

Include dependency graph for gdcmModules.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Modules

  Class for representing a Modules.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)

13.222 gdcmModules.h

Go to the documentation of this file.

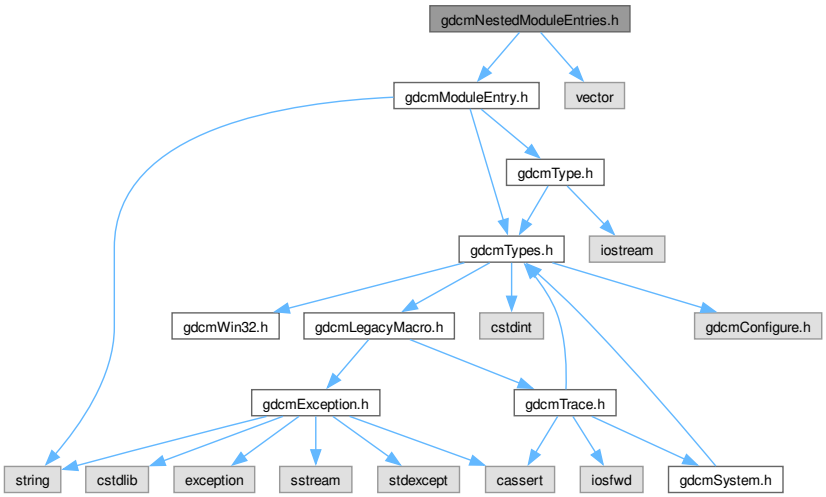
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
```

```
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMODULES_H
#define GDCMMODULES_H

#include "gdcmTypes.h"
#include "gdcmModule.h"

#include <map>

namespace gdcm
{
class GDCM_EXPORT Modules
{
public:
  typedef std::map<std::string, Module> ModuleMapType;

  Modules() = default;
  friend std::ostream& operator«(std::ostream& _os, const Modules &_val);

  void Clear() { ModulesInternal.clear(); }

  // A Module is inserted based on it's ref
  void AddModule(const char *ref, const Module & module )
    {
    assert( ref && *ref );
    assert( ModulesInternal.find( ref ) == ModulesInternal.end() );
    ModulesInternal.insert(
      ModuleMapType::value_type(ref, module));
    }
  const Module &GetModule(const char *name) const
    {
    assert( name && *name );
    ModuleMapType::const_iterator it = ModulesInternal.find( name );
    assert( it != ModulesInternal.end() );
    assert( it->first == name );
    return it->second;
    }

  bool IsEmpty() const { return ModulesInternal.empty(); }

private:
  ModuleMapType ModulesInternal;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const Modules &_val)
{
  Modules::ModuleMapType::const_iterator it = _val.ModulesInternal.begin();
  for(;it != _val.ModulesInternal.end(); ++it)
    {
    const std::string &name = it->first;
    const Module &m = it->second;
    _os « name « " " « m « '\n';
    }

  return _os;
}



} // end namespace gdcm

#endif //GDCMMODULES_H
```

13.223 gdcmNestedModuleEntries.h File Reference

```
#include "gdcmModuleEntry.h"
#include <vector>
```

Include dependency graph for gdcmNestedModuleEntries.h:



Classes

- class gdcm::NestedModuleEntries

  Class for representing a NestedModuleEntries.

Namespaces

- namespace gdcm

Typedefs

- typedef NestedModuleEntries gdcm::NestedMacroEntries

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const NestedModuleEntries &_val)

13.224 gdcmNestedModuleEntries.h

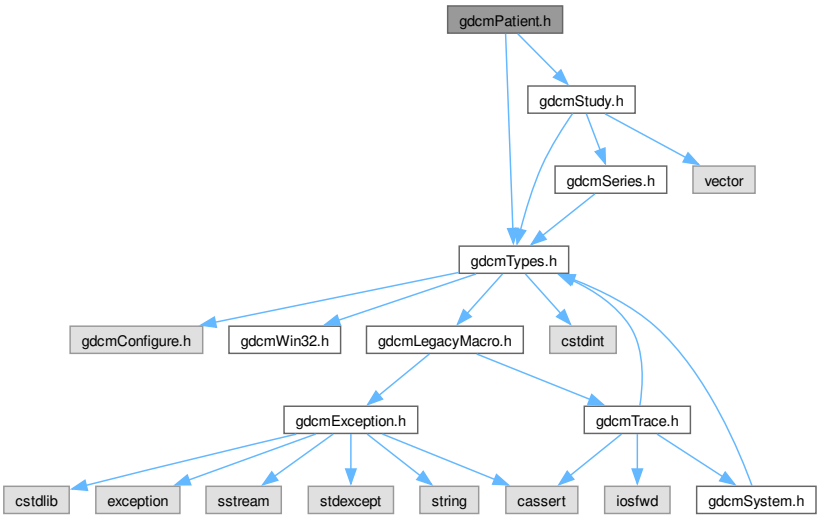
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMNESTEDMODULEENTRIES_H
#define GDCMNESTEDMODULEENTRIES_H

#include "gdcmModuleEntry.h"
#include <vector>

namespace gdcm
{
class GDCM_EXPORT NestedModuleEntries : public ModuleEntry
{
public:
  NestedModuleEntries(const char *name = "", const char *type = "3", const char *description = ""):ModuleEntry(name,type,description) { }
  friend std::ostream& operator«(std::ostream& _os, const NestedModuleEntries &_val);

  typedef std::vector<ModuleEntry>::size_type SizeType;
  SizeType GetNumberOfModuleEntries() { return ModuleEntriesList.size(); }

  const ModuleEntry &GetModuleEntry(SizeType idx) const { return ModuleEntriesList[idx]; }
  ModuleEntry &GetModuleEntry(SizeType idx) { return ModuleEntriesList[idx]; }

  void AddModuleEntry(const ModuleEntry &me) { ModuleEntriesList.push_back( me ); }

private:
  std::vector<ModuleEntry> ModuleEntriesList;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& _os, const NestedModuleEntries &_val)
{
  _os « "Nested:" « _val.Name « "\t" « _val.DataElementType « "\t" « _val.DescriptionField;
  return _os;
}

typedef NestedModuleEntries NestedMacroEntries;


} // end namespace gdcm

#endif //GDCMNESTEDMODULEENTRIES_H
```

13.225 gdcmPatient.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmStudy.h"
```

Include dependency graph for gdcmPatient.h:



Classes

- class gdcm::Patient

  See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

Namespaces

- namespace gdcm

13.226 gdcmPatient.h

Go to the documentation of this file.

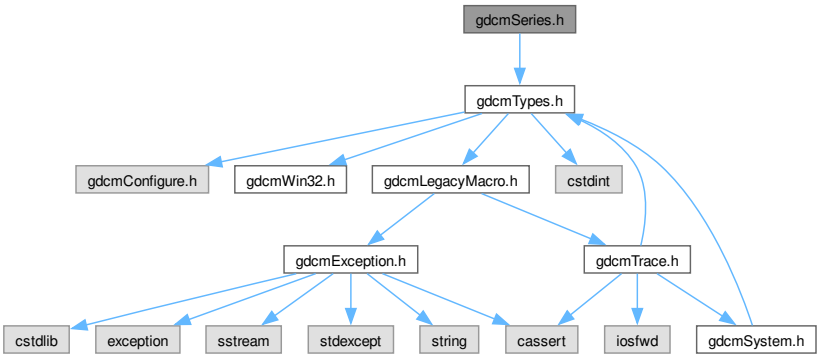
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPATIENT_H
#define GDCMPATIENT_H

#include "gdcmTypes.h"
#include "gdcmStudy.h"

```

```
namespace gdcm
{
class GDCM_EXPORT Patient
{
public:
  Patient() = default;

private:
  std::vector<Study> StudyList;
};

} // end namespace gdcm

#endif //GDCMPATIENT_H
```

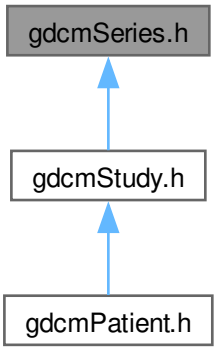
13.227 gdcmSeries.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmSeries.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Series

  Series.

Namespaces

- namespace gdcm

13.228 gdcmSeries.h

Go to the documentation of this file.

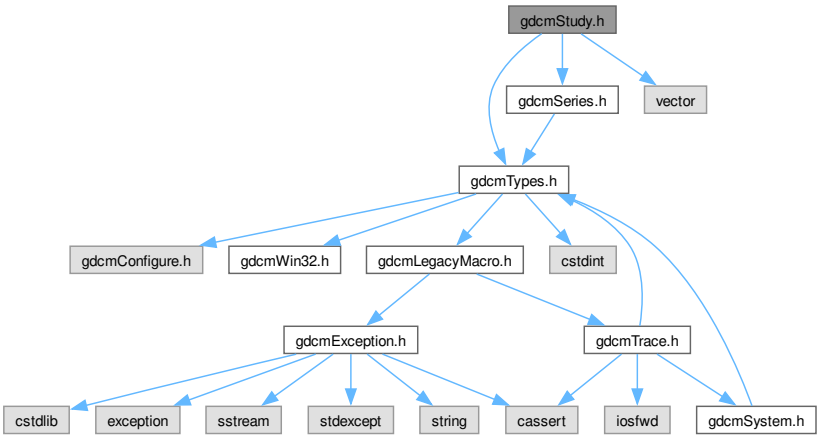
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSERIES_H
#define GDCMSERIES_H

#include "gdcmTypes.h"

namespace gdcm
{
class GDCM_EXPORT Series
{
public:
  Series() = default;
private:
  // Image, Waveform...
};

} // end namespace gdcm

#endif //GDCMSERIES_H
```

13.229 gdcmStudy.h File Reference

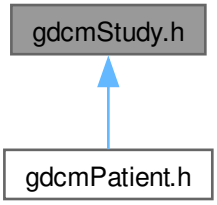
```
#include "gdcmTypes.h"
#include "gdcmSeries.h"
```

```
#include <vector>
```

Include dependency graph for gdcmStudy.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Study

  Study.

Namespaces

- namespace gdcm

13.230 gdcmStudy.h

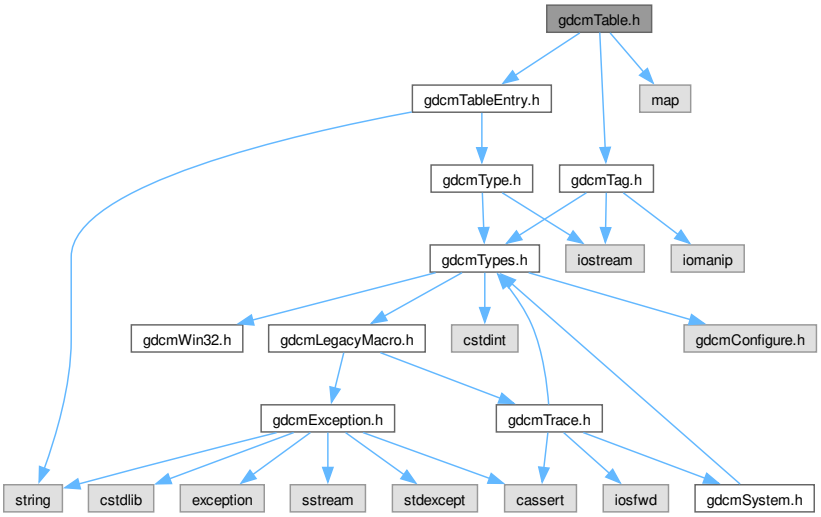
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSTUDY_H
#define GDCMSTUDY_H

#include "gdcmTypes.h"
#include "gdcmSeries.h"

#include <vector>

namespace gdcm
{
class GDCM_EXPORT Study
{
public:
  Study() = default;
private:
  std::vector<Series> SeriesList;
};

} // end namespace gdcm

#endif //GDCMSTUDY_H
```

13.231 gdcmTable.h File Reference

```
#include "gdcmTableEntry.h"
#include "gdcmTag.h"
#include <map>
```

Include dependency graph for gdcmTable.h:



Classes

- class gdcm::Table

  Table.

Namespaces

- namespace gdcm

13.232 gdcmTable.h

Go to the documentation of this file.

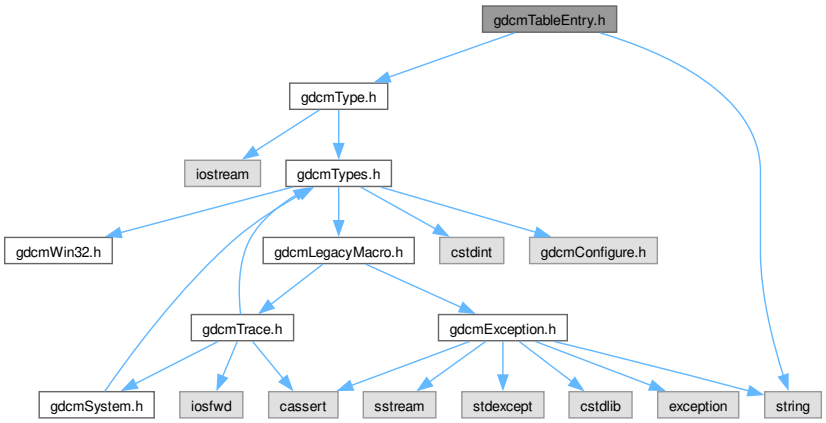
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTABLE_H
#define GDCMTABLE_H

#include "gdcmTableEntry.h"
#include "gdcmTag.h"

#include <map>
```

```
00021
00022 namespace gdcm
00023 {
00024
00028 class Table
00029 {
00030 public:
00031   typedef std::map<Tag, TableEntry> MapTableEntry;
00032   Table() = default;
00033   ~Table() = default;
00034   Table &operator=(const Table &_val) = delete;
00035   Table(const Table&_val) = delete;
00036
00037   friend std::ostream& operator«(std::ostream& _os, const Table &_val);
00038
00039   void InsertEntry(Tag const &tag, TableEntry const &te)
00040     {
00041 #ifndef NDEBUG
00042     MapTableEntry::size_type s = TableInternal.size();
00043 #endif
00044     TableInternal.insert(
00045       MapTableEntry::value_type(tag, te));
00046     assert( s < TableInternal.size() );
00047     }
00048
00049   const TableEntry &GetTableEntry(const Tag &tag) const
00050     {
00051     MapTableEntry::const_iterator it =
00052       TableInternal.find(tag);
00053     if (it == TableInternal.end())
00054       {
00055       assert( 0 && "Impossible" );
00056       return GetTableEntry(Tag(0,0));
00057       }
00058     return it->second;
00059     }
00060
00061   MapTableEntry TableInternal;
00062 };
00063
00064 } // end namespace gdcm
00065
00066 #endif //GDCMTABLE_H
```

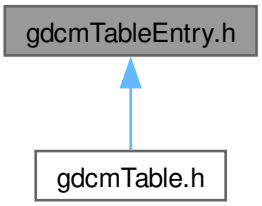
13.233 gdcmTableEntry.h File Reference

```
#include "gdcmType.h"
#include <string>
```

Include dependency graph for gdcmTableEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::TableEntry

  TableEntry.

Namespaces

- namespace gdcm

13.234 gdcmTableEntry.h

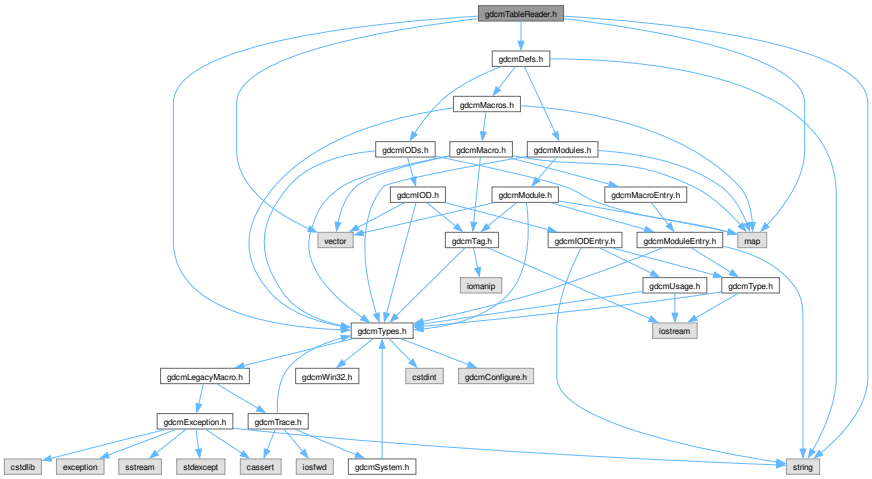
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTABLEENTRY_H
#define GDCMTABLEENTRY_H

#include "gdcmType.h"

#include <string>

namespace gdcm
{

class TableEntry
{
public:
  TableEntry(const char *attribute = nullptr,
    Type const &type = Type(), const char * des = nullptr ) :
    Attribute(attribute ? attribute : ""),TypeField(type),Description(des ? des : "") {}
  ~TableEntry() = default;

private:
  std::string Attribute;
  Type TypeField;
  std::string Description;
};

} // end namespace gdcm

#endif //GDCMTABLEENTRY_H
```

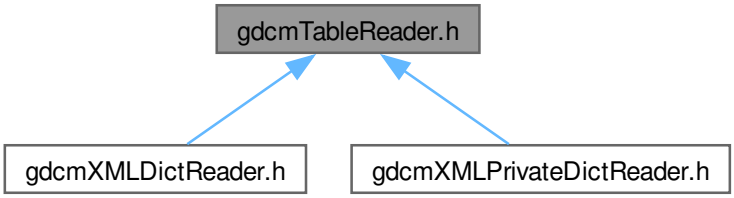
13.235 gdcmTableReader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDefs.h"
#include <string>
#include <vector>
#include <map>
```

Include dependency graph for gdcmTableReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::TableReader

  Class for representing a TableReader.

Namespaces

- namespace gdcm

13.236 gdcmTableReader.h

Go to the documentation of this file.

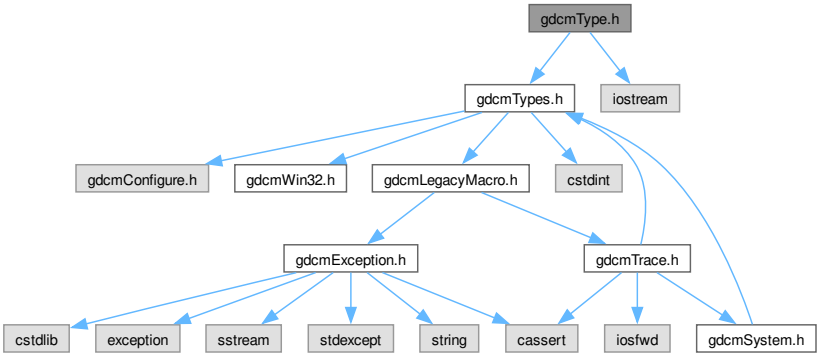
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTABLEREADER_H
#define GDCMTABLEREADER_H

#include "gdcmTypes.h"
#include "gdcmDefs.h"
//#include "gdcmModule.h"
//#include "gdcmIOD.h"
//#include "gdcmIODs.h"
//#include "gdcmModules.h"

#include <string>
#include <vector>
#include <map>

namespace gdcm
{
class GDCM_EXPORT TableReader
{
public:
  TableReader(Defs &defs):CurrentDefs(defs),ParsingModule(false),ParsingModuleEntry(false),
  ParsingModuleEntryDescription(false),
  ParsingMacro(false),
  ParsingMacroEntry(false),
  ParsingMacroEntryDescription(false),
  ParsingIOD(false),
  ParsingIODEntry(false),
  Description() {}
  virtual ~TableReader() = default;

  // Set/Get filename
  void SetFilename(const char *filename) { Filename = filename; }
  const char *GetFilename() { return Filename.c_str(); }

  int Read();

//protected:
  // You need to override those function in your subclasses:
  virtual void StartElement(const char *name, const char **atts);
  virtual void EndElement(const char *name);
  virtual void CharacterDataHandler(const char *data, int length);

void HandleModuleEntry(const char **atts);
void HandleModule(const char **atts);
void HandleModuleEntryDescription(const char **atts);
void HandleMacroEntry(const char **atts);
void HandleMacro(const char **atts);
void HandleMacroEntryDescription(const char **atts);
void HandleModuleInclude(const char **atts);
void HandleIODEntry(const char **atts);
void HandleIOD(const char **atts);

  //const Modules & GetModules() const { return CurrentModules; }
  //const Macros & GetMacros() const { return CurrentMacros; }
  //const IODs & GetIODs() const { return CurrentIODs; }
  const Defs & GetDefs() const { return CurrentDefs; }

private:
  std::string Filename;
  Defs &CurrentDefs;
  //Macros CurrentMacros;
  //Modules CurrentModules;
  //IODs CurrentIODs;
```

```
  Macro CurrentMacro;
  Module CurrentModule;
  IOD CurrentIOD;
  MacroEntry CurrentMacroEntry;
  ModuleEntry CurrentModuleEntry;
  IODEntry CurrentIODEntry;
  std::string CurrentModuleName;
  std::string CurrentModuleRef;
  std::string CurrentMacroRef;
  bool ParsingModule;
  bool ParsingModuleEntry;
  bool ParsingModuleEntryDescription;
  bool ParsingMacro;
  bool ParsingMacroEntry;
  bool ParsingMacroEntryDescription;
  bool ParsingIOD;
  bool ParsingIODEntry;
  Tag CurrentTag;
  std::string Description;
};

} // end namespace gdcm

#endif //GDCMTABLEREADER_H
```

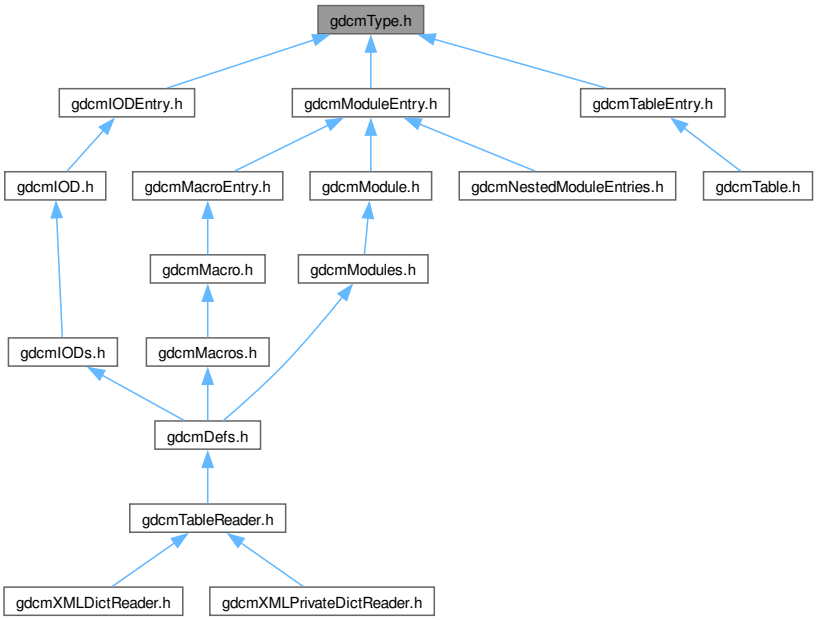
13.237 gdcmType.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmType.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Type

  Type.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)

13.238 gdcmType.h

Go to the documentation of this file.

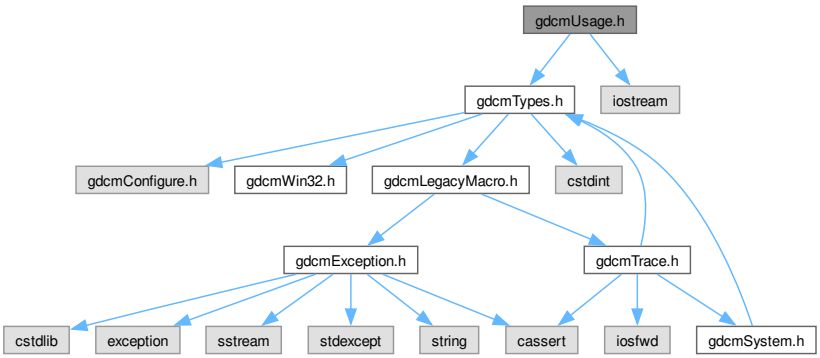
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014
00015 #ifndef GDCMTYPE_H
00016 #define GDCMTYPE_H
00017
00018 #include "gdcmTypes.h"
00019
00020 #include <iostream>
00021
00022 namespace gdcm
00023 {
00024
00041 class GDCM_EXPORT Type
00042 {
00043 public:
00044   typedef enum {
00045     T1 = 0,
00046     T1C,
00047     T2,
00048     T2C,
00049     T3,
00050     UNKNOWN
00051   } TypeType;
00052
00053   Type(TypeType type = UNKNOWN) : TypeField(type) { }
00054
00055   operator TypeType () const { return TypeField; }
00056   friend std::ostream &operator«(std::ostream &os, const Type &vr);
00057
00058   static const char *GetTypeString(TypeType type);
00059   static TypeType GetTypeType(const char *type);
00060
00061 private:
00062   TypeType TypeField;
00063 };
00064 //-----------------------------------------------------------------------------
00065 inline std::ostream &operator«(std::ostream &_os, const Type &val)
00066 {
00067   _os « Type::GetTypeString(val.TypeField);
00068   return _os;
00069 }
00070
00071 } // end namespace gdcm
00072
00073 #endif //GDCMTYPE_H
```

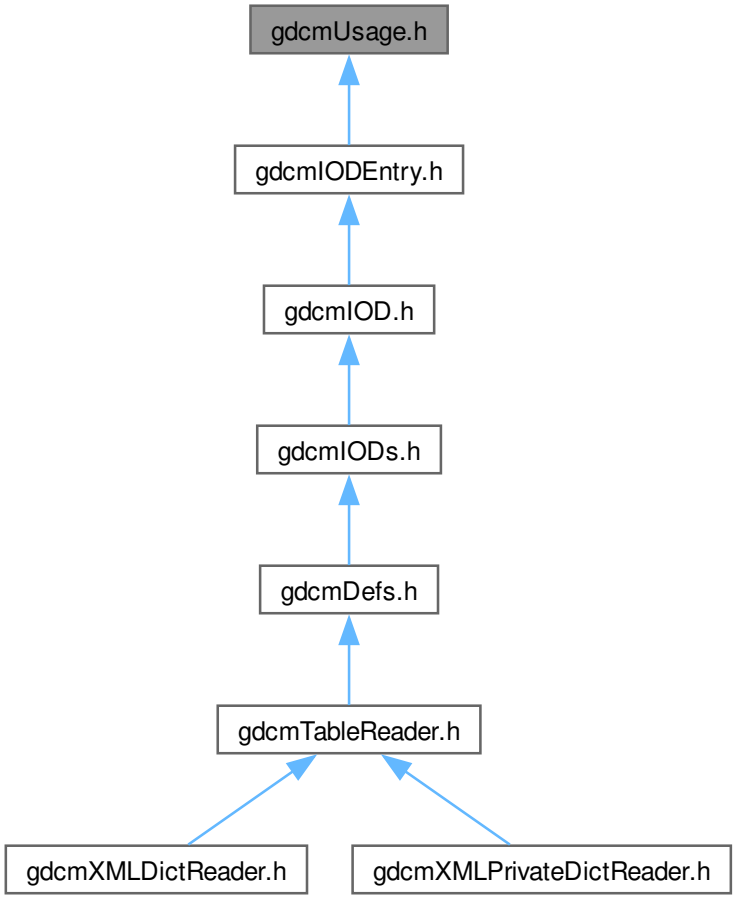
13.239 gdcmUsage.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmUsage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Usage

  Usage.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)

13.240 gdcmUsage.h

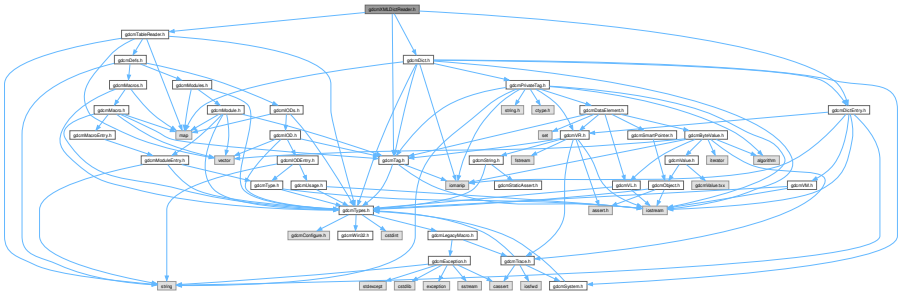
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMUSAGE_H
#define GDCMUSAGE_H

#include "gdcmTypes.h"

#include <iostream>

namespace gdcm
{

class GDCM_EXPORT Usage
{
public:
  typedef enum {
    Mandatory, // (see A.1.3.1) , abbreviated M
    Conditional, // (see A.1.3.2) , abbreviated C
    UserOption, // (see A.1.3.3) , abbreviated U
    Invalid
  } UsageType;

  Usage(UsageType type = Invalid) : UsageField(type) { }

  operator UsageType () const { return UsageField; }
  friend std::ostream &operator«(std::ostream &os, const Usage &vr);

  static const char *GetUsageString(UsageType type);
  static UsageType GetUsageType(const char *type);

private:
  UsageType UsageField;
};
//-----------------------------------------------------------------------------
inline std::ostream &operator«(std::ostream &_os, const Usage &val)
{
  _os « Usage::GetUsageString(val.UsageField);
  return _os;
}

} // end namespace gdcm

#endif //GDCMUSAGE_H
```

13.241 gdcmXMLDictReader.h File Reference

```
#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmXMLDictReader.h:



Classes

- class gdcm::XMLDictReader

  Class for representing a XMLDictReader.

Namespaces

- namespace gdcm

13.242 gdcmXMLDictReader.h

Go to the documentation of this file.

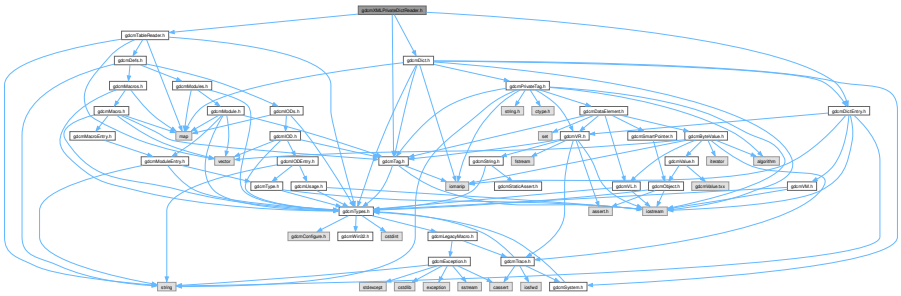
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMXMLDICTREADER_H
#define GDCMXMLDICTREADER_H

#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"

namespace gdcm
{
class GDCM_EXPORT XMLDictReader : public TableReader
{
public:
  XMLDictReader();
  ~XMLDictReader() {}

  void StartElement(const char *name, const char **atts);
  void EndElement(const char *name);
  void CharacterDataHandler(const char *data, int length);

  const Dict & GetDict() { return DICOMDict; }
```

```
00040
00041 protected:
00042   void HandleEntry(const char **atts);
00043   void HandleDescription(const char **atts);
00044
00045 private:
00046   Dict DICOMDict;
00047   Tag CurrentTag;
00048   DictEntry CurrentDE;
00049   bool ParsingDescription;
00050   std::string Description;
00051 };
00052
00053 } // end namespace gdcm
00054
00055 #endif //GDCMXMLDICTREADER_H
```

13.243 gdcmXMLPrivateDictReader.h File Reference

```
#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmXMLPrivateDictReader.h:



Classes

- class gdcm::XMLPrivateDictReader

  Class for representing a XMLPrivateDictReader.

Namespaces

- namespace gdcm

13.244 gdcmXMLPrivateDictReader.h

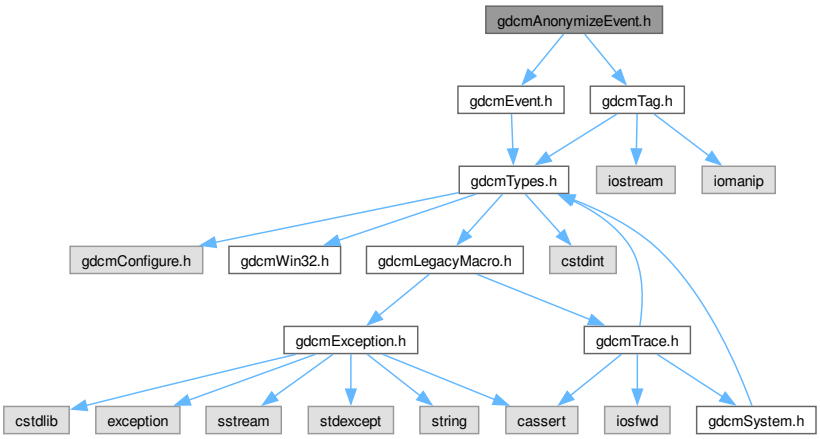
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMXMLPRIVATEDICTREADER_H
#define GDCMXMLPRIVATEDICTREADER_H

#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"

namespace gdcm
{
class GDCM_EXPORT XMLPrivateDictReader : public TableReader
{
public:
  XMLPrivateDictReader();
  ~XMLPrivateDictReader() {}

  void StartElement(const char *name, const char **atts);
  void EndElement(const char *name);
  void CharacterDataHandler(const char *data, int length);

  const PrivateDict & GetPrivateDict() { return PDict; }

protected:
  void HandleEntry(const char **atts);
  void HandleDescription(const char **atts);

private:
  PrivateDict PDict;
  PrivateTag CurrentTag;
  DictEntry CurrentDE;
  bool ParsingDescription;
  std::string Description;
};

} // end namespace gdcm

#endif //GDCMXMLPRIVATEDICTREADER_H
```

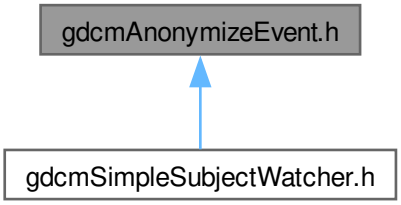
13.245 gdcmAnonymizeEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmAnonymizeEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::AnonymizeEvent

  AnonymizeEvent.

Namespaces

- namespace gdcm

13.246 gdcmAnonymizeEvent.h

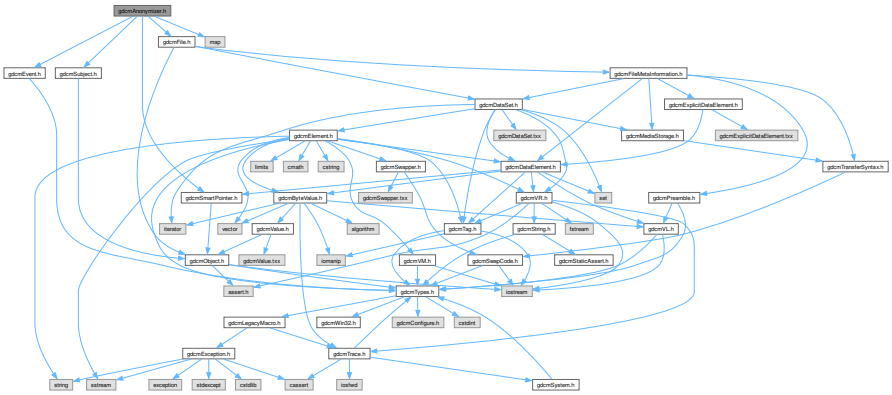
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMANONYMIZEEVENT_H
#define GDCMANONYMIZEEVENT_H

#include "gdcmEvent.h"
#include "gdcmTag.h"

namespace gdcm
{

class AnonymizeEvent : public AnyEvent
{
public:
  typedef AnonymizeEvent Self;
  typedef AnyEvent Superclass;
  AnonymizeEvent(Tag const &tag = 0):m_Tag(tag) {}
  ~AnonymizeEvent() override = default;
  AnonymizeEvent(const Self&s) : AnyEvent(s){}
  void operator=(const Self&) = delete;

  const char * GetEventName() const override { return "AnonymizeEvent"; }
  bool CheckEvent(const ::gdcm::Event* e) const override
  { return (dynamic_cast<const Self*>(e) == nullptr ? false : true) ; }
  ::gdcm::Event* MakeObject() const override
    { return new Self; }

  void SetTag(const Tag& t ) { m_Tag = t; }
  Tag const & GetTag() const { return m_Tag; }
private:
  Tag m_Tag;
};


} // end namespace gdcm

#endif //GDCMANONYMIZEEVENT_H
```

13.247 gdcmAnonymizer.h File Reference

```
#include "gdcmFile.h"
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmSmartPointer.h"
#include <map>
```

Include dependency graph for gdcmAnonymizer.h:



Classes

- class gdcm::Anonymizer

  Anonymizer.

Namespaces

- namespace gdcm

13.248 gdcmAnonymizer.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMANONYMIZER_H
#define GDCMANONYMIZER_H

#include "gdcmFile.h"
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmSmartPointer.h"

#include <map>

namespace gdcm
{
class TagPath;
class IOD;
class CryptographicMessageSyntax;

```

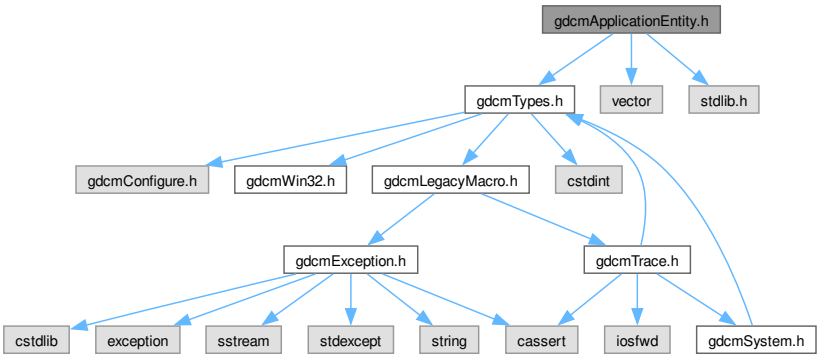
```
class GDCM_EXPORT Anonymizer : public Subject
{
public:
  Anonymizer():F(new File),CMS(nullptr) {}
  ~Anonymizer() override;

  bool Empty( Tag const &t );

  bool Empty( PrivateTag const &pt );

  bool Clear( Tag const &t );
  bool Clear( PrivateTag const &pt );

  bool Remove( Tag const &t );

  bool Remove( PrivateTag const &pt );

  bool Replace( Tag const &t, const char *value );
  bool Replace( PrivateTag const &t, const char *value );

  bool Replace( Tag const &t, const char *value, VL const & vl );
  bool Replace( PrivateTag const &t, const char *value, VL const & vl );

  bool RemovePrivateTags();

  bool RemoveGroupLength();

  bool RemoveRetired();

  void SetFile(const File& f) { F = f; }
  //const File &GetFile() const { return *F; }
  File &GetFile() { return *F; }

  bool BasicApplicationLevelConfidentialityProfile(bool deidentify = true);

  void SetCryptographicMessageSyntax( CryptographicMessageSyntax *cms );
  const CryptographicMessageSyntax *GetCryptographicMessageSyntax() const;

  static SmartPointer<Anonymizer> New() { return new Anonymizer; }

  static std::vector<Tag> GetBasicApplicationLevelConfidentialityProfileAttributes();

  static void ClearInternalUIDs();

protected:
  // Internal function used to either empty a tag or set it's value to a dummy value (Type 1 vs Type 2)
  bool BALCPProtect(DataSet &ds, Tag const & tag, const IOD &iod);
  bool CanEmptyTag(Tag const &tag, const IOD &iod) const;
  void RecurseDataSet( DataSet & ds );

private:
  bool BasicApplicationLevelConfidentialityProfile1();
  bool BasicApplicationLevelConfidentialityProfile2();
  bool CheckIfSequenceContainsAttributeToAnonymize(File const &file, SequenceOfItems* sqi) const;

private:
  // I would prefer to have a smart pointer to DataSet but DataSet does not derive from Object...
  SmartPointer<File> F;
  CryptographicMessageSyntax *CMS;

  typedef std::pair< Tag, std::string > TagValueKey;
  typedef std::map< TagValueKey, std::string > DummyMapNonUIDTags;
  typedef std::map< std::string, std::string > DummyMapUIDTags;
  static DummyMapNonUIDTags dummyMapNonUIDTags;
  static DummyMapUIDTags dummyMapUIDTags;
};


} // end namespace gdcm

#endif //GDCMANONYMIZER_H
```

13.249 gdcmApplicationEntity.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
```

`#include <stdlib.h>`

Include dependency graph for gdcmApplicationEntity.h:



Classes

- class gdcm::ApplicationEntity

  ApplicationEntity.

Namespaces

- namespace gdcm

13.250 gdcmApplicationEntity.h

Go to the documentation of this file.

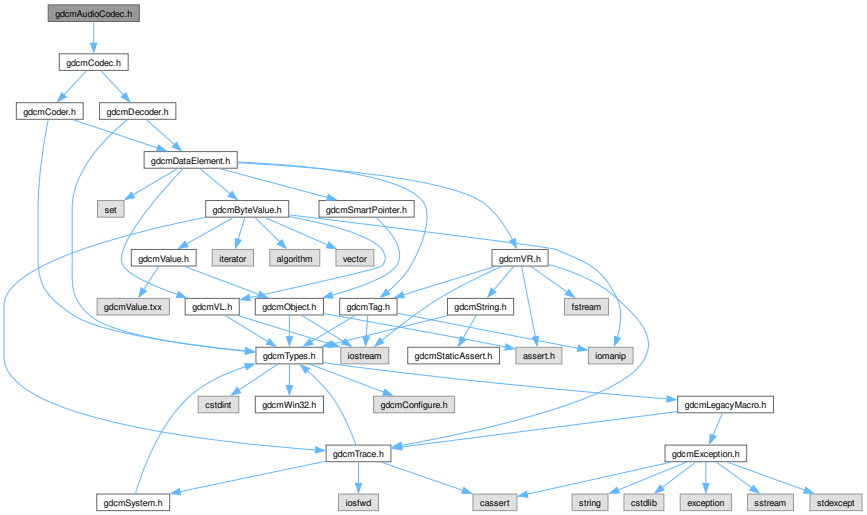
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMAPPLICATIONENTITY_H
#define GDCMAPPLICATIONENTITY_H

#include "gdcmTypes.h"
#include <vector>
#include <stdlib.h> // abort

namespace gdcm
{

class GDCM_EXPORT ApplicationEntity
{
```

```
public:
  static const unsigned int MaxNumberOfComponents = 1;
  static const unsigned int MaxLength = 16;
  std::string Internal;
  static const char Separator = ' ';
  static const char Padding   = ' ';
  //static const char Excluded[5] = { '\\' /* 5CH */, '\n' /* LF */, '\f', /* FF */, '\r' /* CR */, 0x1b /* ESC */};

  bool IsValid() const {
    return true;
  }
  void Squeeze() {
    // trim leading and trailing white spaces
  }
  void SetBlob(const std::vector<char>& v) {
    (void)v;
    assert(0); //TODO
  }
  void Print(std::ostream &os) const {
  (void)os;
    assert(0); //TODO
  }
};

} // end namespace gdcm

#endif //GDCMAPPLICATIONENTITY_H
```

13.251 gdcmAudioCodec.h File Reference

`#include "gdcmCodec.h"`

Include dependency graph for gdcmAudioCodec.h:



Classes

- class gdcm::AudioCodec

  AudioCodec.

Namespaces

- namespace gdcm

13.252 gdcmAudioCodec.h

Go to the documentation of this file.

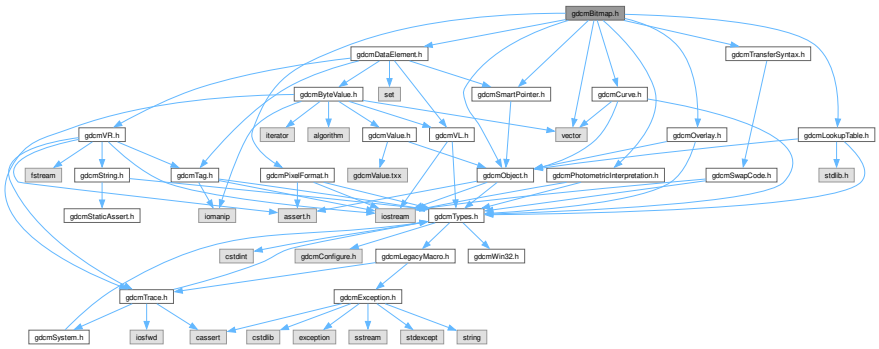
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMAUDIOCODEC_H
#define GDCMAUDIOCODEC_H

#include "gdcmCodec.h"

namespace gdcm
{

class GDCM_EXPORT AudioCodec : public Codec
{
public:
  AudioCodec();
  ~AudioCodec() override;
  bool CanCode(TransferSyntax const &) const override { return false; }
  bool CanDecode(TransferSyntax const &) const override { return false; }
  bool Decode(DataElement const &is, DataElement &os) override;
};

} // end namespace gdcm

#endif //GDCMAUDIOCODEC_H
```

13.253 gdcmBitmap.h File Reference

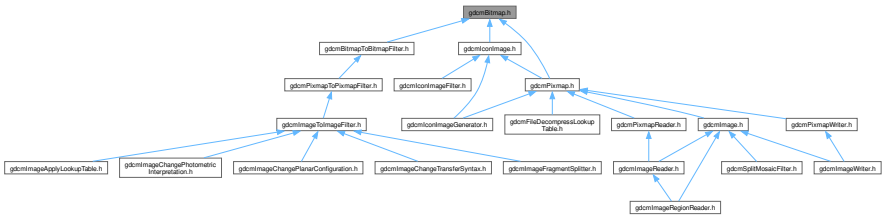
```
#include "gdcmObject.h"
#include "gdcmCurve.h"
#include "gdcmDataElement.h"
#include "gdcmLookupTable.h"
#include "gdcmOverlay.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmPixelFormat.h"
#include "gdcmSmartPointer.h"
#include "gdcmTransferSyntax.h"
```

```
#include <vector>
```

Include dependency graph for gdcmBitmap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Bitmap

  Bitmap class.

Namespaces

- namespace gdcm

13.254 gdcmBitmap.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
```

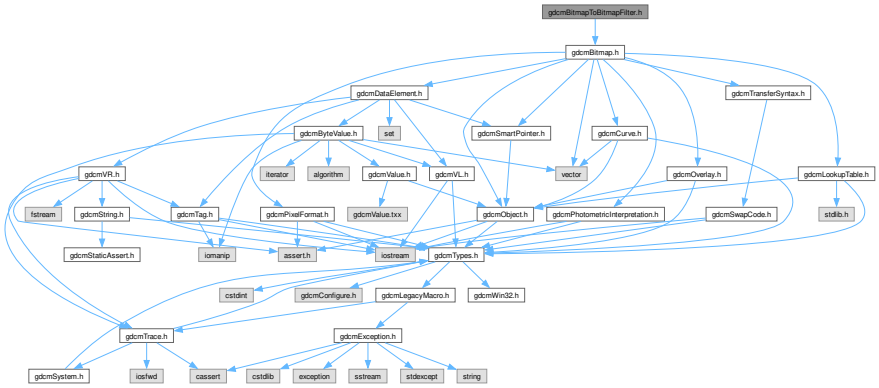
```
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMBITMAP_H
#define GDCMBITMAP_H

#include "gdcmObject.h"
#include "gdcmCurve.h"
#include "gdcmDataElement.h"
//#include "gdcmIconImage.h"
#include "gdcmLookupTable.h"
#include "gdcmOverlay.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmPixelFormat.h"
#include "gdcmSmartPointer.h"
#include "gdcmTransferSyntax.h"

#include <vector>

namespace gdcm
{

class GDCM_EXPORT Bitmap : public Object
{
public:
  Bitmap();
  ~Bitmap() override;
  void Print(std::ostream &) const override;

  virtual bool AreOverlaysInPixelData() const { return false; }
  virtual bool UnusedBitsPresentInPixelData() const { return false; }

  unsigned int GetNumberOfDimensions() const;
  void SetNumberOfDimensions(unsigned int dim);

  unsigned int GetPlanarConfiguration() const;
  void SetPlanarConfiguration(unsigned int pc);

  bool GetNeedByteSwap() const
    {
    return NeedByteSwap;
    }
  void SetNeedByteSwap(bool b)
    {
    NeedByteSwap = b;
    }


  void SetTransferSyntax(TransferSyntax const &ts) {
    TS = ts;
  }
  const TransferSyntax &GetTransferSyntax() const {
    return TS;
  }
  bool IsTransferSyntaxCompatible( TransferSyntax const & ts ) const;
  void SetDataElement(DataElement const &de) {
    PixelData = de;
  }
  const DataElement& GetDataElement() const { return PixelData; }
  DataElement& GetDataElement() { return PixelData; }

  void SetLUT(LookupTable const &lut)
    {
    LUT = SmartPointer<LookupTable>( const_cast<LookupTable*>(&lut) );
    }
  const LookupTable &GetLUT() const
    {
    return *LUT;
    }
  LookupTable &GetLUT()
    {
    return *LUT;
    }

  const unsigned int *GetDimensions() const;
  unsigned int GetDimension(unsigned int idx) const;

  void SetColumns(unsigned int col) { SetDimension(0,col); }
  unsigned int GetColumns() const { return GetDimension(0); }
  void SetRows(unsigned int rows) { SetDimension(1,rows); }
```

```
  unsigned int GetRows() const { return GetDimension(1); }
  void SetDimensions(const unsigned int dims[3]);
  void SetDimension(unsigned int idx, unsigned int dim);
  const PixelFormat &GetPixelFormat() const
    {
    return PF;
    }
  PixelFormat &GetPixelFormat()
    {
    return PF;
    }
  void SetPixelFormat(PixelFormat const &pf)
    {
    PF = pf;
    PF.Validate();
    }

  const PhotometricInterpretation &GetPhotometricInterpretation() const;
  void SetPhotometricInterpretation(PhotometricInterpretation const &pi);

  bool IsEmpty() const { return Dimensions.empty(); }
  void Clear();

  unsigned long GetBufferLength() const;

  bool GetBuffer(char *buffer) const;

  bool IsLossy() const;

  void SetLossyFlag(bool f) { LossyFlag = f; }

protected:
  bool TryRAWCodec(char *buffer, bool &lossyflag) const;
  bool TryJPEGCodec(char *buffer, bool &lossyflag) const;
  bool TryPVRGCodec(char *buffer, bool &lossyflag) const;
  bool TryKAKADUCodec(char *buffer, bool &lossyflag) const;
  bool TryJPEGLSCodec(char *buffer, bool &lossyflag) const;
  bool TryJPEG2000Codec(char *buffer, bool &lossyflag) const;
  bool TryRLECodec(char *buffer, bool &lossyflag) const;

  bool TryJPEGCodec2(std::ostream &os) const;
  bool TryJPEG2000Codec2(std::ostream &os) const;

  bool GetBuffer2(std::ostream &os) const;

  friend class PixmapReader;
  friend class ImageChangeTransferSyntax;
  // Function to compute the lossy flag based only on the image buffer.
  // Watch out that image can be lossy but in implicit little endian format...
  bool ComputeLossyFlag();

//private:
protected:
  unsigned int PlanarConfiguration;
  unsigned int NumberOfDimensions;
  TransferSyntax TS;
  PixelFormat PF; // SamplesPerPixel, BitsAllocated, BitsStored, HighBit, PixelRepresentation
  PhotometricInterpretation PI;
  // Mind dump: unsigned int is required here, since we are reading (0028,0008) Number Of Frames
  // which is VR::IS, so I cannot simply assumed that unsigned short is enough... :(
  std::vector<unsigned int> Dimensions; // Col/Row
  DataElement PixelData; // copied from 7fe0,0010

  typedef SmartPointer<LookupTable> LUTPtr;
  LUTPtr LUT;
  // I believe the following 3 ivars can be derived from TS ...
  bool NeedByteSwap; // FIXME: remove me
  bool LossyFlag;

private:
  bool GetBufferInternal(char *buffer, bool &lossyflag) const;
};

} // end namespace gdcm

#endif //GDCMBITMAP_H
```

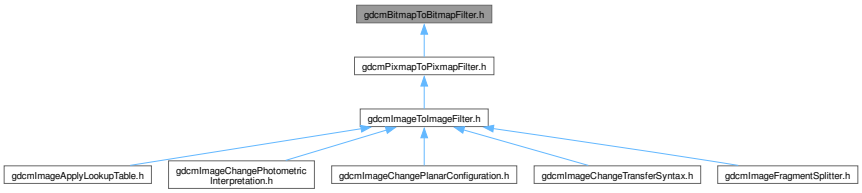
13.255 gdcmBitmapToBitmapFilter.h File Reference

```
#include "gdcmBitmap.h"
```

Include dependency graph for gdcmBitmapToBitmapFilter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::BitmapToBitmapFilter

  BitmapToBitmapFilter class.

Namespaces

- namespace gdcm

13.256 gdcmBitmapToBitmapFilter.h

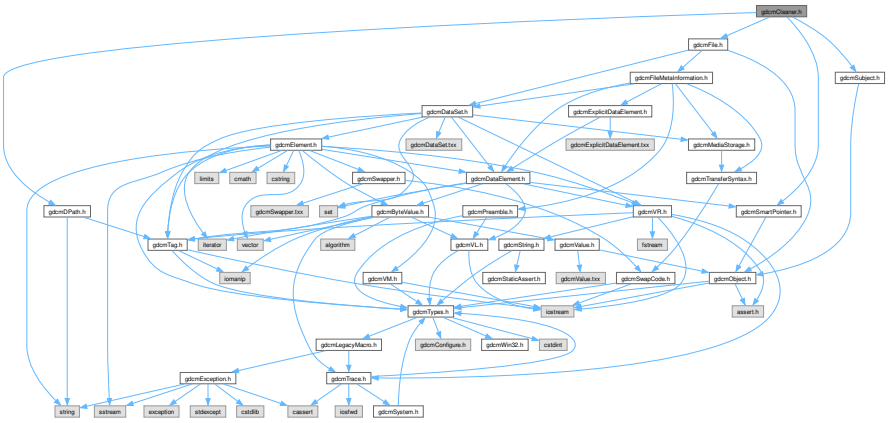
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMBITMAPTOBITMAPFILTER_H
#define GDCMBITMAPTOBITMAPFILTER_H

#include "gdcmBitmap.h"

namespace gdcm
{

class GDCM_EXPORT BitmapToBitmapFilter
{
public:
  BitmapToBitmapFilter();
  ~BitmapToBitmapFilter() = default;

  void SetInput(const Bitmap& image);

  const Bitmap &GetOutput() const { return *Output; }

  // SWIG/Java hack:
  const Bitmap &GetOutputAsBitmap() const;

protected:
  SmartPointer<Bitmap> Input;
  SmartPointer<Bitmap> Output;
};

} // end namespace gdcm

#endif //GDCMBITMAPTOBITMAPFILTER_H
```

13.257 gdcmCleaner.h File Reference

```
#include "gdcmDPath.h"
#include "gdcmFile.h"
#include "gdcmSmartPointer.h"
#include "gdcmSubject.h"
```

Include dependency graph for gdcmCleaner.h:



Classes

- class gdcm::Cleaner

  Cleaner.

Namespaces

- namespace gdcm

13.258 gdcmCleaner.h

Go to the documentation of this file.

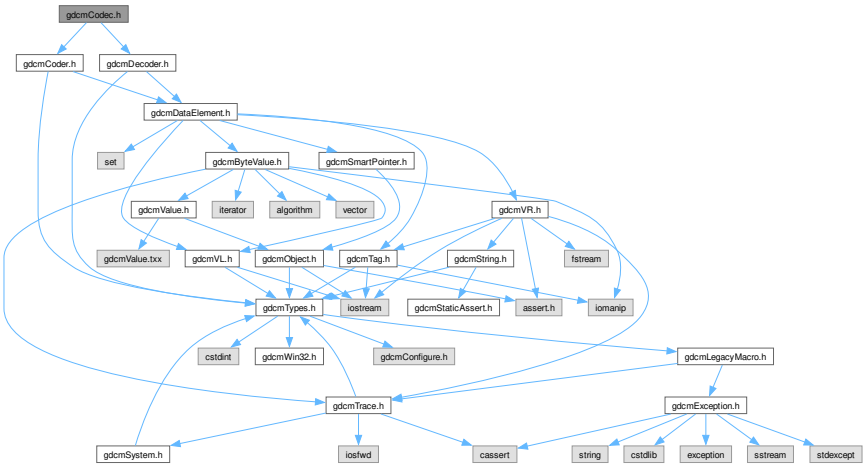
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCLEANER_H
#define GDCMCLEANER_H

#include "gdcmDPath.h"
#include "gdcmFile.h"
#include "gdcmSmartPointer.h"
#include "gdcmSubject.h"

namespace gdcm {
class GDCM_EXPORT Cleaner : public Subject {
 public:
  Cleaner();
  ~Cleaner() override;

```

```
  bool Empty(Tag const &t);
  bool Empty(PrivateTag const &pt);
  bool Empty(DPath const &dpath);
  bool Empty(VR const &vr);

  bool Remove(Tag const &t);
  bool Remove(PrivateTag const &pt);
  bool Remove(DPath const &dpath);
  bool Remove(VR const &vr);

  bool Scrub(Tag const &t);
  bool Scrub(PrivateTag const &pt);
  bool Scrub(DPath const &dpath);
  bool Scrub(VR const &vr);

  bool Preserve(DPath const &dpath);

  void RemoveAllMissingPrivateCreator(bool remove);

  bool RemoveMissingPrivateCreator(Tag const &t);

  void RemoveAllGroupLength(bool remove);

  void RemoveAllIllegal(bool remove);

  void EmptyWhenScrubFails(bool empty);

  bool Clean();

  void SetFile(const File &f) { F = f; }
  // const File &GetFile() const { return *F; }
  File &GetFile() { return *F; }

  static SmartPointer<Cleaner> New() { return new Cleaner; }

 private:
  // I would prefer to have a smart pointer to DataSet but DataSet does not
  // derive from Object...
  SmartPointer<File> F;
  struct impl;
  // PIMPL idiom
  impl *pimpl;
};

}  // end namespace gdcm

#endif  // GDCMCLEANER_H
```

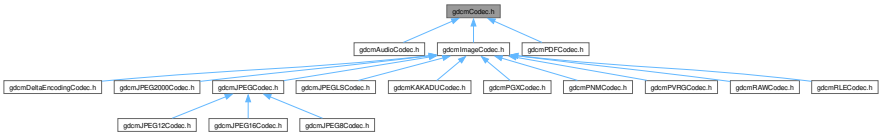
13.259 gdcmCodec.h File Reference

```
#include "gdcmCoder.h"
#include "gdcmDecoder.h"
```

Include dependency graph for gdcmCodec.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Codec

  Codec class.

Namespaces

- namespace gdcm

13.260 gdcmCodec.h

Go to the documentation of this file.

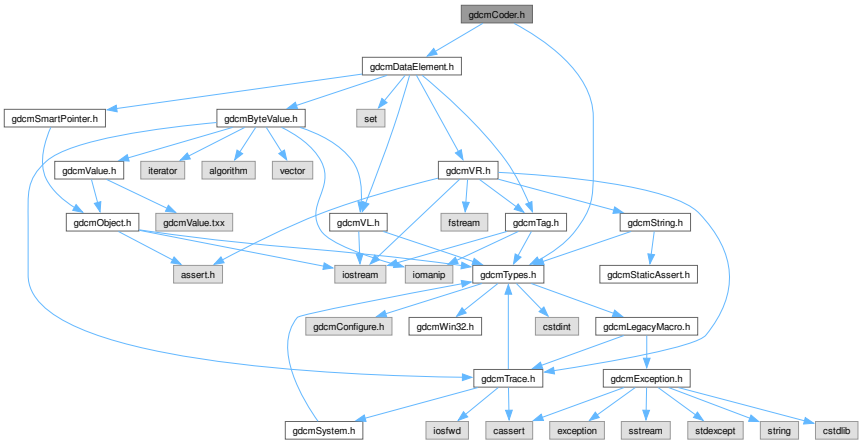
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

```
     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCODEC_H
#define GDCMCODEC_H

#include "gdcmCoder.h"
#include "gdcmDecoder.h"

namespace gdcm
{

class GDCM_EXPORT Codec : public Coder, public Decoder
{
};

} // end namespace gdcm

#endif //GDCMCODEC_H
```

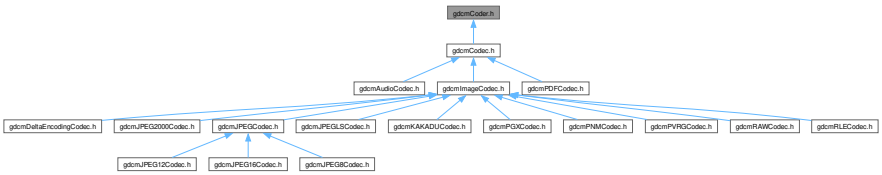
13.261 gdcmCoder.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmCoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Coder

  Coder.

Namespaces

- namespace gdcm

13.262 gdcmCoder.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCODER_H
#define GDCMCODER_H

#include "gdcmTypes.h"
#include "gdcmDataElement.h" // FIXME

namespace gdcm
{

class TransferSyntax;
class DataElement;
class GDCM_EXPORT Coder
{
public:
  virtual ~Coder() = default;

  virtual bool CanCode(TransferSyntax const &) const = 0;

  // Note: in / out are reserved keyword in C#. Change to in_ / out_

  virtual bool Code(DataElement const &in_, DataElement &out_) { (void)in_; (void)out_; return false; }
protected:
  virtual bool InternalCode(const char *bv, unsigned long len, std::ostream &os) {
(void)bv;(void)os;(void)len;return false; }
};

} // end namespace gdcm

#endif //GDCMCODER_H
```

13.263 gdcmConstCharWrapper.h File Reference

Classes

- class gdcm::ConstCharWrapper

  Do not use me.

Namespaces

- namespace gdcm

13.264 gdcmConstCharWrapper.h

Go to the documentation of this file.

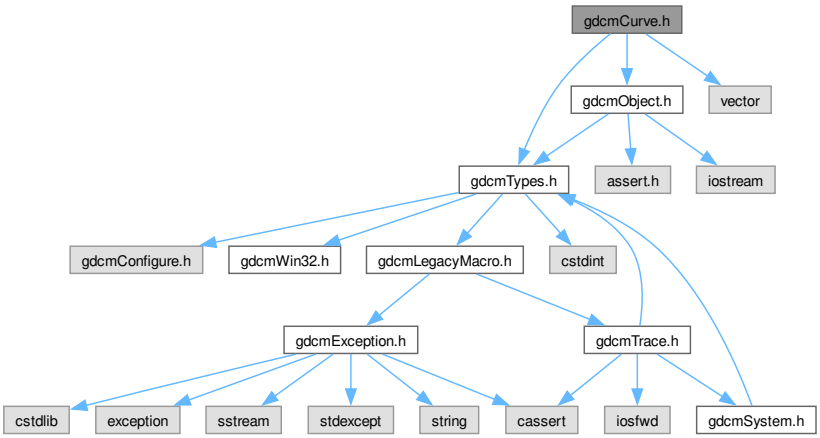
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCONSTCHARWRAPPER_H
#define GDCMCONSTCHARWRAPPER_H

namespace gdcm
{

#error

/*
 * This class is a pure hack. Its only goal is to work around a bad bug in :
 *  $ swig -version
 *  SWIG Version 1.3.31
 *
 *  See
 * - http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum
 *
 *  As a side note there is also a problem with const reference to enum type:
 * - http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum
 *
 *   And to keep track of an issue with swig here is the last one:
 *
 * - http://sourceforge.net/mailarchive/forum.php?thread_name=bf0c3b3f0802290552y5163989t76572b80a044ce28%40mail.gmail.com&forum
 */


class ConstCharWrapper
{
public:
  ConstCharWrapper(const char *i=0):Internal(i) {}
  operator const char * () const { return Internal; }
private:
  const char *Internal;
};

} // end namespace gdcm

#endif //GDCMCONSTCHARWRAPPER_H
```

13.265 gdcmCurve.h File Reference

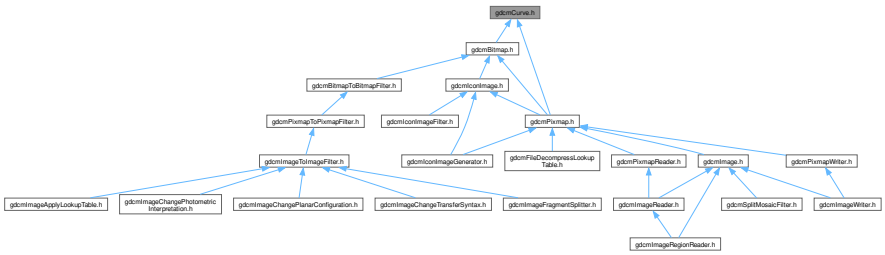
```
#include "gdcmTypes.h"
#include "gdcmObject.h"
```

```
#include <vector>
```

Include dependency graph for gdcmCurve.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Curve

  Curve class to handle element 50xx,3000 Curve Data.

Namespaces

- namespace gdcm

13.266 gdcmCurve.h

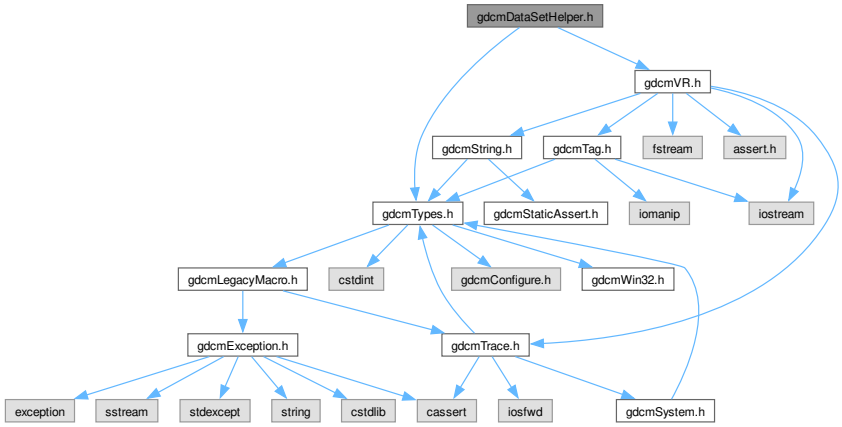
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCURVE_H
#define GDCMCURVE_H

#include "gdcmTypes.h"
#include "gdcmObject.h"

#include <vector>

namespace gdcm
{

class CurveInternal;
class ByteValue;
class DataSet;
class DataElement;
class GDCM_EXPORT Curve : public Object
{
public:
  Curve();
  ~Curve() override;
  void Print(std::ostream &) const override;

  void GetAsPoints(float *array) const;

  static unsigned int GetNumberOfCurves(DataSet const & ds);

  // Update curve data from dataelmenet de:
  void Update(const DataElement & de);

  void SetGroup(unsigned short group);
  unsigned short GetGroup() const;
  void SetDimensions(unsigned short dimensions);
  unsigned short GetDimensions() const;
  void SetNumberOfPoints(unsigned short numberofpoints);
  unsigned short GetNumberOfPoints() const;
  void SetTypeOfData(const char *typeofdata);
  const char *GetTypeOfData() const;
  // See PS 3.3 - 2004 - C.10.2.1.1 Type of data
  const char *GetTypeOfDataDescription() const;
  void SetCurveDescription(const char *curvedescription);
  void SetDataValueRepresentation(unsigned short datavaluerepresentation);
  unsigned short GetDataValueRepresentation() const;
  void SetCurveDataDescriptor(const uint16_t * values, size_t num);
  std::vector<unsigned short> const &GetCurveDataDescriptor() const;
  void SetCoordinateStartValue( unsigned short v );
  void SetCoordinateStepValue( unsigned short v );

  void SetCurve(const char *array, unsigned int length);

  bool IsEmpty() const;

  void Decode(std::istream &is, std::ostream &os);

  Curve(Curve const &ov);
private:
  double ComputeValueFromStartAndStep(unsigned int idx) const;
  CurveInternal *Internal;
};

} // end namespace gdcm

#endif //GDCMCURVE_H
```

13.267 gdcmDataSetHelper.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
```

Include dependency graph for gdcmDataSetHelper.h:



Classes

- class gdcm::DataSetHelper

  DataSetHelper (internal class, not intended for user level).

Namespaces

- namespace gdcm

13.268 gdcmDataSetHelper.h

Go to the documentation of this file.

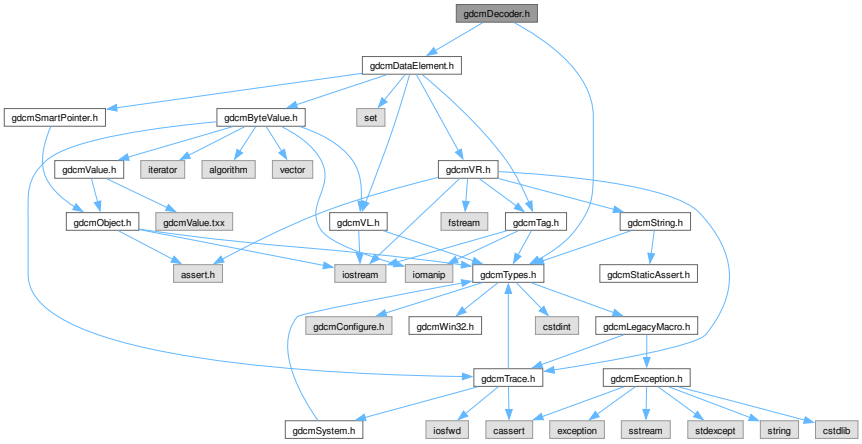
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDATASETHELPER_H
#define GDCMDATASETHELPER_H

#include "gdcmTypes.h"
```

```
#include "gdcmVR.h"

namespace gdcm
{
class DataSet;
class File;
class Tag;
class SequenceOfItems;

class GDCM_EXPORT DataSetHelper
{
public:
  static VR ComputeVR(File const & file, DataSet const &ds, const Tag& tag);

  //static SequenceOfItems* ComputeSQFromByteValue(File const & file, DataSet const &ds, const Tag &tag);

protected:
};

} // end namespace gdcm

#endif // GDCMDATASETHELPER_H
```

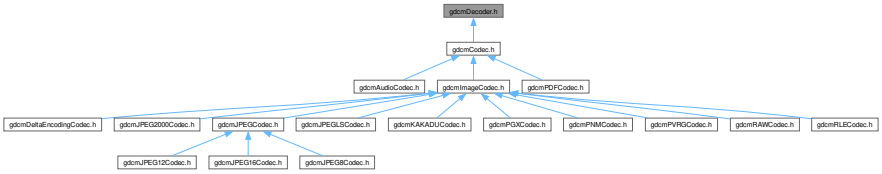
13.269 gdcmDecoder.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmDecoder.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Decoder

  Decoder.

Namespaces

- namespace gdcm

13.270 gdcmDecoder.h

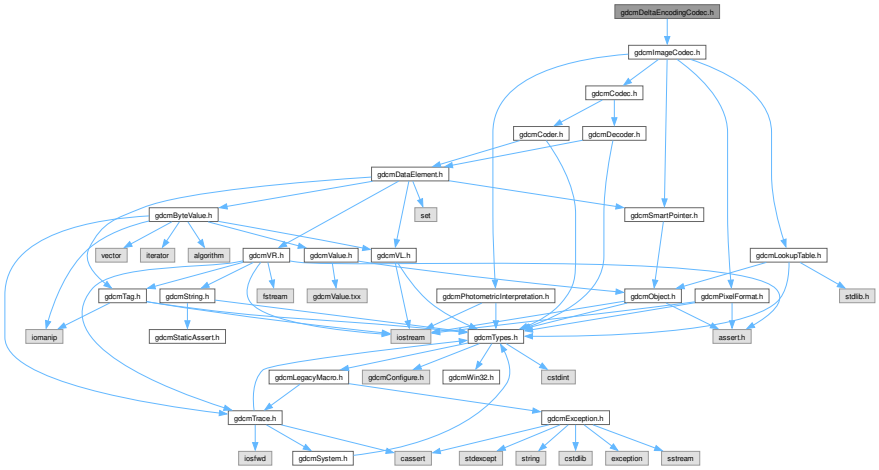
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMDECODER_H
#define GDCMDECODER_H

#include "gdcmTypes.h"
#include "gdcmDataElement.h" // FIXME

namespace gdcm
{

class TransferSyntax;
class DataElement;
class GDCM_EXPORT Decoder
{
public:
  virtual ~Decoder() = default;

  virtual bool CanDecode(TransferSyntax const &) const = 0;

  virtual bool Decode(DataElement const &, DataElement &) { return false; }
protected:
  virtual bool DecodeByStreams(std::istream &, std::ostream &) { return false; }
};

} // end namespace gdcm

#endif //GDCMDECODER_H
```

13.271 gdcmDeltaEncodingCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmDeltaEncodingCodec.h:



Classes

- class gdcm::DeltaEncodingCodec

  DeltaEncodingCodec compression used by some private vendor.

Namespaces

- namespace gdcm

13.272 gdcmDeltaEncodingCodec.h

Go to the documentation of this file.

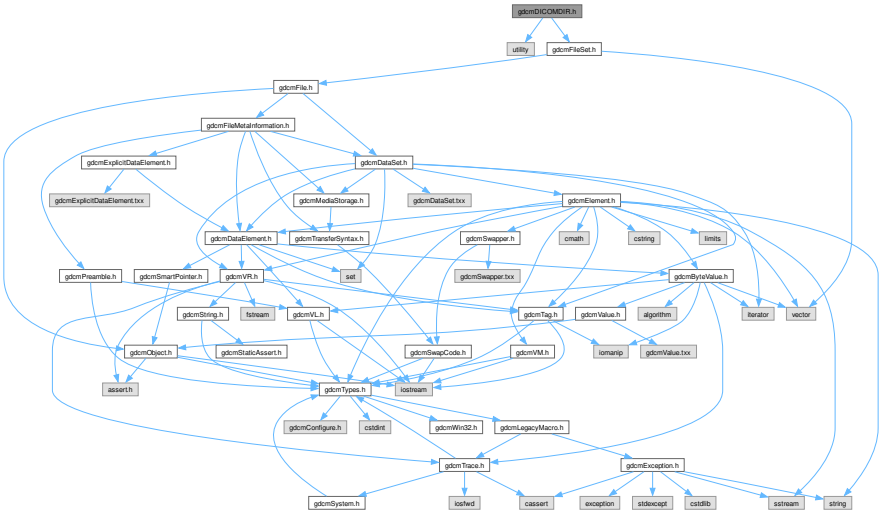
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDELTAENCODINGCODEC_H
#define GDCMDELTAENCODINGCODEC_H

#include "gdcmImageCodec.h"
#error do not use

```

```
namespace gdcm
{

class DeltaEncodingCodec : public ImageCodec
{
public:
  DeltaEncodingCodec();
  ~DeltaEncodingCodec();
  bool CanDecode(TransferSyntax const &ts);
  bool Decode(DataElement const &is, DataElement &os);
protected:
  bool Decode(std::istream &is, std::ostream &os);
};

} // end namespace gdcm

#endif //GDCMDELTAENCODINGCODEC_H
```

13.273 gdcmDICOMDIR.h File Reference

```
#include <utility>
#include "gdcmFileSet.h"
```

Include dependency graph for gdcmDICOMDIR.h:



Classes

- class gdcm::DICOMDIR

  DICOMDIR class.

Namespaces

- namespace gdcm

13.274 gdcmDICOMDIR.h

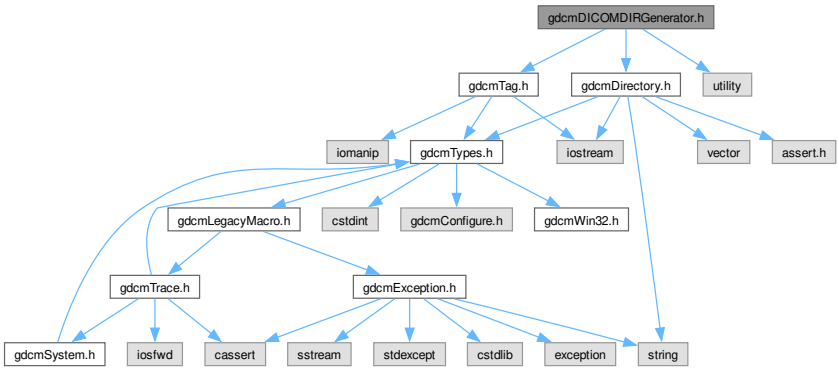
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDICOMDIR_H
#define GDCMDICOMDIR_H

#include <utility>
#include "gdcmFileSet.h"

namespace gdcm
{
class GDCM_EXPORT DICOMDIR
{
public:
  DICOMDIR() = default;
  DICOMDIR(FileSet  fs):_FS(std::move(std::move(fs))) {}

private:
  FileSet _FS;
  //13 sept 2010 mmr-- added the underscore to FS to compile under Sunos gcc
};

} // end namespace gdcm

#endif //GDCMDICOMDIR_H
```

13.275 gdcmDICOMDIRGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <utility>
```

Include dependency graph for gdcmDICOMDIRGenerator.h:



Classes

- class gdcm::DICOMDIRGenerator

  DICOMDIRGenerator class.

Namespaces

- namespace gdcm

13.276 gdcmDICOMDIRGenerator.h

Go to the documentation of this file.

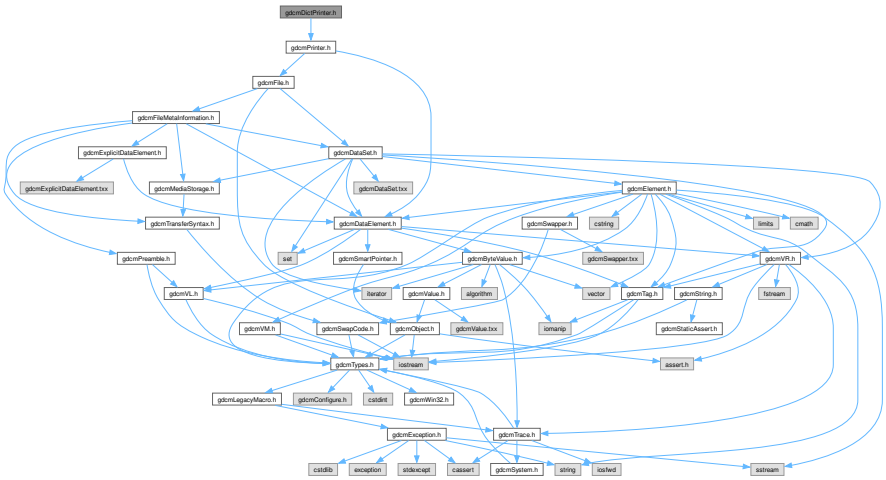
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMDICOMDIRGENERATOR_H
00015 #define GDCMDICOMDIRGENERATOR_H
00016
00017 #include "gdcmDirectory.h"
00018 #include "gdcmTag.h"
00019 #include <utility> // std::pair
00020
00021 namespace gdcm
00022 {
00023 class File;
00024 class Scanner;
00025 class SequenceOfItems;
00026 class VL;
00027 class DICOMDIRGeneratorInternal;
00028
00056 class GDCM_EXPORT DICOMDIRGenerator
00057 {
00058 public:
00059   typedef Directory::FilenamesType  FilenamesType;
00060   typedef Directory::FilenameType  FilenameType;
00061   DICOMDIRGenerator();
00062   ~DICOMDIRGenerator();
00063
00065   void SetFilenames( FilenamesType const & fns );
00066
00068   void SetRootDirectory( FilenameType const & root );
00069
00072   void SetDescriptor( const char *d );
00073
00075   bool Generate();
00076
00078   void SetFile(const File& f);
00079   File &GetFile();
00080
00081 protected:
00082   Scanner &GetScanner();
00083   bool AddPatientDirectoryRecord();
00084   bool AddStudyDirectoryRecord();
00085   bool AddSeriesDirectoryRecord();
00086   bool AddImageDirectoryRecord();
00087
00088 private:
00089   const char *ComputeFileID(const char *);
00090   bool TraverseDirectoryRecords(VL start );
```

```
  bool ComputeDirectoryRecordsOffset(const SequenceOfItems *sqi, VL start);
  size_t FindNextDirectoryRecord( size_t item1, const char *directorytype );
  SequenceOfItems *GetDirectoryRecordSequence();
  size_t FindLowerLevelDirectoryRecord( size_t item1, const char *directorytype );
  typedef std::pair< std::string, Tag> MyPair;
  MyPair GetReferenceValueForDirectoryType(size_t item);
  bool SeriesBelongToStudy(const char *seriesuid, const char *studyuid);
  bool ImageBelongToSeries(const char *sopuid, const char *seriesuid, Tag const &t1, Tag const &t2);
  bool ImageBelongToSameSeries(const char *sopuid, const char *seriesuid, Tag const &t);

  DICOMDIRGeneratorInternal * Internals;
};


} // end namespace gdcm

#endif //GDCMDICOMDIRGENERATOR_H
```

13.277 gdcmDictPrinter.h File Reference

`#include "gdcmPrinter.h"`

Include dependency graph for gdcmDictPrinter.h:



Classes

- class gdcm::DictPrinter

  DictPrinter class.

Namespaces

- namespace gdcm

13.278 gdcmDictPrinter.h

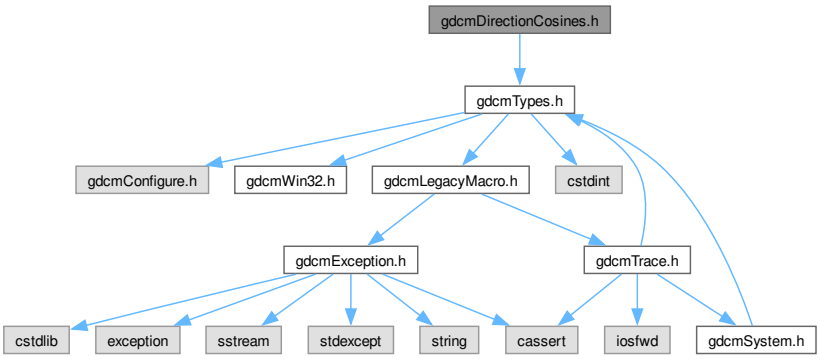
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDICTPRINTER_H
#define GDCMDICTPRINTER_H

#include "gdcmPrinter.h"

namespace gdcm
{

// It's a sink there is no output
class GDCM_EXPORT DictPrinter : public Printer
{
public:
  DictPrinter();
  ~DictPrinter() = default;

  void Print(std::ostream& os);

protected:
  void PrintDataElement2(std::ostream& os, const DataSet &ds, const DataElement &ide);
  void PrintDataSet2(std::ostream& os, const DataSet &ds);
};

} // end namespace gdcm

#endif //GDCMDICTPRINTER_H
```

13.279 gdcmDirectionCosines.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmDirectionCosines.h:



Classes

- class gdcm::DirectionCosines

  class to handle DirectionCosines

Namespaces

- namespace gdcm

13.280 gdcmDirectionCosines.h

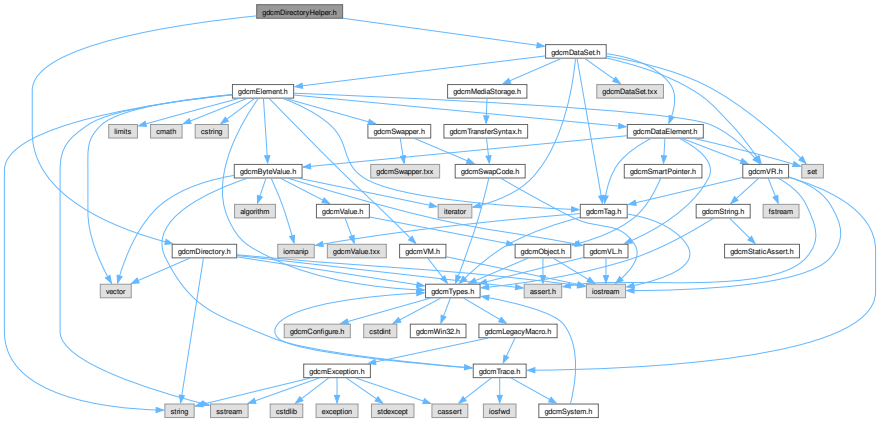
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDIRECTIONCOSINES_H
#define GDCMDIRECTIONCOSINES_H

#include "gdcmTypes.h"

namespace gdcm
{

class GDCM_EXPORT DirectionCosines
{
public:
  DirectionCosines();
  DirectionCosines(const double dircos[6]);
  // Cannot get the following signature to be wrapped with swig...
  //DirectionCosines(const double *dircos = 0 );
  ~DirectionCosines();

  void Print(std::ostream &) const;

  void Cross(double z[3]) const;

  double Dot() const;

  static double Dot(const double x[3], const double y[3]);

  void Normalize();

  static void Normalize(double v[3]);

  static double Norm(const double v[3]);

  operator const double* () const { return Values; }

  bool IsValid() const;

  bool SetFromString(const char *str);

  double CrossDot(DirectionCosines const &dc) const;

  double ComputeDistAlongNormal(const double ipp[3]) const;

private:
  double Values[6];
};

} // end namespace gdcm

#endif //GDCMDIRECTIONCOSINES_H
```

13.281 gdcmDirectoryHelper.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDirectoryHelper.h:



Classes

- class gdcm::DirectoryHelper

  DirectoryHelper.

Namespaces

- namespace gdcm

13.282 gdcmDirectoryHelper.h

Go to the documentation of this file.

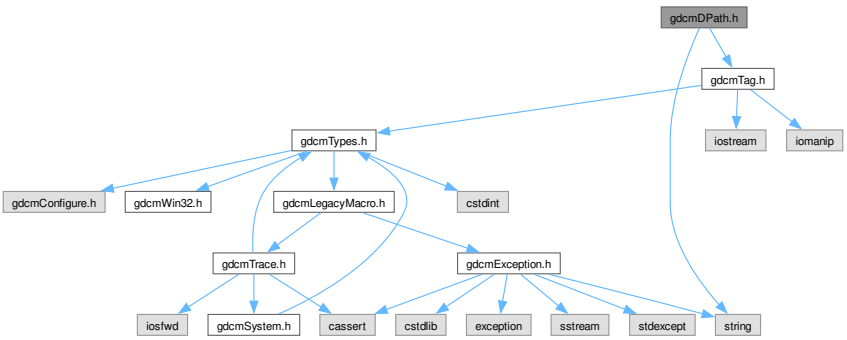
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#include "gdcmDirectory.h"
#include "gdcmDataSet.h"

namespace gdcm
{
```

```
00020
00035 class GDCM_EXPORT DirectoryHelper
00036 {
00037 public:
00038   //returns all series UIDs in a given directory that match a particular SOP Instance UID
00039   static Directory::FilenamesType GetSeriesUIDsBySOPClassUID(const std::string& inDirectory,
00040     const std::string& inSOPClassUID);
00041
00042   //specific implementations of the SOPClassUID grabber, so you don't have to
00043   //remember the SOP Class UIDs of CT or MR images.
00044   static Directory::FilenamesType GetCTImageSeriesUIDs(const std::string& inDirectory);
00045   static Directory::FilenamesType GetMRImageSeriesUIDs(const std::string& inDirectory);
00046   static Directory::FilenamesType GetRTStructSeriesUIDs(const std::string& inDirectory);
00047
00048   //given a directory and a series UID, provide all filenames with that series UID.
00049   static Directory::FilenamesType GetFilenamesFromSeriesUIDs(const std::string& inDirectory,
00050     const std::string& inSeriesUID);
00051
00052   //given a series UID, load all the images associated with that series UID
00053   //these images will be IPP sorted, so that they can be used for gathering all
00054   //the necessary information for generating an RTStruct
00055   //this function should be called by the writer once, if the writer's dataset
00056   //vector is empty.  Make sure to have a new writer for new rtstructs.
00057   static std::vector<DataSet> LoadImageFromFiles(const std::string& inDirectory,
00058     const std::string& inSeriesUID);
00059
00060   //When writing RTStructs, each contour will have z position defined.
00061   //use that z position to determine the SOPInstanceUID for that plane.
00062   static std::string RetrieveSOPInstanceUIDFromZPosition(double inZPos,
00063     const std::vector<DataSet>& inDS);
00064
00065   //When writing RTStructs, the frame of reference is done by planes to start with
00066   static std::string RetrieveSOPInstanceUIDFromIndex(int inIndex,
00067    const std::vector<DataSet>& inDS);
00068
00069   //each plane needs to know the SOPClassUID, and that won't change from image to image
00070   //so, retrieve this once at the start of writing.
00071   static std::string GetSOPClassUID(const std::vector<DataSet>& inDS);
00072
00073   //retrieve the frame of reference from the set of datasets
00074   static std::string GetFrameOfReference(const std::vector<DataSet>& inDS);
00075
00076   //both the image and polydata readers use these functions to get std::strings
00077   static std::string GetStringValueFromTag(const Tag& t, const DataSet& ds);
00078 };
00079
00080 }
```

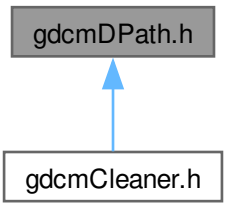
13.283 gdcmDPath.h File Reference

```
#include "gdcmTag.h"
#include <string>
```

Include dependency graph for gdcmDPath.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::DPath

  class to handle a DICOM path While supp 118 did introduced a notion of XPath for XML Native model this convention is too XML-centric. Instead prefer DCMTK style notation `https://groups.google.com/g/comp.protocols.↩dicom/c/IyIH0IOBMPA`

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DPath &val)

13.284 gdcmDPath.h

Go to the documentation of this file.

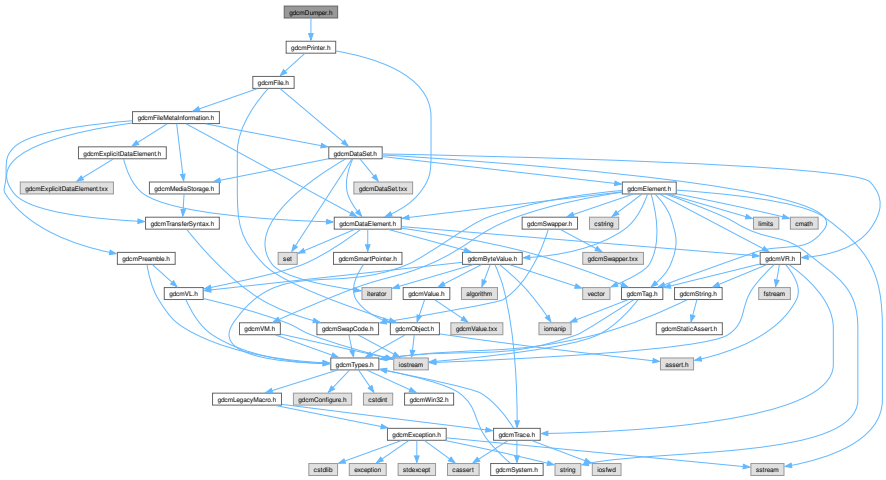
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDPATH_H
#define GDCMDPATH_H

#include "gdcmTag.h"
#include <string>

```

```
namespace gdcm {

class GDCM_EXPORT DPath {
  friend std::ostream &operator«(std::ostream &_os, const DPath &_val);

 public:
  DPath();
  ~DPath();
  void Print(std::ostream &) const;
  bool operator<(const DPath &rhs) const;

  bool ConstructFromString(const char *path);

  bool Match(DPath const &other) const;

  static bool IsValid(const char *path);

 private:
  std::string Path;
};

inline std::ostream &operator«(std::ostream &os, const DPath &val) {
  os « val.Path;
  return os;
}

}  // end namespace gdcm

#endif  // GDCMDPATH_H
```

13.285 gdcmDumper.h File Reference

`#include "gdcmPrinter.h"`

Include dependency graph for gdcmDumper.h:



Classes

- class gdcm::Dumper

  Codec class.

Namespaces

- namespace gdcm

13.286 gdcmDumper.h

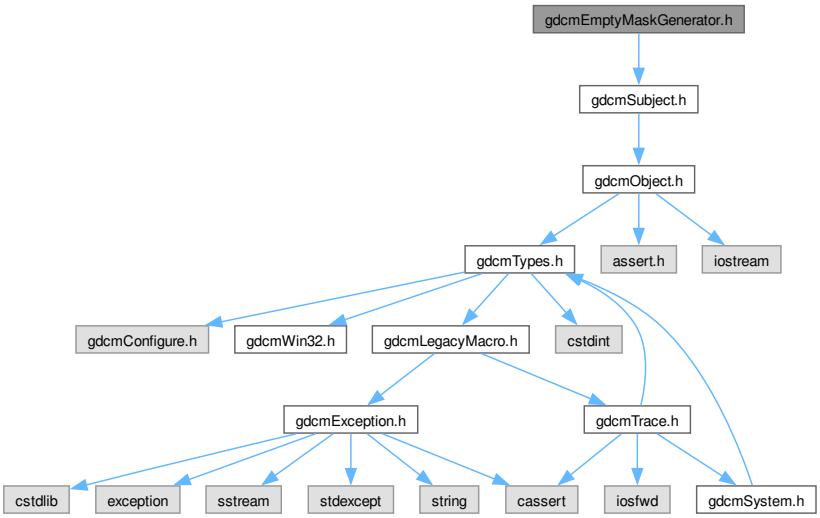
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDUMPER_H
#define GDCMDUMPER_H

#include "gdcmPrinter.h"

namespace gdcm
{

// It's a sink there is no output
class GDCM_EXPORT Dumper : public Printer
{
public:
  Dumper() { PrintStyle = CONDENSED_STYLE; }
  ~Dumper() = default;
};

} // end namespace gdcm

#endif //GDCMDUMPER_H
```

13.287 gdcmEmptyMaskGenerator.h File Reference

`#include "gdcmSubject.h"`

Include dependency graph for gdcmEmptyMaskGenerator.h:



Classes

- class gdcm::EmptyMaskGenerator

  EmptyMaskGenerator Main class to generate a Empty Mask Series from an input Series. This class takes an input folder and generates a series of DICOM files in the specified output directory. This class handles multiples DICOM Series within the same input directory.

Namespaces

- namespace gdcm

13.288 gdcmEmptyMaskGenerator.h

Go to the documentation of this file.

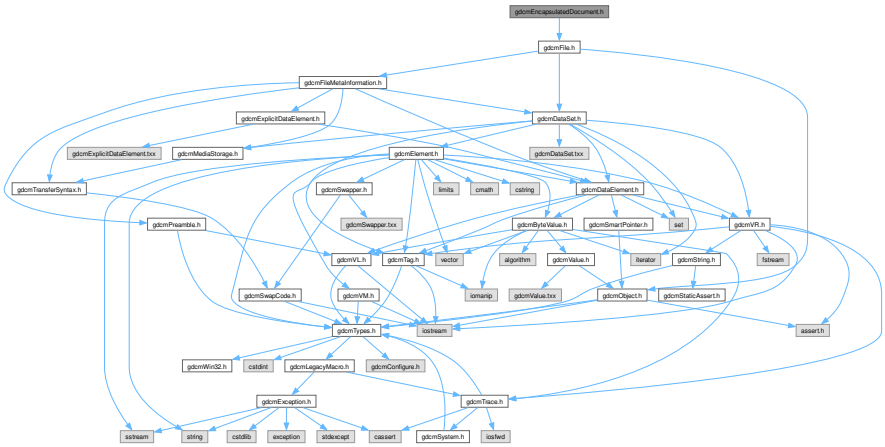
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.
```

```
00012
00013 ==========================================================================*/
00014 #ifndef GDCMEMPTYMASKGENERATOR_H
00015 #define GDCMEMPTYMASKGENERATOR_H
00016
00017 #include "gdcmSubject.h"
00018
00019 namespace gdcm {
00050 class GDCM_EXPORT EmptyMaskGenerator
00051 {
00052 public:
00053   EmptyMaskGenerator();
00054   ~EmptyMaskGenerator();
00055
00056   enum SOPClassUIDMode {
00057     UseOriginalSOPClassUID = 0, // default
00058     UseGrayscaleSecondaryImageStorage
00059   };
00060
00063   void SetSOPClassUIDMode( SOPClassUIDMode mode );
00064
00066   void SetInputDirectory( const char * dirname );
00067
00069   void SetOutputDirectory( const char * dirname );
00070
00072   bool Execute();
00073
00074 private:
00075   struct impl;
00076   // PIMPL idiom
00077   impl* pimpl;
00078 };
00079 } // end namespace gdcm
00080 #endif //GDCMEMPTYMASKGENERATOR_H
```

13.289 gdcmEncapsulatedDocument.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmEncapsulatedDocument.h:



Classes

- class gdcm::EncapsulatedDocument

  EncapsulatedDocument.

Namespaces

- namespace gdcm

13.290 gdcmEncapsulatedDocument.h

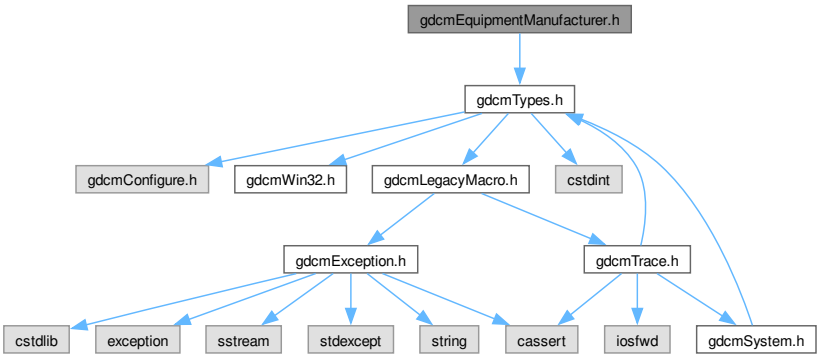
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMENCAPSULATEDDOCUMENT_H
#define GDCMENCAPSULATEDDOCUMENT_H

#include "gdcmFile.h"

namespace gdcm
{
class GDCM_EXPORT EncapsulatedDocument
{
public:
  EncapsulatedDocument() = default;

private:
};

} // end namespace gdcm

#endif //GDCMENCAPSULATEDDOCUMENT_H
```

13.291 gdcmEquipmentManufacturer.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmEquipmentManufacturer.h:



Classes

- class gdcm::EquipmentManufacturer

Namespaces

- namespace gdcm

13.292 gdcmEquipmentManufacturer.h

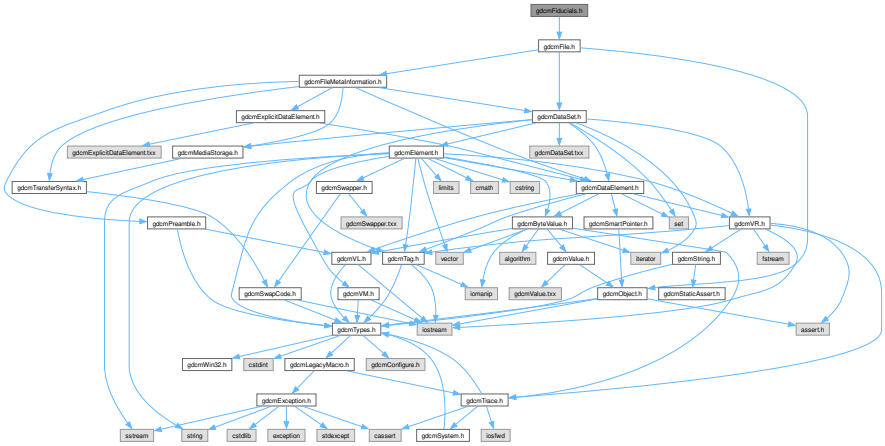
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMEQUIPMENTMANUFACTURER_H
#define GDCMEQUIPMENTMANUFACTURER_H

#include "gdcmTypes.h"

namespace gdcm {

class DataSet;
class GDCM_EXPORT EquipmentManufacturer {
 public:
  typedef enum {
    UNKNOWN = 0,
    FUJI,
    GEMS,
    HITACHI,
    KODAK,
    MARCONI,
    PMS,
    SIEMENS,
    TOSHIBA,
    AGFA,
    SAMSUNG,
    UIH
  } Type;

  static Type Compute(DataSet const &ds);

  static const char *TypeToString(Type type);

 private:
  static EquipmentManufacturer::Type GuessFromPrivateAttributes(
      DataSet const &ds);
};

}  // end namespace gdcm

#endif  // GDCMEQUIPMENTMANUFACTURER_H
```

13.293 gdcmFiducials.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmFiducials.h:



Classes

- class gdcm::Fiducials

  Fiducials.

Namespaces

- namespace gdcm

13.294 gdcmFiducials.h

Go to the documentation of this file.

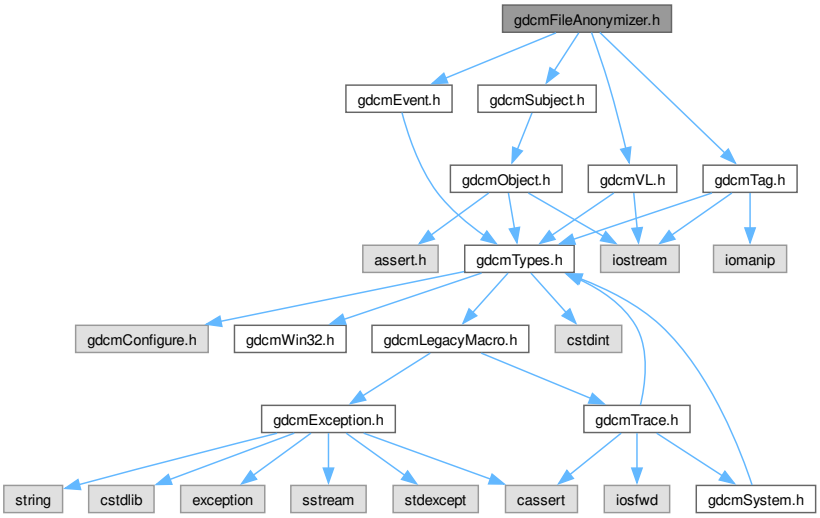
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMFIDUCIALS_H
00015 #define GDCMFIDUCIALS_H
00016
00017 #include "gdcmFile.h"
00018
00019 namespace gdcm
00020 {
```

```
class GDCM_EXPORT Fiducials
{
public:
  Fiducials() = default;

private:
};

} // end namespace gdcm

#endif //GDCMFIDUCIALS_H
```

13.295 gdcmFileAnonymizer.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmTag.h"
#include "gdcmVL.h"
```

Include dependency graph for gdcmFileAnonymizer.h:



Classes

- class gdcm::FileAnonymizer

  FileAnonymizer.

Namespaces

- namespace gdcm

13.296 gdcmFileAnonymizer.h

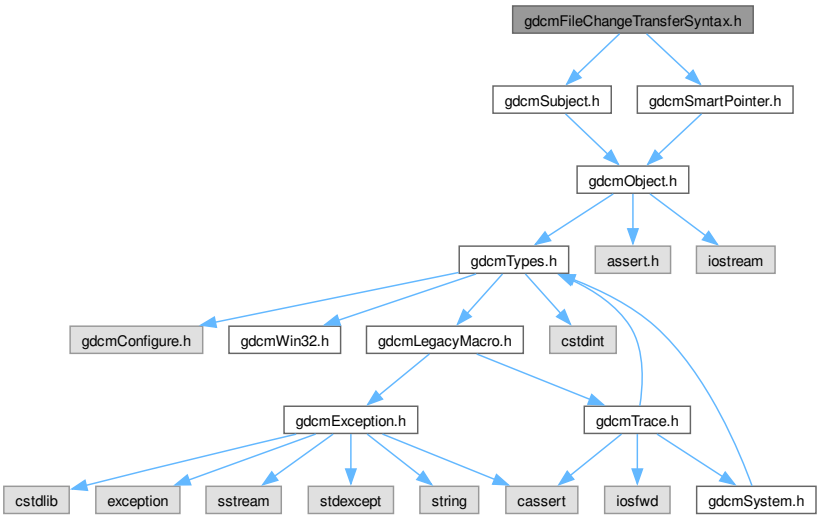
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILEANONYMIZER_H
#define GDCMFILEANONYMIZER_H

#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmTag.h"
#include "gdcmVL.h"

namespace gdcm
{
class FileAnonymizerInternals;

class GDCM_EXPORT FileAnonymizer : public Subject
{
public:
  FileAnonymizer();
  ~FileAnonymizer() override;

  void Empty( Tag const &t );

  void Remove( Tag const &t );

  void Replace( Tag const &t, const char *value_str );

  void Replace( Tag const &t, const char *value_data, VL const & vl );

  void SetInputFileName(const char *filename_native);

  void SetOutputFileName(const char *filename_native);

  bool Write();

private:
  bool ComputeEmptyTagPosition();
  bool ComputeRemoveTagPosition();
  bool ComputeReplaceTagPosition();
  FileAnonymizerInternals *Internals;
};

} // end namespace gdcm

#endif //GDCMFILEANONYMIZER_H
```

13.297 gdcmFileChangeTransferSyntax.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



Classes

- class gdcm::FileChangeTransferSyntax

  FileChangeTransferSyntax.

Namespaces

- namespace gdcm

13.298 gdcmFileChangeTransferSyntax.h

Go to the documentation of this file.

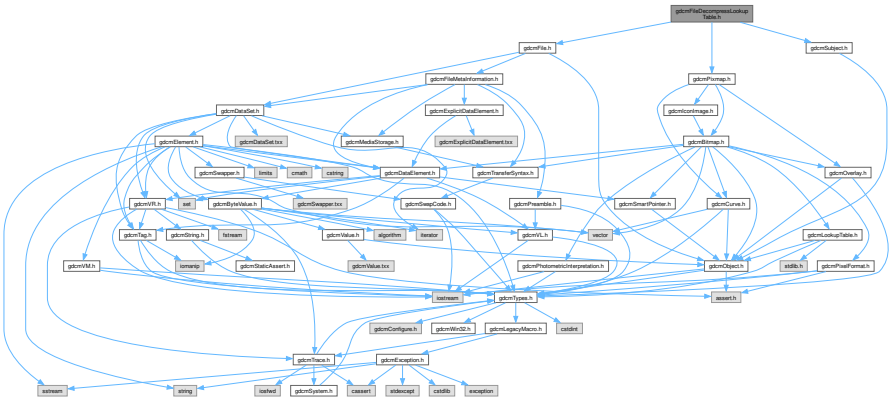
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILECHANGETRANSFERSYNTAX_H
#define GDCMFILECHANGETRANSFERSYNTAX_H

#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"

```

```
namespace gdcm
{
class FileChangeTransferSyntaxInternals;
class ImageCodec;
class TransferSyntax;

class GDCM_EXPORT FileChangeTransferSyntax : public Subject
{
public:
  FileChangeTransferSyntax();
  ~FileChangeTransferSyntax() override;

  void SetInputFileName(const char *filename_native);

  void SetOutputFileName(const char *filename_native);

  bool Change();

  void SetTransferSyntax( TransferSyntax const & ts );

  ImageCodec * GetCodec();

  static SmartPointer<FileChangeTransferSyntax> New() { return new FileChangeTransferSyntax; }

private:
  bool InitializeCopy();
  bool UpdateCompressionLevel(double level);
  FileChangeTransferSyntaxInternals *Internals;
};

} // end namespace gdcm

#endif //GDCMFILEANONYMIZER_H
```

13.299 gdcmFileDecompressLookupTable.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmFile.h"
#include "gdcmPixmap.h"
```

Include dependency graph for gdcmFileDecompressLookupTable.h:



Classes

- class gdcm::FileDecompressLookupTable

  FileDecompressLookupTable class.

Namespaces

- namespace gdcm

13.300 gdcmFileDecompressLookupTable.h

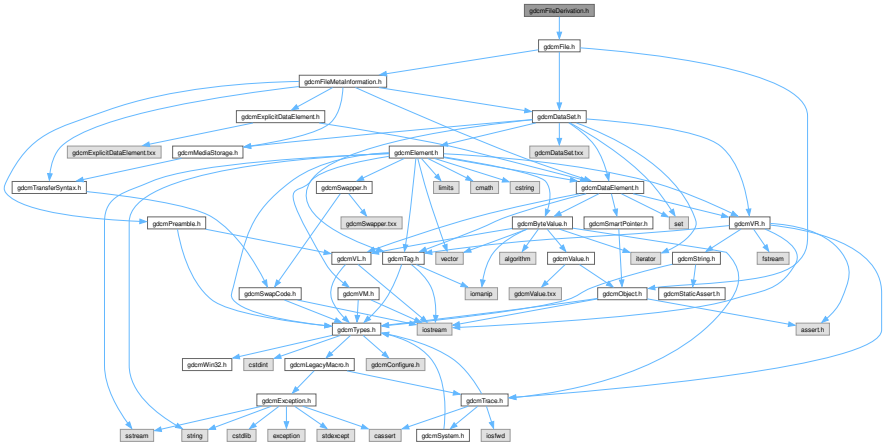
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILEDECOMPRESSLOOKUPTABLE_H
#define GDCMFILEDECOMPRESSLOOKUPTABLE_H

#include "gdcmSubject.h"
#include "gdcmFile.h"
#include "gdcmPixmap.h"

namespace gdcm
{

class DataElement;
class GDCM_EXPORT FileDecompressLookupTable : public Subject
{
public:
  FileDecompressLookupTable() = default;
  ~FileDecompressLookupTable() override = default;

  bool Change();

  void SetFile(const File& f) { F = f; }
  File &GetFile() { return *F; }

  const Pixmap& GetPixmap() const { return *PixelData; }
  Pixmap& GetPixmap() { return *PixelData; }
  void SetPixmap(Pixmap const &img) { PixelData = img; }

protected:

private:
  SmartPointer<File> F;
  SmartPointer<Pixmap> PixelData;
};

} // end namespace gdcm

#endif //GDCMFILEDECOMPRESSLOOKUPTABLE_H
```

13.301 gdcmFileDerivation.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmFileDerivation.h:



Classes

- class gdcm::FileDerivation

  FileDerivation class.

Namespaces

- namespace gdcm

13.302 gdcmFileDerivation.h

Go to the documentation of this file.

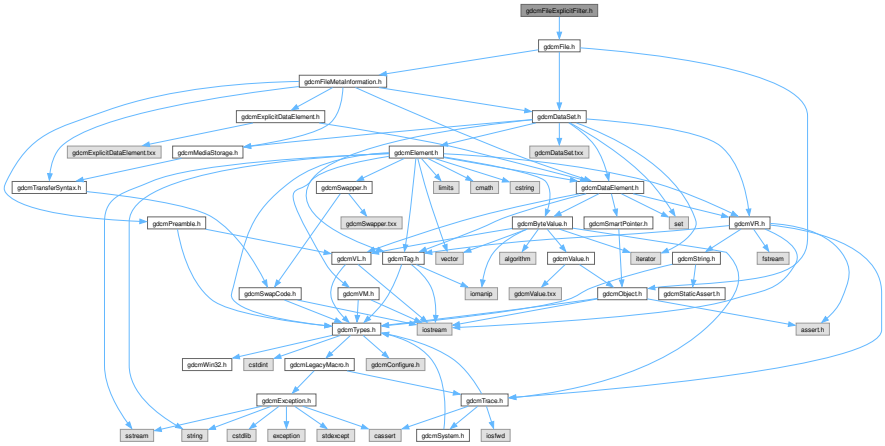
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILEDERIVATION_H
#define GDCMFILEDERIVATION_H

#include "gdcmFile.h"

namespace gdcm
{
```

```
00021
00022 class FileDerivationInternals;
00023 class DataSet;
00039 class GDCM_EXPORT FileDerivation
00040 {
00041 public:
00042   FileDerivation();
00043   ~FileDerivation();
00044
00049   bool AddReference(const char *referencedsopclassuid, const char *referencedsopinstanceuid);
00050
00051   // CID 7202 Source Image Purposes of Reference
00052   // {"DCM",121320,"Uncompressed predecessor"},
00053
00055   void SetPurposeOfReferenceCodeSequenceCodeValue(unsigned int codevalue);
00056
00057   // CID 7203 Image Derivation
00058   // { "DCM",113040,"Lossy Compression" },
00059
00061   void SetDerivationCodeSequenceCodeValue(unsigned int codevalue);
00062
00064   void SetDerivationDescription( const char *dd );
00065
00069   void SetAppendDerivationHistory(bool b);
00070
00072   bool Derive();
00073
00075   void SetFile(const File& f) { F = f; }
00076   File &GetFile() { return *F; }
00077   const File &GetFile() const { return *F; }
00078
00079 protected:
00080   bool AddDerivationDescription();
00081   bool AddSourceImageSequence();
00082   bool AddPurposeOfReferenceCodeSequence(DataSet &ds);
00083
00084 private:
00085   SmartPointer<File> F;
00086   FileDerivationInternals *Internals;
00087 };
00088
00094
00095
00096 } // end namespace gdcm
00097
00098 #endif //GDCMFILEDERIVATION_H
```

13.303 gdcmFileExplicitFilter.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmFileExplicitFilter.h:



Classes

- class gdcm::FileExplicitFilter

  FileExplicitFilter class.

Namespaces

- namespace gdcm

13.304 gdcmFileExplicitFilter.h

Go to the documentation of this file.

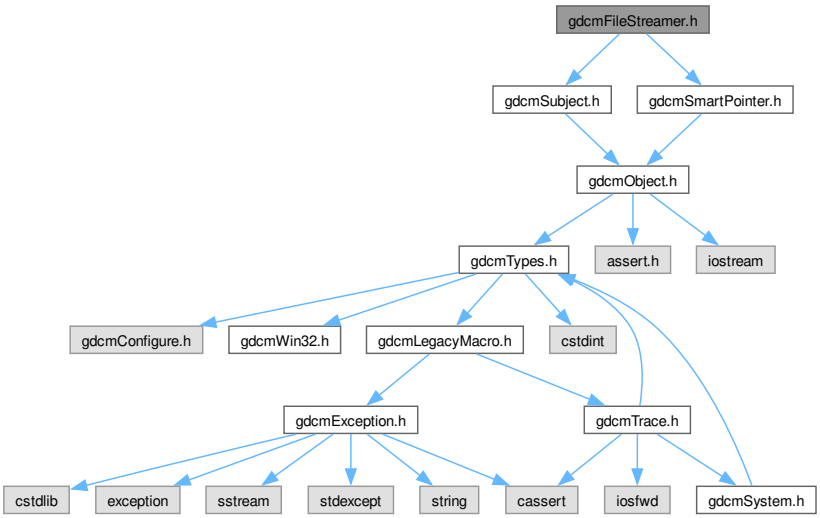
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMFILEEXPLICITFILTER_H
00015 #define GDCMFILEEXPLICITFILTER_H
00016
00017 #include "gdcmFile.h"
00018
00019 namespace gdcm
00020 {
```

```
class Dicts;

class GDCM_EXPORT FileExplicitFilter
{
public:
  FileExplicitFilter():F(new File),ChangePrivateTags(false),UseVRUN(true),RecomputeItemLength(false),RecomputeSequenceLength(false) {}
  ~FileExplicitFilter() = default;

  void SetChangePrivateTags(bool b) { ChangePrivateTags = b;}

  void SetUseVRUN(bool b) { UseVRUN = b; }

  void SetRecomputeItemLength(bool b);
  void SetRecomputeSequenceLength(bool b);


  bool Change();

  void SetFile(const File& f) { F = f; }
  File &GetFile() { return *F; }

protected:
  bool ProcessDataSet(DataSet &ds, Dicts const & dicts);
  bool ChangeFMI();

private:
  SmartPointer<File> F;
  bool ChangePrivateTags;
  bool UseVRUN;
  bool RecomputeItemLength;
  bool RecomputeSequenceLength;
};


} // end namespace gdcm

#endif //GDCMFILEEXPLICITFILTER_H
```

13.305 gdcmFileStreamer.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileStreamer.h:



Classes

- class gdcm::FileStreamer

  FileStreamer.

Namespaces

- namespace gdcm

13.306 gdcmFileStreamer.h

Go to the documentation of this file.

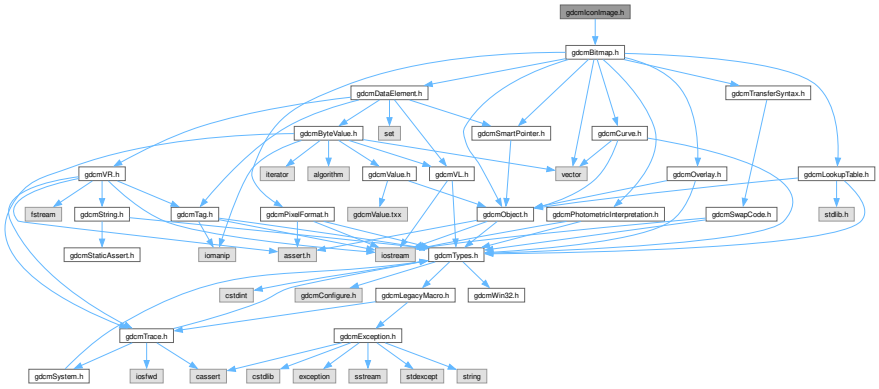
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFILESTREAMER_H
#define GDCMFILESTREAMER_H

#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"

```

```
namespace gdcm
{
class FileStreamerInternals;

class Tag;
class PrivateTag;
class GDCM_EXPORT FileStreamer : public Subject
{
public:
  FileStreamer();
  ~FileStreamer() override;

  void SetTemplateFileName(const char *filename_native);

  // Decide to check template or not (default: false)
  void CheckTemplateFileName(bool check);

  void SetOutputFileName(const char *filename_native);

  bool CheckDataElement( const Tag & t );

  bool StartDataElement( const Tag & t );
  bool AppendToDataElement( const Tag & t, const char *array, size_t len );
  bool StopDataElement( const Tag & t );
  bool ReserveDataElement( size_t len );

  bool StartGroupDataElement( const PrivateTag & pt, size_t maxsizede = 0, uint8_t startoffset = 0 );
  bool AppendToGroupDataElement( const PrivateTag & pt, const char *array, size_t len );
  bool StopGroupDataElement( const PrivateTag & pt );
  bool ReserveGroupDataElement( unsigned short ndataelement );

  static SmartPointer<FileStreamer> New() { return new FileStreamer; }

private:
  bool InitializeCopy();
  FileStreamerInternals *Internals;
};

} // end namespace gdcm

#endif //GDCMFILESTREAMER_H
```

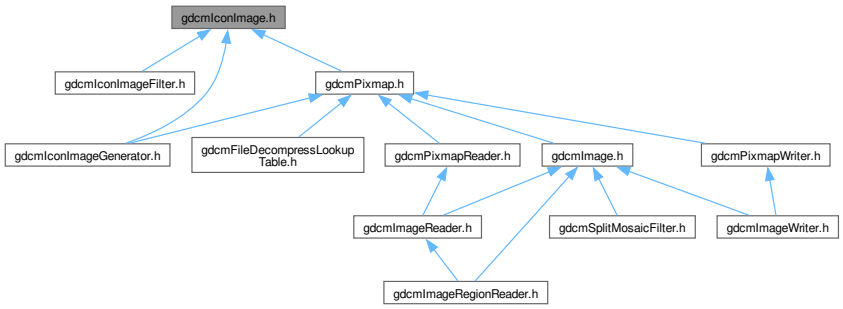
13.307 gdcmIconImage.h File Reference

`#include "gdcmBitmap.h"`

Include dependency graph for gdcmIconImage.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace gdcm

Typedefs

- typedef Bitmap gdcm::IconImage

13.308 gdcmIconImage.h

Go to the documentation of this file.

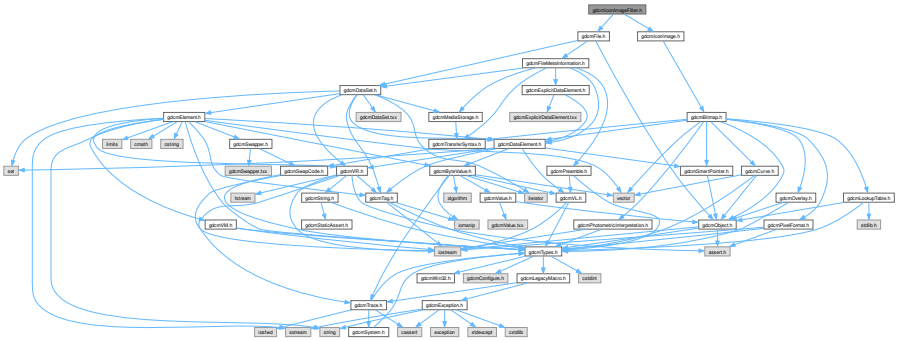
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMICONIMAGE_H
#define GDCMICONIMAGE_H

#if 0
#include "gdcmObject.h"
#include "gdcmDataElement.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmPixelFormat.h"
#include "gdcmTransferSyntax.h"

#include <vector>

namespace gdcm
{

class GDCM_EXPORT IconImage : public Object
{
public:
  IconImage();
```

```
  ~IconImage();
  void Print(std::ostream &) const {}

  void SetTransferSyntax(TransferSyntax const &ts) {
    TS = ts;
  }
  const TransferSyntax &GetTransferSyntax() const {
    return TS;
  }
  void SetDataElement(DataElement const &de) {
    PixelData = de;
  }
  const DataElement& GetDataElement() const { return PixelData; }

  void SetColumns(unsigned int col) { SetDimension(0,col); }
  void SetRows(unsigned int rows) { SetDimension(1,rows); }
  void SetDimension(unsigned int idx, unsigned int dim);
  int GetColumns() const { return Dimensions[0]; }
  int GetRows() const { return Dimensions[1]; }
  // Get/Set PixelFormat
  const PixelFormat &GetPixelFormat() const
    {
    return PF;
    }
  void SetPixelFormat(PixelFormat const &pf)
    {
    PF = pf;
    }

  const PhotometricInterpretation &GetPhotometricInterpretation() const;
  void SetPhotometricInterpretation(PhotometricInterpretation const &pi);

  bool IsEmpty() const { return Dimensions.size() == 0; }
  void Clear();

  bool GetBuffer(char *buffer) const;

private:
  TransferSyntax TS;
  PixelFormat PF; // SamplesPerPixel, BitsAllocated, BitsStored, HighBit, PixelRepresentation
  PhotometricInterpretation PI;
  std::vector<unsigned int> Dimensions; // Col/Row
  std::vector<double> Spacing; // PixelAspectRatio ?
  DataElement PixelData; // copied from 7fe0,0010
  static const unsigned int NumberOfDimensions = 2;
};

} // end namespace gdcm
#endif
#include "gdcmBitmap.h"

namespace gdcm
{
  //class GDCM_EXPORT IconImage : public Pixmap {};
  typedef Bitmap IconImage;
}

#endif //GDCMICONIMAGE_H
```

13.309 gdcmIconImageFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageFilter.h:



Classes

- class gdcm::IconImageFilter

  IconImageFilter.

Namespaces

- namespace gdcm

13.310 gdcmIconImageFilter.h

Go to the documentation of this file.

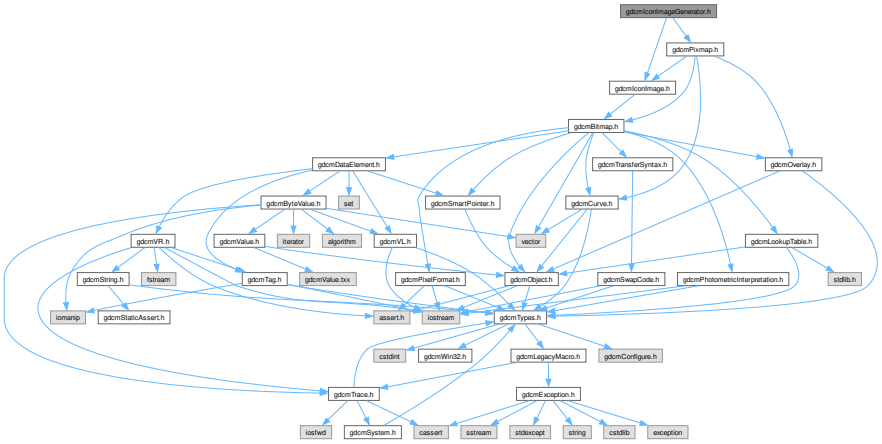
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMICONIMAGEFILTER_H
#define GDCMICONIMAGEFILTER_H

#include "gdcmFile.h"
#include "gdcmIconImage.h"

namespace gdcm
{
class IconImageFilterInternals;

class GDCM_EXPORT IconImageFilter
{
public:
  IconImageFilter();
  ~IconImageFilter();

  void SetFile(const File& f) { F = f; }
  File &GetFile() { return *F; }
```

```
00058   const File &GetFile() const { return *F; }
00059
00061   bool Extract();
00062
00064   unsigned int GetNumberOfIconImages() const;
00065   IconImage& GetIconImage( unsigned int i ) const;
00066
00067 protected:
00068   void ExtractIconImages();
00069   void ExtractVeproIconImages();
00070
00071 private:
00072   SmartPointer<File> F;
00073   IconImageFilterInternals *Internals;
00074 };
00075
00076 } // end namespace gdcm
00077
00078 #endif //GDCMICONIMAGEFILTER_H
```

13.311 gdcmIconImageGenerator.h File Reference

```
#include "gdcmPixmap.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageGenerator.h:



Classes

- class gdcm::IconImageGenerator

  IconImageGenerator.

Namespaces

- namespace gdcm

13.312 gdcmIconImageGenerator.h

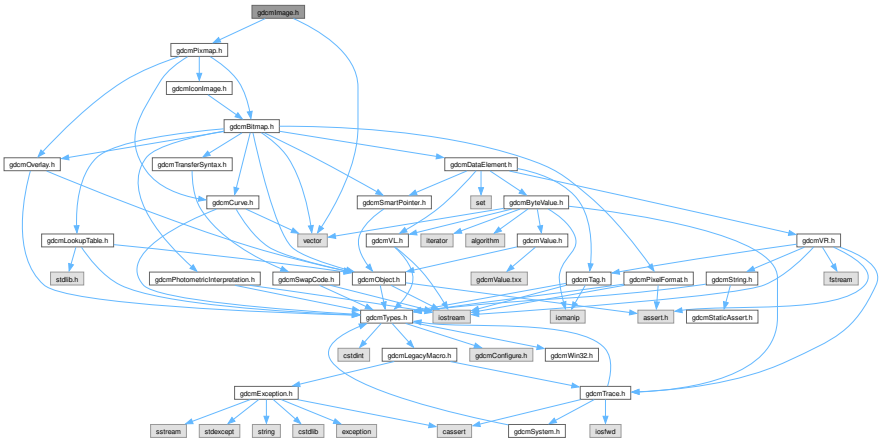
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMICONIMAGEGENERATOR_H
#define GDCMICONIMAGEGENERATOR_H

#include "gdcmPixmap.h"
#include "gdcmIconImage.h"

namespace gdcm
{
class IconImageGeneratorInternals;
class GDCM_EXPORT IconImageGenerator
{
public:
  IconImageGenerator();
  ~IconImageGenerator();

  void SetPixmap(const Pixmap& p) { P = p; }
  Pixmap &GetPixmap() { return *P; }
  const Pixmap &GetPixmap() const { return *P; }

  void SetOutputDimensions(const unsigned int dims[2]);

  void SetPixelMinMax(double min, double max);

  void AutoPixelMinMax(bool b);

  void ConvertRGBToPaletteColor(bool b);

  void SetOutsideValuePixel(double v);

  bool Generate();

  const IconImage& GetIconImage() const { return *I; }

protected:

private:
  void BuildLUT( Bitmap & bitmap, unsigned int maxcolor );

  SmartPointer<Pixmap> P;
  SmartPointer<IconImage> I;
  IconImageGeneratorInternals *Internals;
};

} // end namespace gdcm

#endif //GDCMICONIMAGEGENERATOR_H
```

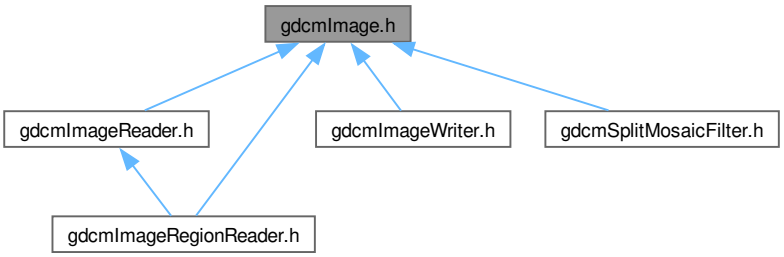
13.313 gdcmImage.h File Reference

```
#include "gdcmPixmap.h"
#include <vector>
```

Include dependency graph for gdcmImage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Image

  Image.

Namespaces

- namespace gdcm

13.314 gdcmImage.h

Go to the documentation of this file.

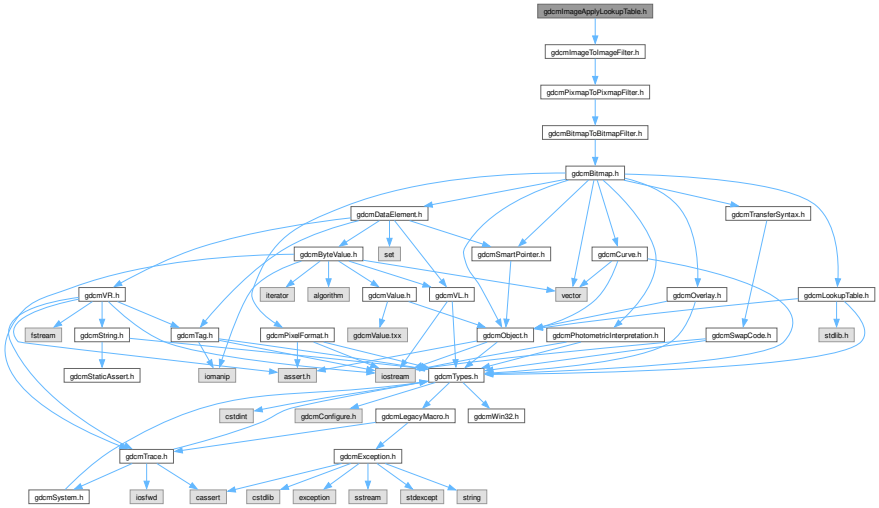
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGE_H
#define GDCMIMAGE_H

#include "gdcmPixmap.h"

#include <vector>

namespace gdcm
{

class GDCM_EXPORT Image : public Pixmap
{
public:
  Image ():Spacing(),SC(),Intercept(0),Slope(1) {
    //DirectionCosines.resize(6);
  Origin.resize( 3 /*NumberOfDimensions*/ ); // fill with 0
  DirectionCosines.resize( 6 ); // fill with 0
  DirectionCosines[0] = 1;
  DirectionCosines[4] = 1;
  Spacing.resize( 3 /*NumberOfDimensions*/, 1 ); // fill with 1

  }
  ~Image() override = default;

  const double *GetSpacing() const;
  double GetSpacing(unsigned int idx) const;
  void SetSpacing(const double spacing[3]);
  void SetSpacing(unsigned int idx, double spacing);

  const double *GetOrigin() const;
  double GetOrigin(unsigned int idx) const;
  void SetOrigin(const float origin[3]);
  void SetOrigin(const double origin[3]);
  void SetOrigin(unsigned int idx, double ori);

  const double *GetDirectionCosines() const;
  double GetDirectionCosines(unsigned int idx) const;
  void SetDirectionCosines(const float dircos[6]);
  void SetDirectionCosines(const double dircos[6]);
  void SetDirectionCosines(unsigned int idx, double dircos);

  void Print(std::ostream &os) const override;

  void SetIntercept(double intercept) { Intercept = intercept; }
  double GetIntercept() const { return Intercept; }

  void SetSlope(double slope) { Slope = slope; }
  double GetSlope() const { return Slope; }

private:
  std::vector<double> Spacing;
  std::vector<double> Origin;
  std::vector<double> DirectionCosines;

  // I believe the following 3 ivars can be derived from TS ...
  SwapCode SC;
  double Intercept;
  double Slope;
};


} // end namespace gdcm
```

```
00112
00113 #endif //GDCMIMAGE_H
```

13.315 gdcmImageApplyLookupTable.h File Reference

`#include "gdcmImageToImageFilter.h"`

Include dependency graph for gdcmImageApplyLookupTable.h:



Classes

- class gdcm::ImageApplyLookupTable

  ImageApplyLookupTable class.

Namespaces

- namespace gdcm

13.316 gdcmImageApplyLookupTable.h

Go to the documentation of this file.

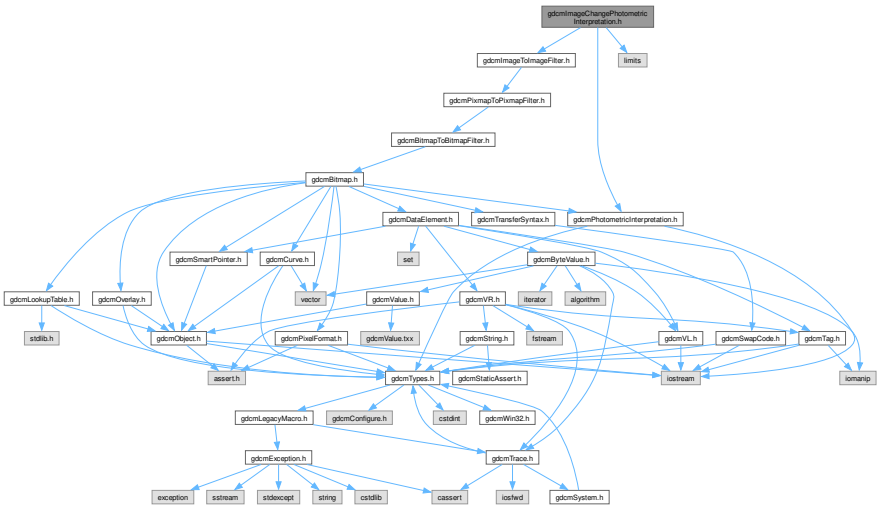
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGEAPPLYLOOKUPTABLE_H
#define GDCMIMAGEAPPLYLOOKUPTABLE_H

#include "gdcmImageToImageFilter.h"

namespace gdcm
{

class DataElement;
class GDCM_EXPORT ImageApplyLookupTable : public ImageToImageFilter
{
public:
  ImageApplyLookupTable();
  ~ImageApplyLookupTable();

  bool Apply();

  void SetRGB8(bool b);

protected:

private:
  struct impl;
  // PIMPL idiom
  impl* pimpl;
};

} // end namespace gdcm

#endif //GDCMIMAGEAPPLYLOOKUPTABLE_H
```

13.317 gdcmImageChangePhotometricInterpretation.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmPhotometricInterpretation.h"
#include <limits>
```

Include dependency graph for gdcmImageChangePhotometricInterpretation.h:



Classes

- class gdcm::ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class.

Namespaces

- namespace gdcm

Functions

- template<typename T>
  static T gdcm::Clamp (int v)
- template<typename T>
  static int gdcm::Round (T x)

13.318 gdcmImageChangePhotometricInterpretation.h

Go to the documentation of this file.

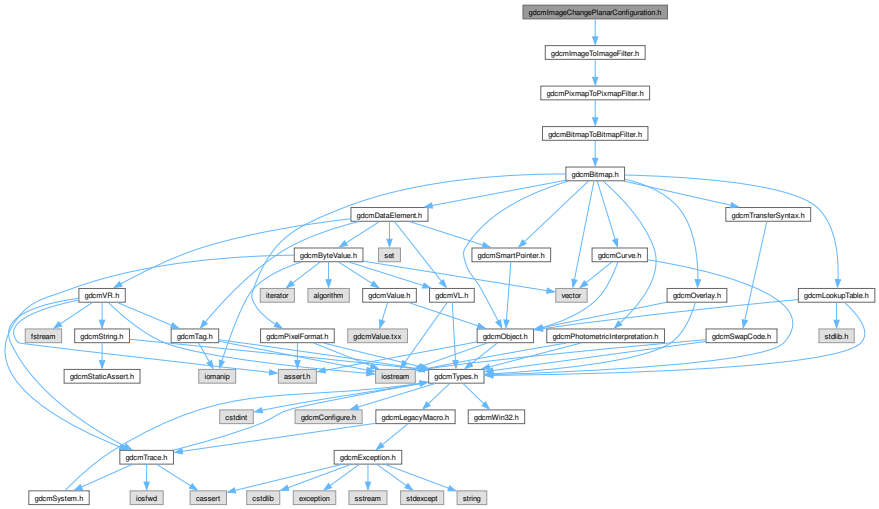
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
00015 #define GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
00016
00017 #include "gdcmImageToImageFilter.h"
00018 #include "gdcmPhotometricInterpretation.h"
00019 #include <limits>
00020
00021 namespace gdcm
00022 {
00023
00024 class DataElement;
00029 class GDCM_EXPORT ImageChangePhotometricInterpretation : public ImageToImageFilter
00030 {
00031 public:
00032   ImageChangePhotometricInterpretation():PI() {}
00033   ~ImageChangePhotometricInterpretation() = default;
00034
00036   void SetPhotometricInterpretation(PhotometricInterpretation const &pi) { PI = pi; }
00037   const PhotometricInterpretation &GetPhotometricInterpretation() const { return PI; }
00038
00040   bool Change();
00041
00044   template <typename T>
00045   static void RGB2YBR(T ybr[3], const T rgb[3], unsigned short storedbits = 8);
00046   template <typename T>
00047   static void YBR2RGB(T rgb[3], const T ybr[3], unsigned short storedbits = 8);
00048
00049 protected:
00050   bool ChangeMonochrome();
00051   bool ChangeYBR2RGB();
00052   bool ChangeRGB2YBR();
```

```
00053
00054 private:
00055   PhotometricInterpretation PI;
00056 };
00057
00058 template <typename T>
00059 static inline int Round(T x)
00060 {
00061   return (int)(x+0.5);
00062 }
00063
00064 template <typename T>
00065 static inline T Clamp(int v)
00066 {
00067   assert( std::numeric_limits<T>::min() == 0 );
00068   return v < 0 ? 0 : (v > std::numeric_limits<T>::max() ? std::numeric_limits<T>::max() : v);
00069 }
00070
00071
00072 template <typename T>
00073 void ImageChangePhotometricInterpretation::RGB2YBR(T ybr[3], const T rgb[3], unsigned short storedbits)
00074 {
00075   // Implementation details, since the equations from:
00076   // http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.7.6.3.html#sect_C.7.6.3.1.2
00077   // are rounded to the 4th decimal precision, prefer the exact equation from the original document at:
00078   // CCIR Recommendation 601-2, also found in T.871 (Section §7, page 4)
00079   const double R = rgb[0];
00080   const double G = rgb[1];
00081   const double B = rgb[2];
00082   assert( storedbits <= sizeof(T) * 8 );
00083   const int halffullscale = 1 « (storedbits - 1);
00084   const int Y  = Round(  0.299 * R + 0.587 * G + 0.114 * B                       );
00085   const int CB = Round((-0.299 * R - 0.587 * G + 0.886 * B)/1.772 + halffullscale);
00086   const int CR = Round(( 0.701 * R - 0.587 * G - 0.114 * B)/1.402 + halffullscale);
00087   ybr[0] = Clamp<T>(Y );
00088   ybr[1] = Clamp<T>(CB);
00089   ybr[2] = Clamp<T>(CR);
00090 }
00091
00092 template <typename T>
00093 void ImageChangePhotometricInterpretation::YBR2RGB(T rgb[3], const T ybr[3], unsigned short storedbits)
00094 {
00095   const double Y  = ybr[0];
00096   const double Cb = ybr[1];
00097   const double Cr = ybr[2];
00098   assert( storedbits <= sizeof(T) * 8 );
00099   const int halffullscale = 1 « (storedbits - 1);
00100   const int R = Round(Y                                       + 1.402 * (Cr-halffullscale)
      );
00101   const int G = Round(Y -( 0.114 * 1.772 * (Cb-halffullscale) + 0.299 * 1.402 *
      (Cr-halffullscale))/0.587);
00102   const int B = Round(Y          + 1.772 * (Cb-halffullscale)
      );
00103   rgb[0] = Clamp<T>(R);
00104   rgb[1] = Clamp<T>(G);
00105   rgb[2] = Clamp<T>(B);
00106 }
00107
00108 } // end namespace gdcm
00109
00110 #endif //GDCMIMAGECHANGEPHOTOMETRICINTERPRETATION_H
```

13.319 gdcmImageChangePlanarConfiguration.h File Reference

`#include "gdcmImageToImageFilter.h"`

Include dependency graph for gdcmImageChangePlanarConfiguration.h:



Classes

- class gdcm::ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class.

Namespaces

- namespace gdcm

13.320 gdcmImageChangePlanarConfiguration.h

Go to the documentation of this file.

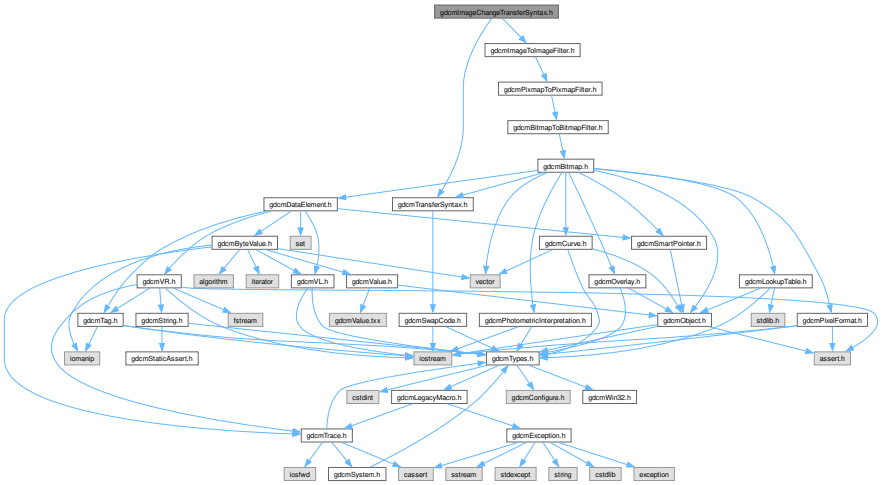
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMIMAGECHANGEPLANARCONFIGURATION_H
00015 #define GDCMIMAGECHANGEPLANARCONFIGURATION_H
00016
00017 #include "gdcmImageToImageFilter.h"
```

```
namespace gdcm
{

class DataElement;
class GDCM_EXPORT ImageChangePlanarConfiguration : public ImageToImageFilter
{
public:
  ImageChangePlanarConfiguration():PlanarConfiguration(0) {}
  ~ImageChangePlanarConfiguration() = default;

  void SetPlanarConfiguration(unsigned int pc) { PlanarConfiguration = pc; }
  unsigned int GetPlanarConfiguration() const { return PlanarConfiguration; }

  template <typename T>
  static size_t RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b, size_t s);

  template <typename T>
  static size_t RGBPixelsToRGBPlanes(T *r, T *g, T *b, const T* rgb, size_t s);

  bool Change();

protected:

private:
  unsigned int PlanarConfiguration;
};

template <typename T>
size_t ImageChangePlanarConfiguration::RGBPlanesToRGBPixels(T *out, const T *r, const T *g, const T *b,
size_t s)
{
  T *pout = out;
  for(size_t i = 0; i < s; ++i )
    {
    *pout++ = *r++;
    *pout++ = *g++;
    *pout++ = *b++;
    }

  assert( (size_t)(pout - out) == 3 * s );
  return pout - out;
}

template <typename T>
size_t ImageChangePlanarConfiguration::RGBPixelsToRGBPlanes(T *r, T *g, T *b, const T *rgb, size_t s)
{
  const T *prgb = rgb;
  for(size_t i = 0; i < s; ++i )
    {
    *r++ = *prgb++;
    *g++ = *prgb++;
    *b++ = *prgb++;
    }
  assert( (size_t)(prgb - rgb) == 3 * s );
  return prgb - rgb;
}


} // end namespace gdcm

#endif //GDCMIMAGECHANGEPLANARCONFIGURATION_H
```

13.321 gdcmImageChangeTransferSyntax.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmTransferSyntax.h"
```

Include dependency graph for gdcmImageChangeTransferSyntax.h:



Classes

- class gdcm::ImageChangeTransferSyntax

  ImageChangeTransferSyntax class.

Namespaces

- namespace gdcm

13.322 gdcmImageChangeTransferSyntax.h

Go to the documentation of this file.

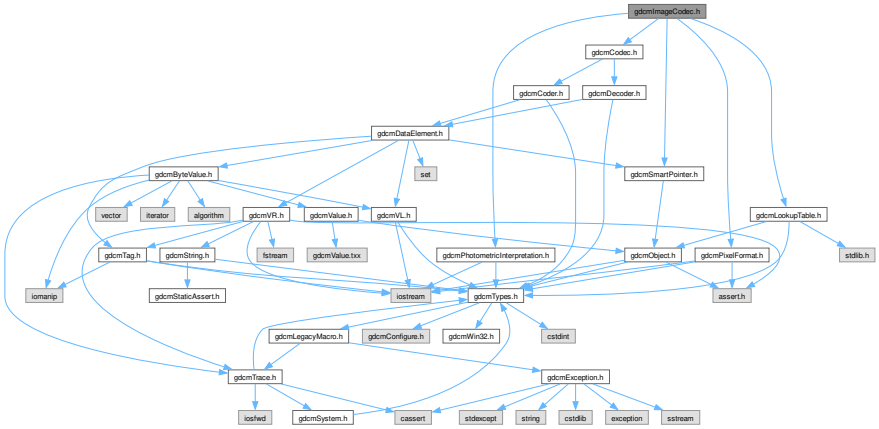
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGECHANGETRANSFERSYNTAX_H
#define GDCMIMAGECHANGETRANSFERSYNTAX_H

#include "gdcmImageToImageFilter.h"
#include "gdcmTransferSyntax.h"

namespace gdcm
{

class DataElement;
class ImageCodec;
```

```
class GDCM_EXPORT ImageChangeTransferSyntax : public ImageToImageFilter
{
public:

ImageChangeTransferSyntax():TS(TransferSyntax::TS_END),Force(false),CompressIconImage(false),UserCodec(nullptr)
{}
  ~ImageChangeTransferSyntax() = default;

  void SetTransferSyntax(const TransferSyntax &ts) { TS = ts; }
  const TransferSyntax &GetTransferSyntax() const { return TS; }

  bool Change();

  void SetCompressIconImage(bool b) { CompressIconImage = b; }

  void SetForce( bool f ) { Force = f; }

  void SetUserCodec(ImageCodec *ic) { UserCodec = ic; }

protected:
  bool TryJPEGCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
  bool TryJPEG2000Codec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
  bool TryJPEGLSCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
  bool TryRAWCodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);
  bool TryRLECodec(const DataElement &pixelde, Bitmap const &input, Bitmap &output);

private:
  TransferSyntax TS;
  bool Force;
  bool CompressIconImage;

  ImageCodec *UserCodec;
};


} // end namespace gdcm

#endif //GDCMIMAGECHANGETRANSFERSYNTAX_H
```

13.323 gdcmImageCodec.h File Reference

```
#include "gdcmCodec.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmLookupTable.h"
#include "gdcmSmartPointer.h"
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmImageCodec.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::ImageCodec

  ImageCodec.

Namespaces

- namespace gdcm

13.324 gdcmImageCodec.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGECODEC_H
#define GDCMIMAGECODEC_H

#include "gdcmCodec.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmLookupTable.h"
#include "gdcmSmartPointer.h"
#include "gdcmPixelFormat.h"

namespace gdcm
{

class GDCM_EXPORT ImageCodec : public Codec
{
  friend class ImageChangePhotometricInterpretation;
public:
  ImageCodec();
  ~ImageCodec() override;
  bool CanCode(TransferSyntax const &) const override { return false; }
  bool CanDecode(TransferSyntax const &) const override { return false; }
  bool Decode(DataElement const &is_, DataElement &os) override;
  bool IsLossy() const;
  void SetLossyFlag(bool l);
  bool GetLossyFlag() const;

  virtual bool GetHeaderInfo(std::istream &is_, TransferSyntax &ts);

  virtual ImageCodec * Clone() const = 0;

```

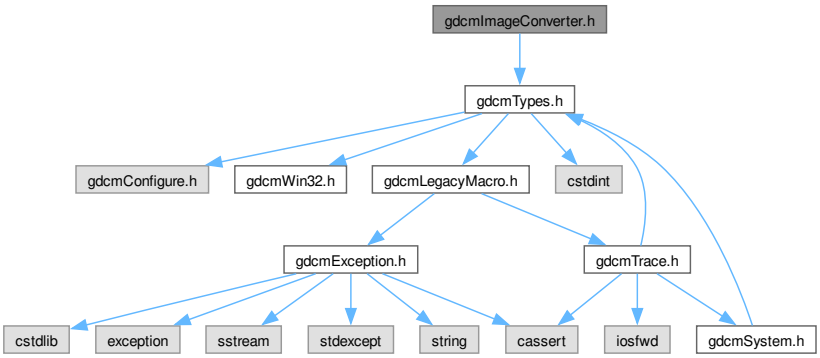
```
protected:
  bool DecodeByStreams(std::istream &is_, std::ostream &os) override;
  virtual bool IsValid(PhotometricInterpretation const &pi);
public:

  unsigned int GetPlanarConfiguration() const
    {
    return PlanarConfiguration;
    }
  void SetPlanarConfiguration(unsigned int pc)
    {
    assert( pc == 0 || pc == 1 );
    PlanarConfiguration = pc;
    }

  PixelFormat &GetPixelFormat()
    {
    return PF;
    }
  const PixelFormat &GetPixelFormat() const
    {
    return PF;
    }
  virtual void SetPixelFormat(PixelFormat const &pf)
    {
    PF = pf;
    }
  const PhotometricInterpretation &GetPhotometricInterpretation() const;
  void SetPhotometricInterpretation(PhotometricInterpretation const &pi);

  bool GetNeedByteSwap() const
    {
    return NeedByteSwap;
    }
  void SetNeedByteSwap(bool b)
    {
    NeedByteSwap = b;
    }
  void SetNeedOverlayCleanup(bool b)
    {
    NeedOverlayCleanup = b;
    }
  void SetLUT(LookupTable const &lut)
    {
    LUT = SmartPointer<LookupTable>( const_cast<LookupTable*>(&lut) );
    }
  const LookupTable &GetLUT() const
    {
    return *LUT;
    }

  void SetDimensions(const unsigned int d[3]);
  void SetDimensions(const std::vector<unsigned int> & d);
  const unsigned int *GetDimensions() const { return Dimensions; }
  void SetNumberOfDimensions(unsigned int dim);
  unsigned int GetNumberOfDimensions() const;

  bool CleanupUnusedBits(char * data, size_t datalen);

protected:
  // Streaming (write) API:
  friend class FileChangeTransferSyntax;
  virtual bool StartEncode( std::ostream & os );
  virtual bool IsRowEncoder();
  virtual bool IsFrameEncoder();
  virtual bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen );
  virtual bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen );
  virtual bool StopEncode( std::ostream & os);

protected:
  bool RequestPlanarConfiguration;
  bool RequestPaddedCompositePixelCode;
//private:
  unsigned int PlanarConfiguration;
  PhotometricInterpretation PI;
  PixelFormat PF;
  bool NeedByteSwap;
  bool NeedOverlayCleanup;

  typedef SmartPointer<LookupTable> LUTPtr;
  LUTPtr LUT;
```

```
  unsigned int Dimensions[3]; // FIXME
  unsigned int NumberOfDimensions;
  bool LossyFlag;

  bool DoOverlayCleanup(std::istream &is_, std::ostream &os);
  bool DoByteSwap(std::istream &is_, std::ostream &os);
  bool DoYBR(std::istream &is_, std::ostream &os);
  bool DoYBRFull422(std::istream &is_, std::ostream &os);
  bool DoPlanarConfiguration(std::istream &is_, std::ostream &os);
  bool DoSimpleCopy(std::istream &is_, std::ostream &os);
  bool DoPaddedCompositePixelCode(std::istream &is_, std::ostream &os);
  bool DoInvertMonochrome(std::istream &is_, std::ostream &os);

  //template <typename T>
  //bool DoInvertPlanarConfiguration(T *output, const T *input, uint32_t length);
};

} // end namespace gdcm

#endif //GDCMIMAGECODEC_H
```

13.325 gdcmImageConverter.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmImageConverter.h:



Classes

- class gdcm::ImageConverter

  Image Converter.

Namespaces

- namespace gdcm

13.326 gdcmImageConverter.h

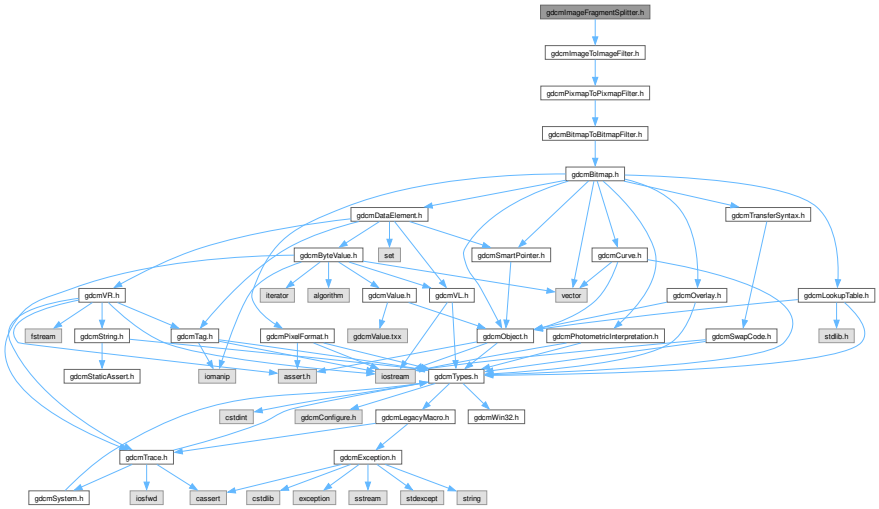
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMIMAGECONVERTER_H
#define GDCMIMAGECONVERTER_H

#include "gdcmTypes.h"

namespace gdcm
{

class Image;
class GDCM_EXPORT ImageConverter
{
public:
  ImageConverter();
  ~ImageConverter();

  void SetInput(Image const &input);
  const Image& GetOuput() const;

  void Convert();

private:
  Image *Input;
  Image *Output;
};

} // end namespace gdcm

#endif //GDCMIMAGECONVERTER_H
```

13.327 gdcmImageFragmentSplitter.h File Reference

`#include "gdcmImageToImageFilter.h"`

Include dependency graph for gdcmImageFragmentSplitter.h:



Classes

- class gdcm::ImageFragmentSplitter

  ImageFragmentSplitter class.

Namespaces

- namespace gdcm

13.328 gdcmImageFragmentSplitter.h

Go to the documentation of this file.

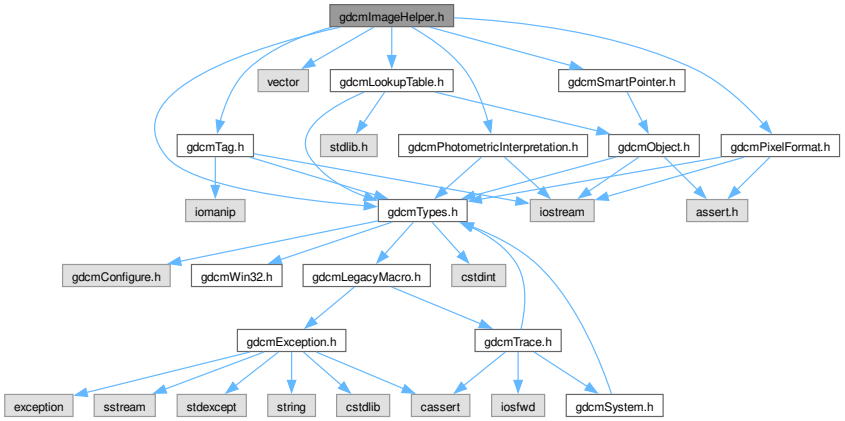
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGEFRAGMENTSPLITTER_H
#define GDCMIMAGEFRAGMENTSPLITTER_H

#include "gdcmImageToImageFilter.h"
```

```
namespace gdcm
{

class DataElement;
class GDCM_EXPORT ImageFragmentSplitter : public ImageToImageFilter
{
public:
  ImageFragmentSplitter():FragmentSizeMax(0),Force(false) {}
  ~ImageFragmentSplitter() = default;

  bool Split();

  void SetFragmentSizeMax(unsigned int fragsize);
  unsigned int GetFragmentSizeMax() const { return FragmentSizeMax; }

  void SetForce( bool f ) { Force = f; }

protected:

private:
  unsigned int FragmentSizeMax;
  bool Force;
};

} // end namespace gdcm

#endif //GDCMIMAGEFRAGMENTSPLITTER_H
```

13.329 gdcmImageHelper.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include <vector>
#include "gdcmPixelFormat.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmSmartPointer.h"
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmImageHelper.h:



Classes

- class gdcm::ImageHelper

ImageHelper (internal class, not intended for user level).

- struct gdcm::RealWorldValueMappingContent

Namespaces

- namespace gdcm

13.330 gdcmImageHelper.h

Go to the documentation of this file.

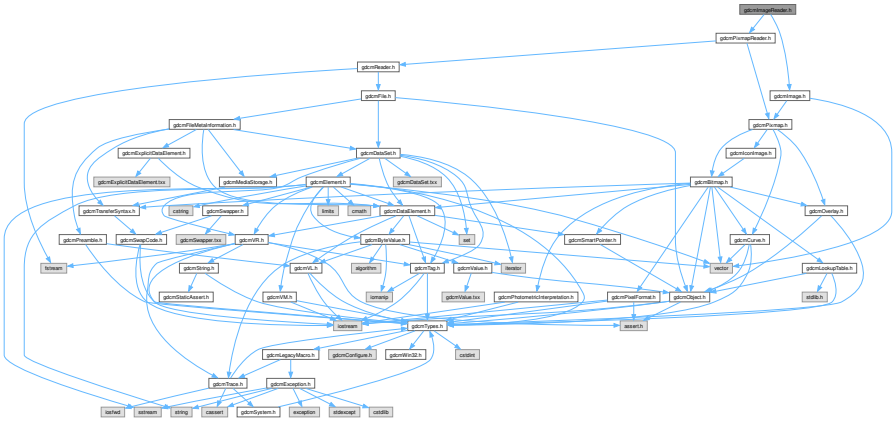
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGEHELPER_H
#define GDCMIMAGEHELPER_H

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include <vector>
#include "gdcmPixelFormat.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmSmartPointer.h"
#include "gdcmLookupTable.h"

namespace gdcm
{

class MediaStorage;
class DataSet;
class File;
class Image;
class Pixmap;
class ByteValue;

// minimal struct:
struct RealWorldValueMappingContent {
  double RealWorldValueIntercept;
  double RealWorldValueSlope;
  // http://dicom.nema.org/MEDICAL/DICOM/2014c/output/chtml/part16/sect_CID_7181.html
  std::string CodeValue;
  std::string CodeMeaning;
};

class GDCM_EXPORT ImageHelper
{
public:
  static void SetForceRescaleInterceptSlope(bool);
  static bool GetForceRescaleInterceptSlope();

  static void SetPMSRescaleInterceptSlope(bool);
  static bool GetPMSRescaleInterceptSlope();

  static void SetForcePixelSpacing(bool);
  static bool GetForcePixelSpacing();

  static void SetSecondaryCaptureImagePlaneModule(bool);
  static bool GetSecondaryCaptureImagePlaneModule();

  static std::vector<unsigned int> GetDimensionsValue(const File& f);
  static void SetDimensionsValue(File& f, const Pixmap & img);
```

```
00100
00103   static PixelFormat GetPixelFormatValue(const File& f);
00104
00109   static std::vector<double> GetRescaleInterceptSlopeValue(File const & f);
00110   static void SetRescaleInterceptSlopeValue(File & f, const Image & img);
00111
00112   // read only for now
00113   static bool GetRealWorldValueMappingContent(File const & f, RealWorldValueMappingContent & rwvmc);
00114
00116   static std::vector<double> GetOriginValue(File const & f);
00117   static void SetOriginValue(DataSet & ds, const Image & img);
00118
00121   static std::vector<double> GetDirectionCosinesValue(File const & f);
00127   // FIXME: There is a major issue for image with multiple IOP (eg. Enhanced * Image Storage).
00128   static void SetDirectionCosinesValue(DataSet & ds, const std::vector<double> & dircos);
00129
00131   static std::vector<double> GetSpacingValue(File const & f);
00133   static void SetSpacingValue(DataSet & ds, const std::vector<double> & spacing);
00134
00136   static bool ComputeSpacingFromImagePositionPatient(const std::vector<double> &imageposition,
      std::vector<double> & spacing);
00137
00138   static bool GetDirectionCosinesFromDataSet(DataSet const & ds, std::vector<double> & dircos);
00139
00140   //functions to get more information from a file
00141   //useful for the stream image reader, which fills in necessary image information
00142   //distinctly from the reader-style data input
00143   static PhotometricInterpretation GetPhotometricInterpretationValue(File const& f);
00144   //returns the configuration of colors in a plane, either RGB RGB RGB or RRR GGG BBB
00145   static unsigned int GetPlanarConfigurationValue(const File& f);
00146
00148   static SmartPointer<LookupTable> GetLUT(File const& f);
00149
00150   // Moved from PixampReader to here. Generally used for photometric interpretation.
00151   static const ByteValue* GetPointerFromElement(Tag const &tag, File const& f);
00152
00154   static MediaStorage ComputeMediaStorageFromModality(const char *modality,
00155     unsigned int dimension = 2, PixelFormat const & pf = PixelFormat(),
00156     PhotometricInterpretation const & pi = PhotometricInterpretation(),
00157     double rescaleintercept = 0, double rescaleslope = 1 );
00158
00159 protected:
00160   static Tag GetSpacingTagFromMediaStorage(MediaStorage const &ms);
00161   static Tag GetZSpacingTagFromMediaStorage(MediaStorage const &ms);
00162
00163 private:
00164   static bool ForceRescaleInterceptSlope;
00165   static bool PMSRescaleInterceptSlope;
00166   static bool ForcePixelSpacing;
00167   static bool SecondaryCaptureImagePlaneModule;
00168 };
00169
00170 } // end namespace gdcm
00171
00172 #endif // GDCMIMAGEHELPER_H
```

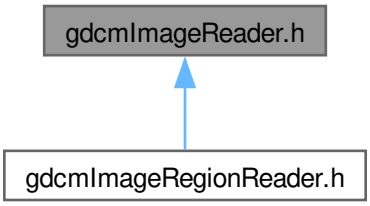
13.331 gdcmImageReader.h File Reference

```
#include "gdcmPixmapReader.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ImageReader

  ImageReader.

Namespaces

- namespace gdcm

13.332 gdcmImageReader.h

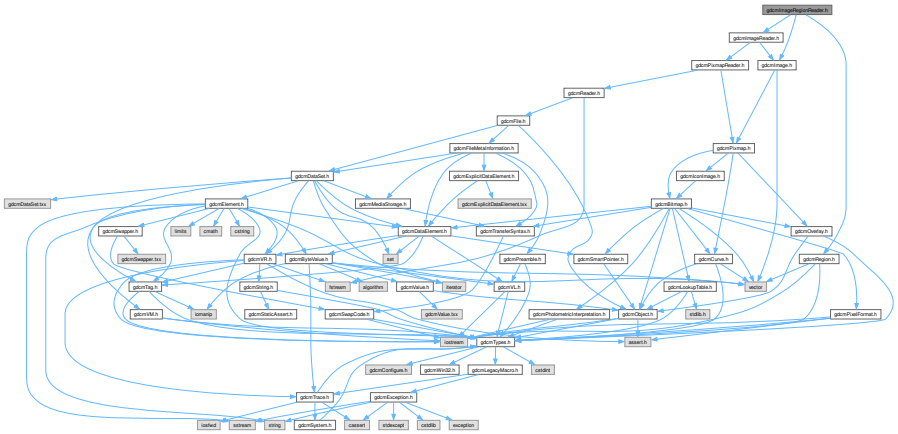
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGEREADER_H
#define GDCMIMAGEREADER_H

#include "gdcmPixmapReader.h"
#include "gdcmImage.h"

namespace gdcm
{

class MediaStorage;
class GDCM_EXPORT ImageReader : public PixmapReader
{
public:
  ImageReader();
  ~ImageReader() override;//needs to be virtual to ensure lack of memory leaks


  bool Read() override;

  // Following methods are valid only after a call to 'Read'

  const Image& GetImage() const;
  Image& GetImage();
  //void SetImage(Image const &img);

protected:
  bool ReadImage(MediaStorage const &ms) override;
  bool ReadACRNEMAImage() override;
};

} // end namespace gdcm

#endif //GDCMIMAGEREADER_H
```

13.333 gdcmImageRegionReader.h File Reference

```
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmRegion.h"
```

Include dependency graph for gdcmImageRegionReader.h:



Classes

- class gdcm::ImageRegionReader

  ImageRegionReader.

Namespaces

- namespace gdcm

13.334 gdcmImageRegionReader.h

Go to the documentation of this file.

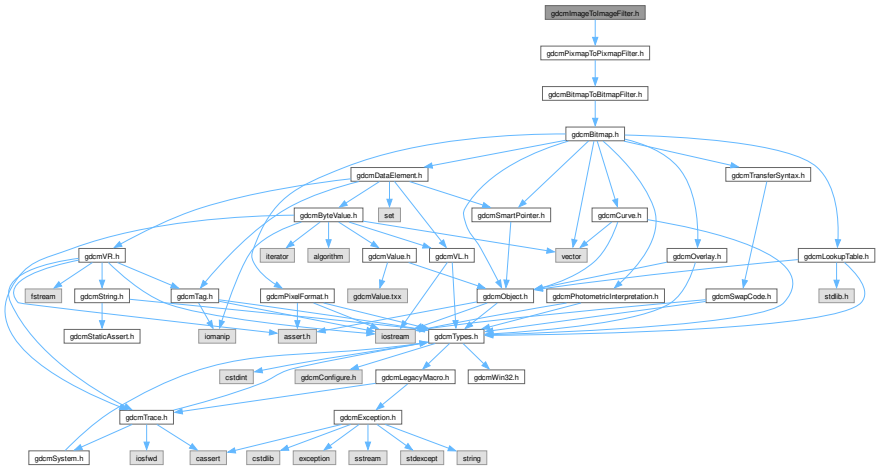
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGEEXTENTREADER_H
#define GDCMIMAGEEXTENTREADER_H

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmRegion.h"

namespace gdcm
{

class ImageRegionReaderInternals;
class GDCM_EXPORT ImageRegionReader : public ImageReader
{
public:
```

```
  ImageRegionReader();
  ~ImageRegionReader() override;

  void SetRegion(Region const & region);
  Region const &GetRegion() const;

  size_t ComputeBufferLength() const;

  bool ReadInformation();

  bool ReadIntoBuffer(char *inreadbuffer, size_t buflen);

protected:
  bool Read() override;

private:
  BoxRegion ComputeBoundingBox();
  bool ReadRAWIntoBuffer(char *buffer, size_t buflen);
  bool ReadRLEIntoBuffer(char *buffer, size_t buflen);
  bool ReadJPEG2000IntoBuffer(char *buffer, size_t buflen);
  bool ReadJPEGIntoBuffer(char *buffer, size_t buflen);
  bool ReadJPEGLSIntoBuffer(char *buffer, size_t buflen);
  ImageRegionReaderInternals *Internals;
};

} // end namespace gdcm

#endif //GDCMIMAGEEXTENTREADER_H
```

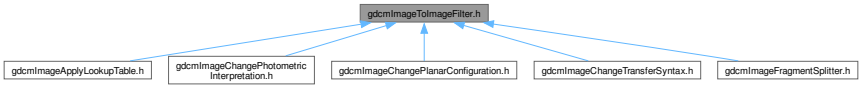
13.335 gdcmImageToImageFilter.h File Reference

`#include "gdcmPixmapToPixmapFilter.h"`

Include dependency graph for gdcmImageToImageFilter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::ImageToImageFilter

  ImageToImageFilter class.

Namespaces

- namespace gdcm

13.336 gdcmImageToImageFilter.h

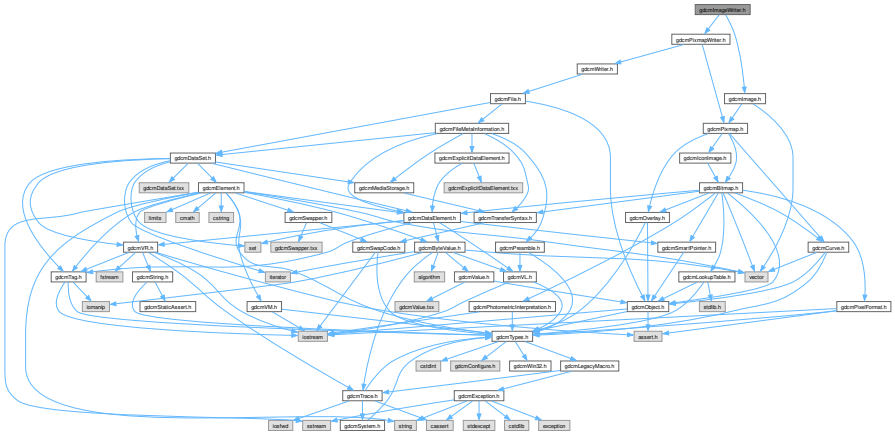
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGETOIMAGEFILTER_H
#define GDCMIMAGETOIMAGEFILTER_H

#include "gdcmPixmapToPixmapFilter.h"

namespace gdcm
{

class Image;
class GDCM_EXPORT ImageToImageFilter : public PixmapToPixmapFilter
{
public:
  ImageToImageFilter();
  ~ImageToImageFilter() = default;

  Image &GetInput();

  // NOTE: covariant return-type to preserve backward compatible API
  const Image &GetOutput() const;

protected:
};

} // end namespace gdcm

#endif //GDCMIMAGETOIMAGEFILTER_H
```

13.337 gdcmImageWriter.h File Reference

```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmImageWriter.h:



Classes

- class gdcm::ImageWriter

  ImageWriter.

Namespaces

- namespace gdcm

13.338 gdcmImageWriter.h

Go to the documentation of this file.

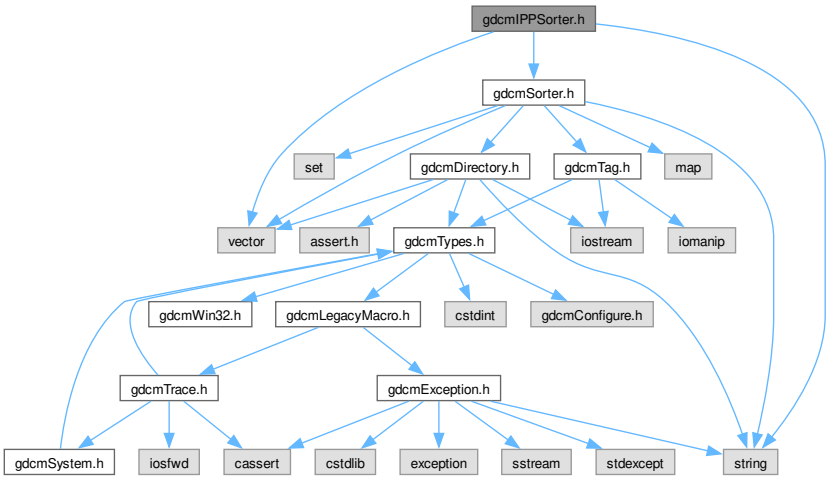
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMAGEWRITER_H
#define GDCMIMAGEWRITER_H

#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"

namespace gdcm
{

class Image;
class GDCM_EXPORT ImageWriter : public PixmapWriter
{
public:
  ImageWriter();
```

```
  ~ImageWriter() override;

  const Image& GetImage() const override { return dynamic_cast<const Image&>(*PixelData); }
  Image& GetImage() override { return dynamic_cast<Image&>(*PixelData); } // FIXME
  //void SetImage(Image const &img);

  bool Write() override; // Execute()

  MediaStorage ComputeTargetMediaStorage();
protected:

private:
};

} // end namespace gdcm

#endif //GDCMIMAGEWRITER_H
```

13.339 gdcmIPPSorter.h File Reference

```
#include "gdcmSorter.h"
#include <vector>
#include <string>
```

Include dependency graph for gdcmIPPSorter.h:



Classes

- class gdcm::IPPSorter

  IPPSorter.

Namespaces

- namespace gdcm

13.340 gdcmIPPSorter.h

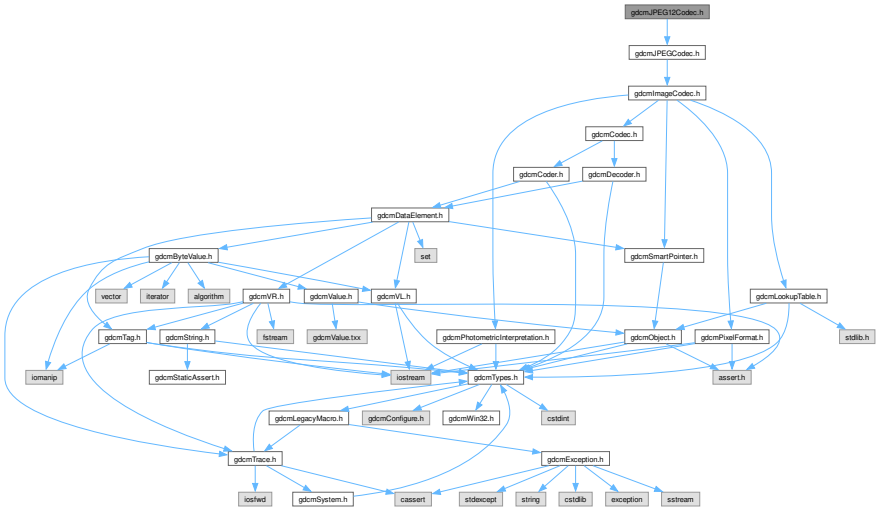
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIPPSORTER_H
#define GDCMIPPSORTER_H

#include "gdcmSorter.h"

#include <vector>
#include <string>

namespace gdcm
{
class GDCM_EXPORT IPPSorter : public Sorter
{
public:
  IPPSorter();

  // FIXME: I do not like public virtual function...
  bool Sort(std::vector<std::string> const & filenames) override;

  void SetComputeZSpacing(bool b) { ComputeZSpacing = b; }
  void SetZSpacingTolerance(double tol) { ZTolerance = tol; }
  double GetZSpacingTolerance() const { return ZTolerance; }

  void SetDirectionCosinesTolerance(double tol) { DirCosTolerance = tol; }
  double GetDirectionCosinesTolerance() const { return DirCosTolerance; }

  void SetDropDuplicatePositions(bool b) { DropDuplicatePositions = b; }

  double GetZSpacing() const { return ZSpacing; }

protected:
  bool ComputeZSpacing;
  bool DropDuplicatePositions;
  double ZSpacing;
  double ZTolerance;
  double DirCosTolerance;

private:
  GDCM_LEGACY(bool ComputeSpacing(std::vector<std::string> const & filenames))
};


} // end namespace gdcm

#endif //GDCMIPPSORTER_H
```

13.341 gdcmJPEG12Codec.h File Reference

`#include "gdcmJPEGCodec.h"`

Include dependency graph for gdcmJPEG12Codec.h:



Classes

- class gdcm::JPEG12Codec

  Class to do JPEG 12bits (lossy & lossless).

Namespaces

- namespace gdcm

13.342 gdcmJPEG12Codec.h

Go to the documentation of this file.

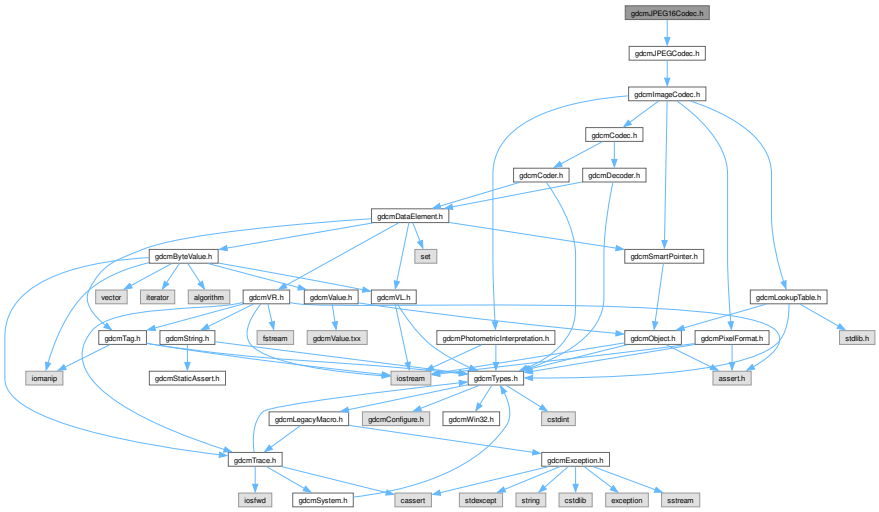
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMJPEG12CODEC_H
00015 #define GDCMJPEG12CODEC_H
00016
00017 #include "gdcmJPEGCodec.h"
```

```
namespace gdcm
{

class JPEGInternals_12BIT;
class ByteValue;
class JPEG12Codec : public JPEGCodec
{
public:
  JPEG12Codec();
  ~JPEG12Codec() override;

  bool DecodeByStreams(std::istream &is, std::ostream &os) override;
  bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;

protected:
  bool IsStateSuspension() const override;
  bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;

private:
  JPEGInternals_12BIT *Internals;
};

} // end namespace gdcm

#endif //GDCMJPEG12CODEC_H
```

13.343 gdcmJPEG16Codec.h File Reference

`#include "gdcmJPEGCodec.h"`

Include dependency graph for gdcmJPEG16Codec.h:



Classes

- class gdcm::JPEG16Codec

  Class to do JPEG 16bits (lossless).

Namespaces

- namespace gdcm

13.344 gdcmJPEG16Codec.h

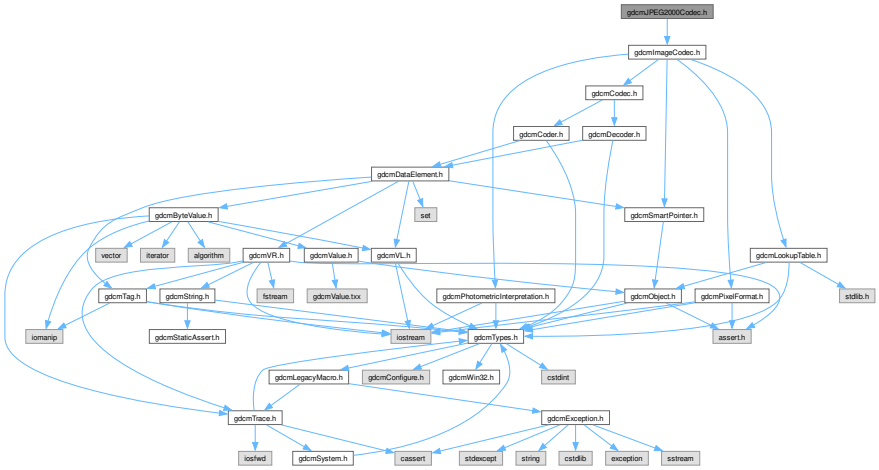
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMJPEG16CODEC_H
#define GDCMJPEG16CODEC_H

#include "gdcmJPEGCodec.h"

namespace gdcm
{

class JPEGInternals_16BIT;
class ByteValue;
class JPEG16Codec : public JPEGCodec
{
public:
  JPEG16Codec();
  ~JPEG16Codec() override;

  bool DecodeByStreams(std::istream &is, std::ostream &os) override;
  bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;

protected:
  bool IsStateSuspension() const override;
  bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;

private:
  JPEGInternals_16BIT *Internals;
};

} // end namespace gdcm

#endif //GDCMJPEG16CODEC_H
```

13.345 gdcmJPEG2000Codec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmJPEG2000Codec.h:



Classes

- class gdcm::JPEG2000Codec

  Class to do JPEG 2000.

Namespaces

- namespace gdcm

13.346 gdcmJPEG2000Codec.h

Go to the documentation of this file.

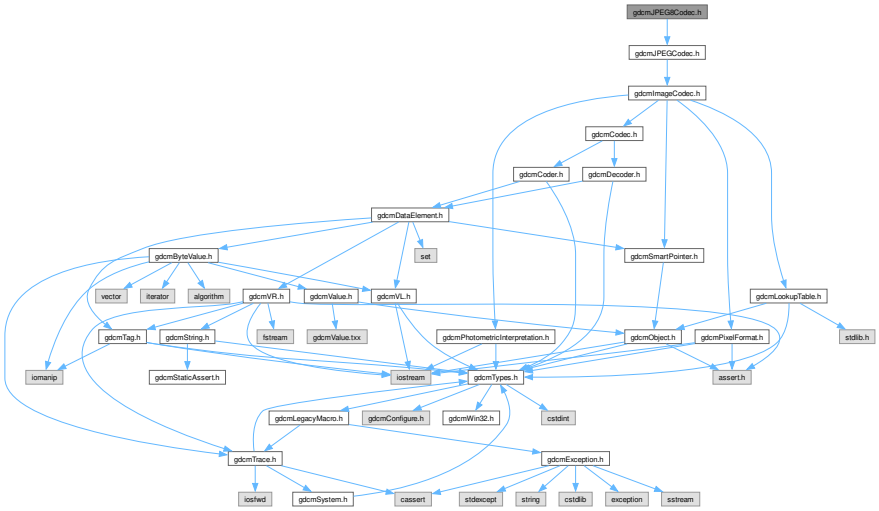
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMJPEG2000CODEC_H
00015 #define GDCMJPEG2000CODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
```

```
{

class JPEG2000Internals;
class GDCM_EXPORT JPEG2000Codec : public ImageCodec
{
friend class ImageRegionReader;
  friend class Bitmap;
public:
  JPEG2000Codec();
  ~JPEG2000Codec() override;

  bool CanDecode(TransferSyntax const &ts) const override;
  bool CanCode(TransferSyntax const &ts) const override;

  bool Decode(DataElement const &is, DataElement &os) override;
  bool Code(DataElement const &in, DataElement &out) override;

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
  ImageCodec * Clone() const override;

  // JPEG-2000 / OpenJPEG specific way of encoding lossy-ness
  // ref: http://www.openjpeg.org/index.php?menu=doc#encoder
  void SetRate(unsigned int idx, double rate);
  double GetRate(unsigned int idx = 0) const;

  void SetQuality(unsigned int idx, double q);
  double GetQuality(unsigned int idx = 0) const;

  void SetTileSize(unsigned int tx, unsigned int ty);

  void SetNumberOfResolutions(unsigned int nres);

  void SetNumberOfThreadsForDecompression(int nThreads);

  void SetReversible(bool res);
  void SetMCT(unsigned int mct);

protected:
  bool DecodeExtent(
    char *buffer,
    unsigned int xmin, unsigned int xmax,
    unsigned int ymin, unsigned int ymax,
    unsigned int zmin, unsigned int zmax,
    std::istream & is
  );

  bool DecodeByStreams(std::istream &is, std::ostream &os) override;

  bool StartEncode( std::ostream & ) override;
  bool IsRowEncoder() override;
  bool IsFrameEncoder() override;
  bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
  bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
  bool StopEncode( std::ostream & ) override;

private:
  std::pair<char *, size_t> DecodeByStreamsCommon(char *dummy_buffer, size_t buf_size);
  bool CodeFrameIntoBuffer(char * outdata, size_t outlen, size_t & complen, const char * indata, size_t
inlen );
  bool GetHeaderInfo(const char * dummy_buffer, size_t len, TransferSyntax &ts);
  JPEG2000Internals *Internals;
};

} // end namespace gdcm

#endif //GDCMJPEG2000CODEC_H
```

13.347 gdcmJPEG8Codec.h File Reference

`#include "gdcmJPEGCodec.h"`

Include dependency graph for gdcmJPEG8Codec.h:



Classes

- class gdcm::JPEG8Codec

  Class to do JPEG 8bits (lossy & lossless).

Namespaces

- namespace gdcm

13.348 gdcmJPEG8Codec.h

Go to the documentation of this file.

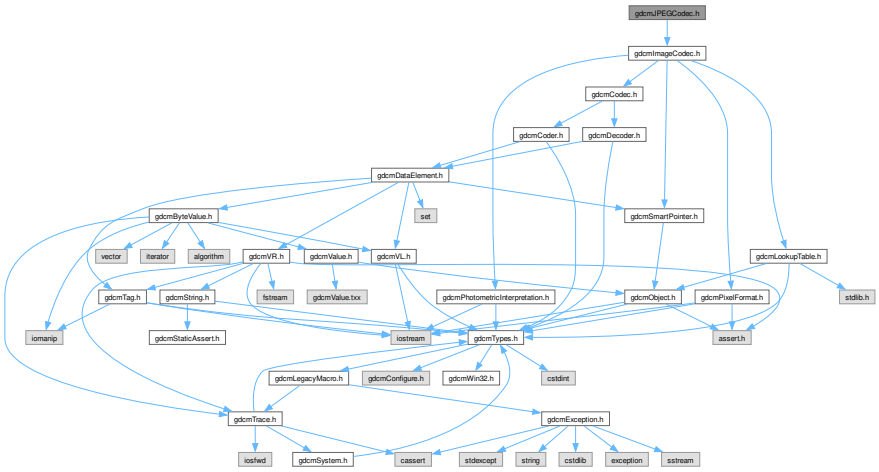
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMJPEG8CODEC_H
00015 #define GDCMJPEG8CODEC_H
00016
00017 #include "gdcmJPEGCodec.h"
```

```
namespace gdcm
{

class JPEGInternals_8BIT;
class ByteValue;
class JPEG8Codec : public JPEGCodec
{
public:
  JPEG8Codec();
  ~JPEG8Codec() override;

  bool DecodeByStreams(std::istream &is, std::ostream &os) override;
  bool InternalCode(const char *input, unsigned long len, std::ostream &os) override;

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;

protected:
  bool IsStateSuspension() const override;
  bool EncodeBuffer(std::ostream &os, const char *data, size_t datalen) override;

private:
  JPEGInternals_8BIT *Internals;
};

} // end namespace gdcm

#endif //GDCMJPEG8CODEC_H
```

13.349 gdcmJPEGCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmJPEGCodec.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::JPEGCodec

  JPEG codec.

Namespaces

- namespace gdcm

13.350 gdcmJPEGCodec.h

Go to the documentation of this file.

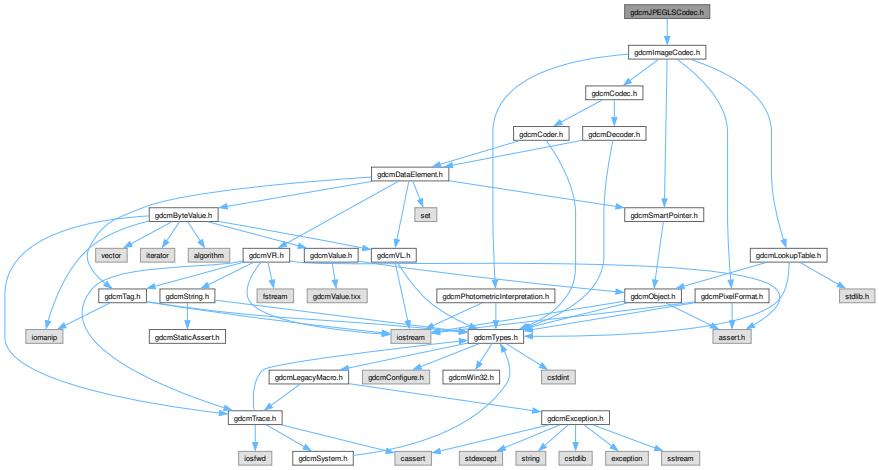
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMJPEGCODEC_H
#define GDCMJPEGCODEC_H

#include "gdcmImageCodec.h"

namespace gdcm
{

class PixelFormat;
class TransferSyntax;
class GDCM_EXPORT JPEGCodec : public ImageCodec
{
friend class ImageRegionReader;
public:
  JPEGCodec();
  ~JPEGCodec() override;
  bool CanDecode(TransferSyntax const &ts) const override;
  bool CanCode(TransferSyntax const &ts) const override;
  bool Decode(DataElement const &is, DataElement &os) override;
  void SetPixelFormat(PixelFormat const &pf) override;

```

```
  void ComputeOffsetTable(bool b);

  bool Code(DataElement const &in, DataElement &out) override;

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
  ImageCodec * Clone() const override;

  //void SetReversible(bool res);

  void SetQuality(double q);
  double GetQuality() const;

  void SetLossless(bool l);
  bool GetLossless() const;

  virtual bool EncodeBuffer( std::ostream & out,
    const char *inbuffer, size_t inlen);

protected:
  bool DecodeExtent(
    char *buffer,
    unsigned int xmin, unsigned int xmax,
    unsigned int ymin, unsigned int ymax,
    unsigned int zmin, unsigned int zmax,
    std::istream & is
  );

  bool DecodeByStreams(std::istream &is, std::ostream &os) override;
  bool IsValid(PhotometricInterpretation const &pi) override;

  bool StartEncode( std::ostream & ) override;
  bool IsRowEncoder() override;
  bool IsFrameEncoder() override;
  bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
  bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
  bool StopEncode( std::ostream & ) override;

protected:
  // Internal method called by SetPixelFormat
  // Instantiate the right jpeg codec (8, 12 or 16)
  void SetBitSample(int bit);

  virtual bool IsStateSuspension() const;

protected:
  int BitSample;
  //bool Lossless;
  int Quality;

private:
  void SetupJPEGBitCodec(int bit);
  JPEGCodec *Internal;
};

} // end namespace gdcm

#endif //GDCMJPEGCODEC_H
```

13.351 gdcmJPEGLSCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmJPEGLSCodec.h:



Classes

- class gdcm::JPEGLSCodec

  JPEG-LS.

Namespaces

- namespace gdcm

13.352 gdcmJPEGLSCodec.h

Go to the documentation of this file.

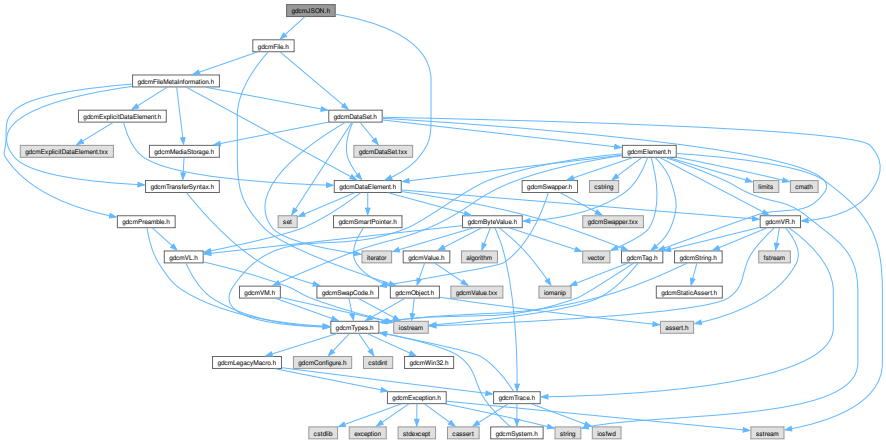
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMJPEGLSCODEC_H
00015 #define GDCMJPEGLSCODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
```

```
{

class JPEGLSInternals;
class GDCM_EXPORT JPEGLSCodec : public ImageCodec
{
friend class ImageRegionReader;
public:
  JPEGLSCodec();
  ~JPEGLSCodec() override;
  bool CanDecode(TransferSyntax const &ts) const override;
  bool CanCode(TransferSyntax const &ts) const override;

  unsigned long GetBufferLength() const { return BufferLength; }
  void SetBufferLength(unsigned long l) { BufferLength = l; }

  bool Decode(DataElement const &is, DataElement &os) override;
  bool Decode(DataElement const &in, char* outBuffer, size_t inBufferLength,
              uint32_t inXMin, uint32_t inXMax, uint32_t inYMin,
              uint32_t inYMax, uint32_t inZMin, uint32_t inZMax);
  bool Code(DataElement const &in, DataElement &out) override;

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
  ImageCodec * Clone() const override;

  void SetLossless(bool l);
  bool GetLossless() const;

/*
 * test.acr can look pretty bad, even with a lossy error of 2. Explanation follows:
 * I agree that the test image looks ugly. In this particular case I can
 * explain though.
 *
 * The image is 8 bit, but it does not use the full 8 bit dynamic range. The
 * black pixels have value 234 and the white 255. If you set allowed lossy
 * error to 2, you allow an error of about 10% of the actual dynamic range.
 * That is of course very visible.
 */
  void SetLossyError(int error);

protected:
  bool DecodeExtent(
    char *buffer,
    unsigned int xmin, unsigned int xmax,
    unsigned int ymin, unsigned int ymax,
    unsigned int zmin, unsigned int zmax,
    std::istream & is
  );

  bool StartEncode( std::ostream & ) override;
  bool IsRowEncoder() override;
  bool IsFrameEncoder() override;
  bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
  bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
  bool StopEncode( std::ostream & ) override;

private:
  bool DecodeByStreamsCommon(const char *buffer, size_t totalLen, std::vector<unsigned char> &rgbyteOut);
  bool CodeFrameIntoBuffer(char * outdata, size_t outlen, size_t & complen, const char * indata, size_t
inlen );

  unsigned long BufferLength;
  int LossyError;
};

} // end namespace gdcm

#endif //GDCMJPEGLSCODEC_H
```

13.353 gdcmJSON.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmJSON.h:



Classes

- class gdcm::JSON

Namespaces

- namespace gdcm

13.354 gdcmJSON.h

Go to the documentation of this file.

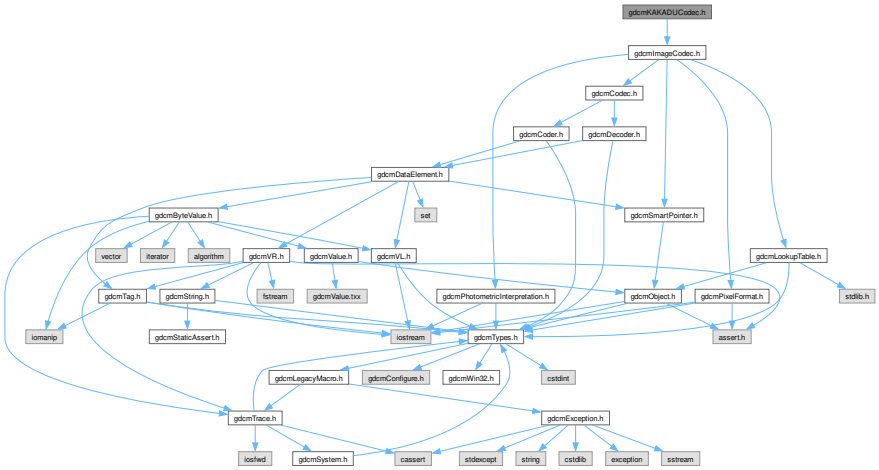
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMJSON_H
#define GDCMJSON_H

/*
See Sup 166 (QIDO-RS)
http://www.dclunie.com/dicom-status/status.html#Supplement166
*/

#include "gdcmFile.h"
#include "gdcmDataElement.h"

namespace gdcm
{

class JSONInternal;
```

```
class GDCM_EXPORT JSON
{
public:
  JSON();
  ~JSON();

  bool GetPrettyPrint() const;
  void SetPrettyPrint(bool onoff);
  void PrettyPrintOn();
  void PrettyPrintOff();

  bool Code(DataSet const & in, std::ostream & os);
  bool Decode(std::istream & is, DataSet & out);

private:
  JSONInternal *Internals;
};

} // end namespace gdcm

#endif //GDCMXMLPRINTER_H
```

13.355 gdcmKAKADUCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmKAKADUCodec.h:



Classes

- class gdcm::KAKADUCodec

  KAKADUCodec.

Namespaces

- namespace gdcm

13.356 gdcmKAKADUCodec.h

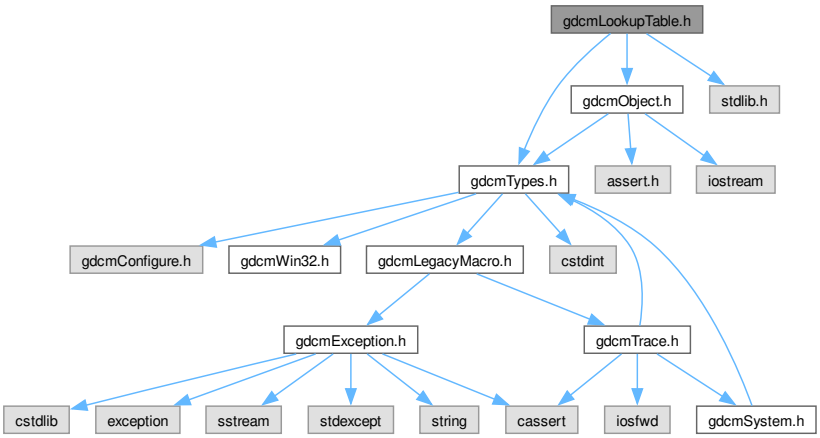
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMKAKADUCODEC_H
#define GDCMKAKADUCODEC_H

#include "gdcmImageCodec.h"

namespace gdcm
{

class KAKADUCodec : public ImageCodec
{
public:
  KAKADUCodec();
  ~KAKADUCodec() override;
  bool CanDecode(TransferSyntax const &ts) const override;
  bool CanCode(TransferSyntax const &ts) const override;

  bool Decode(DataElement const &is, DataElement &os) override;
  bool Code(DataElement const &in, DataElement &out) override;

  ImageCodec * Clone() const override;
private:
};

} // end namespace gdcm

#endif //GDCMKAKADUCODEC_H
```

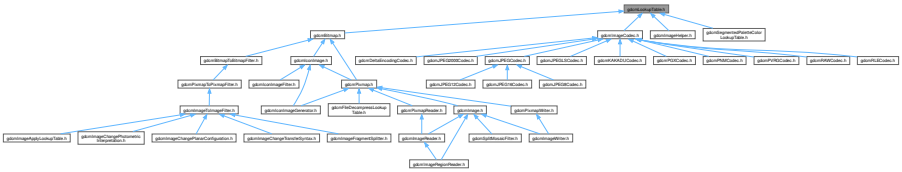
13.357 gdcmLookupTable.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <stdlib.h>
```

Include dependency graph for gdcmLookupTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::LookupTable

  LookupTable class.

Namespaces

- namespace gdcm

13.358 gdcmLookupTable.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
```

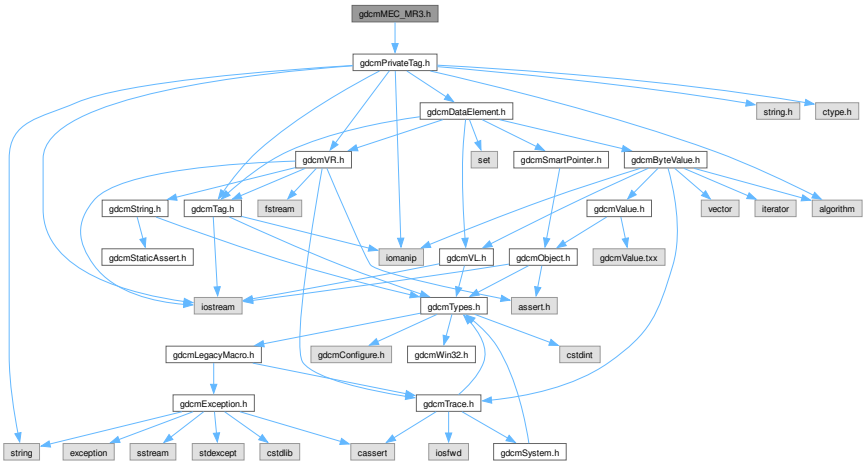
```
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMLOOKUPTABLE_H
#define GDCMLOOKUPTABLE_H

#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <stdlib.h>

namespace gdcm
{

class LookupTableInternal;
class GDCM_EXPORT LookupTable : public Object
{
public:
  typedef enum {
    RED = 0,  // Keep RED == 0
    GREEN,
    BLUE,
    GRAY,
    UNKNOWN
  } LookupTableType;

  LookupTable();
  ~LookupTable() override;
  void Print(std::ostream &) const override;

  void Allocate( unsigned short bitsample = 8 );
  //TODO: check to see if length should be unsigned short, unsigned int, or whatever
  void InitializeLUT(LookupTableType type, unsigned short length,
    unsigned short subscript, unsigned short bitsize);
  unsigned int GetLUTLength(LookupTableType type) const;
  virtual void SetLUT(LookupTableType type, const unsigned char *array,
    unsigned int length);
  void GetLUT(LookupTableType type, unsigned char *array, unsigned int &length) const;
  void GetLUTDescriptor(LookupTableType type, unsigned short &length,
    unsigned short &subscript, unsigned short &bitsize) const;

  void InitializeRedLUT(unsigned short length, unsigned short subscript,
    unsigned short bitsize);
  void SetRedLUT(const unsigned char *red, unsigned int length);
  void InitializeGreenLUT(unsigned short length, unsigned short subscript,
    unsigned short bitsize);
  void SetGreenLUT(const unsigned char *green, unsigned int length);
  void InitializeBlueLUT(unsigned short length, unsigned short subscript,
    unsigned short bitsize);
  void SetBlueLUT(const unsigned char *blue, unsigned int length);

  void Clear();

  void Decode(std::istream &is, std::ostream &os) const;

  bool Decode(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const;

  bool IsRGB8() const;

  bool Decode8(char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const;

  LookupTable(LookupTable const &lut):Object(lut)
    {
    assert(0);
    }

  bool GetBufferAsRGBA(unsigned char *rgba) const;

  const unsigned char *GetPointer() const;

  bool WriteBufferAsRGBA(const unsigned char *rgba);

  unsigned short GetBitSample() const { return BitSample; }

  bool Initialized() const;

private:
```

```
  void Encode(std::istream &is, std::ostream &os);

protected:
  LookupTableInternal *Internal;
  unsigned short BitSample; // refer to the pixel type (not the bit size of LUT)
  bool IncompleteLUT:1;
};

} // end namespace gdcm

#endif //GDCMLOOKUPTABLE_H
```

13.359 gdcmMEC_MR3.h File Reference

`#include "gdcmPrivateTag.h"`

Include dependency graph for gdcmMEC_MR3.h:



Classes

- class gdcm::MEC_MR3

  Class for MEC_MR3.

Namespaces

- namespace gdcm

13.360 gdcmMEC_MR3.h

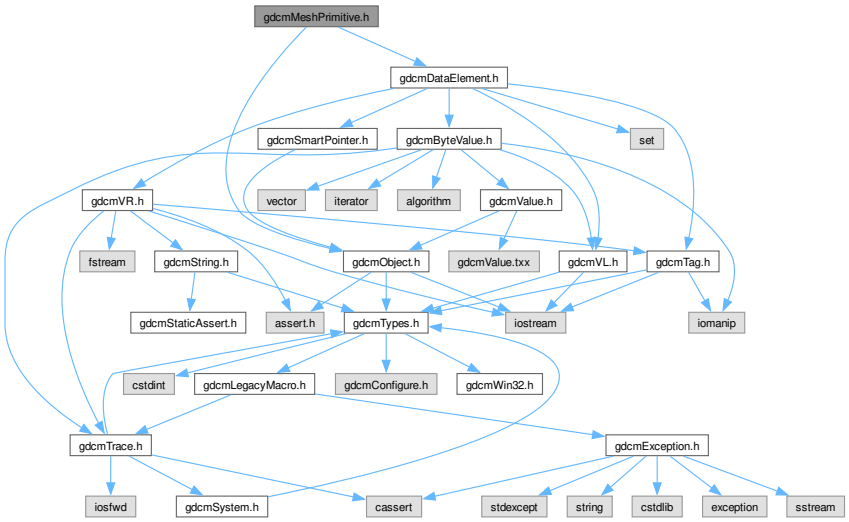
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMEC_MR3_H
#define GDCMMEC_MR3_H

#include "gdcmPrivateTag.h"

namespace gdcm {
class GDCM_EXPORT MEC_MR3 {
 public:
  static bool Print(const char *src, size_t srclen);

  static const PrivateTag &GetPMTFInformationDataTag();

  static const PrivateTag &GetCanonMECMR3Tag();

  static const PrivateTag &GetToshibaMECMR3Tag();
};

}  // end namespace gdcm

#endif  // GDCMMEC_MR3_H
```

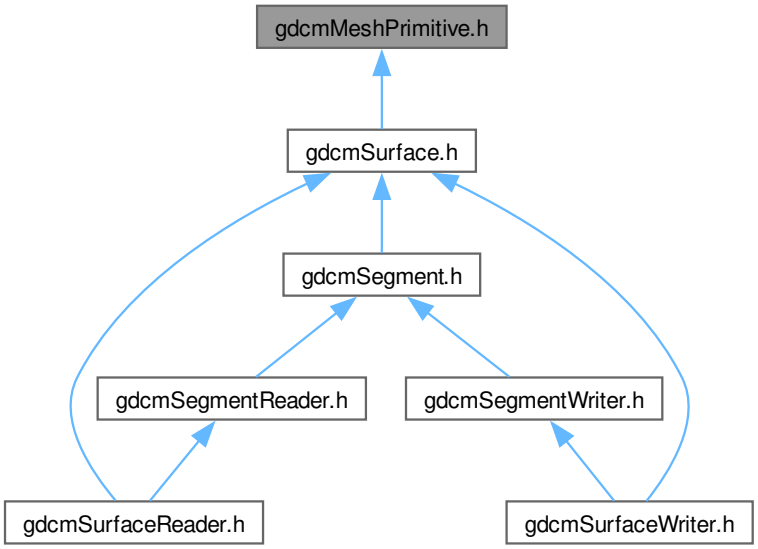
13.361 gdcmMeshPrimitive.h File Reference

```
#include <gdcmObject.h>
#include <gdcmDataElement.h>
```

Include dependency graph for gdcmMeshPrimitive.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::MeshPrimitive

  This class defines surface mesh primitives.

Namespaces

- namespace gdcm

13.362 gdcmMeshPrimitive.h

Go to the documentation of this file.

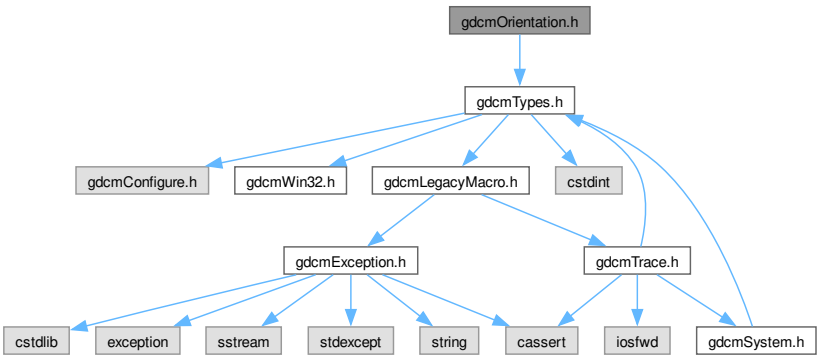
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMMESHPRIMITIVE_H
```

```
#define GDCMMESHPRIMITIVE_H

#include <gdcmObject.h>
#include <gdcmDataElement.h>

namespace gdcm
{

class GDCM_EXPORT MeshPrimitive : public Object
{
public:

  typedef std::vector< DataElement > PrimitivesData;

    typedef enum {
        VERTEX = 0,
        EDGE,
        TRIANGLE,
        TRIANGLE_STRIP,
        TRIANGLE_FAN,
        LINE,
        FACET,
        MPType_END
        } MPType;

    static const char * GetMPTypeString(const MPType type);

    static MPType GetMPType(const char * type);

    MeshPrimitive();

    ~MeshPrimitive() override;

    MPType GetPrimitiveType() const;
    void SetPrimitiveType(const MPType type);

    const DataElement & GetPrimitiveData() const;
    DataElement & GetPrimitiveData();
    void SetPrimitiveData(DataElement const & de);

    const PrimitivesData & GetPrimitivesData() const;
    PrimitivesData & GetPrimitivesData();
    void SetPrimitivesData(PrimitivesData const & DEs);

    const DataElement & GetPrimitiveData(const unsigned int idx) const;
    DataElement & GetPrimitiveData(const unsigned int idx);
    void SetPrimitiveData(const unsigned int idx, DataElement const & de);
    void AddPrimitiveData(DataElement const & de);

    unsigned int GetNumberOfPrimitivesData() const;

protected:

    // Use to define tag where PrimitiveData will be put.
    MPType          PrimitiveType;

    // PrimitiveData contains point index list.
    // It shall have 1 or 1-n DataElement following PrimitiveType.
    PrimitivesData  PrimitiveData;
};

}

#endif // GDCMMESHPRIMITIVE_H
```

13.363 gdcmOrientation.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmOrientation.h:



Classes

- class gdcm::Orientation

  class to handle Orientation

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)

13.364 gdcmOrientation.h

Go to the documentation of this file.

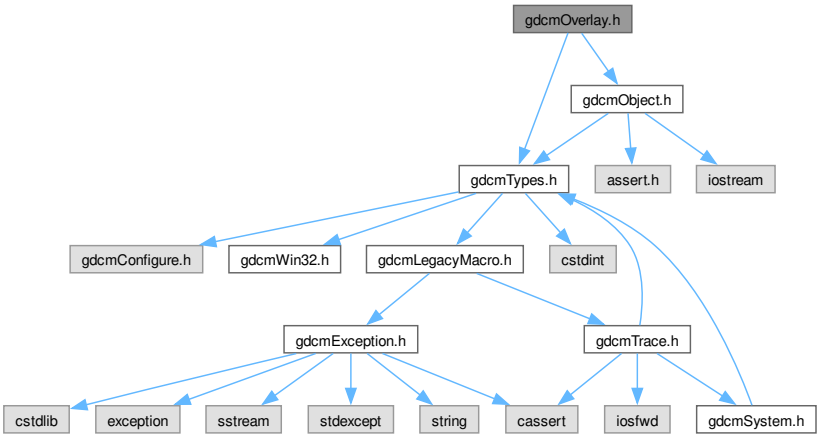
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
```

```
#ifndef GDCMORIENTATION_H
#define GDCMORIENTATION_H

#include "gdcmTypes.h"

namespace gdcm
{

class GDCM_EXPORT Orientation
{
  friend std::ostream& operator«(std::ostream &_os, const Orientation &o);
public:
  Orientation();
  ~Orientation() = default;

  void Print(std::ostream &) const;

  typedef enum {
    UNKNOWN,
    AXIAL,
    CORONAL,
    SAGITTAL,
    OBLIQUE
  } OrientationType;

  static OrientationType GetType(const double dircos[6]);

  static void SetObliquityThresholdCosineValue(double val);
  static double GetObliquityThresholdCosineValue();

  static const char *GetLabel(OrientationType type);

protected:
  static char GetMajorAxisFromPatientRelativeDirectionCosine(double x, double y, double z);

private:
  static double ObliquityThresholdCosineValue;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Orientation &o)
{
  o.Print( os );
  return os;
}

} // end namespace gdcm

#endif //GDCMORIENTATION_H
```

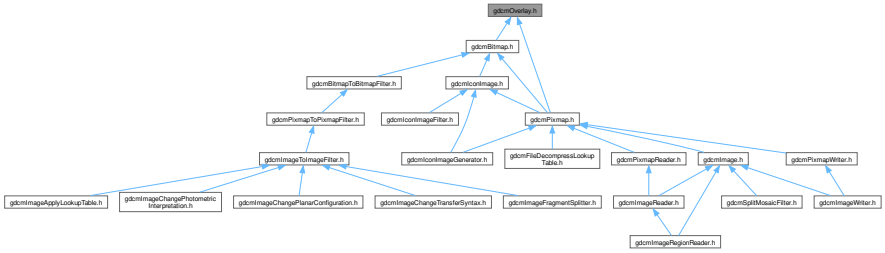
13.365 gdcmOverlay.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
```

Include dependency graph for gdcmOverlay.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Overlay

  Overlay class.

Namespaces

- namespace gdcm

13.366 gdcmOverlay.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
```

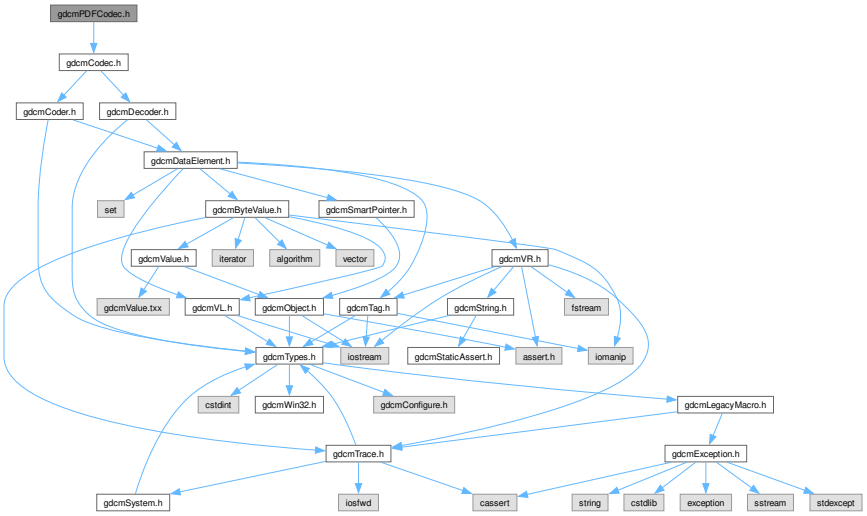
```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMOVERLAY_H
#define GDCMOVERLAY_H

#include "gdcmTypes.h"
#include "gdcmObject.h"

namespace gdcm
{

class OverlayInternal;
class ByteValue;
class DataSet;
class DataElement;
class GDCM_EXPORT Overlay : public Object
{
public:
  Overlay();
  ~Overlay() override;
  void Print(std::ostream &) const override;

  void Update(const DataElement & de);

  void SetGroup(unsigned short group);
  unsigned short GetGroup() const;
  void SetRows(unsigned short rows);
  unsigned short GetRows() const;
  void SetColumns(unsigned short columns);
  unsigned short GetColumns() const;
  void SetNumberOfFrames(unsigned int numberofframes);
  void SetDescription(const char* description);
  const char *GetDescription() const;
  typedef enum {
    Invalid  = 0,
    Graphics = 1,
    ROI      = 2
  } OverlayType;
  void SetType(const char* type);
  const char *GetType() const;
  OverlayType GetTypeAsEnum() const;
  static const char *GetOverlayTypeAsString(OverlayType ot);
  static OverlayType GetOverlayTypeFromString(const char *);
  void SetOrigin(const signed short origin[2]);
  const signed short * GetOrigin() const;
  void SetFrameOrigin(unsigned short frameorigin);
  void SetBitsAllocated(unsigned short bitsallocated);
  unsigned short GetBitsAllocated() const;
  void SetBitPosition(unsigned short bitposition);
  unsigned short GetBitPosition() const;

  void SetOverlay(const char *array, size_t length);

  bool GrabOverlayFromPixelData(DataSet const &ds);

  const ByteValue &GetOverlayData() const;

  bool IsEmpty() const;

  bool IsZero() const;

  bool IsInPixelData() const;

  void IsInPixelData(bool b);

  void Decompress(std::ostream &os) const;

  size_t GetUnpackBufferLength() const;

  bool GetUnpackBuffer(char *buffer, size_t len) const;

  Overlay(Overlay const &ov);
```

```
  Overlay &operator=(Overlay const &ov);

private:
  OverlayInternal *Internal;
};

} // end namespace gdcm

#endif //GDCMOVERLAY_H
```

13.367 gdcmPDFCodec.h File Reference

`#include "gdcmCodec.h"`

Include dependency graph for gdcmPDFCodec.h:



Classes

- class gdcm::PDFCodec

  PDFCodec class.

Namespaces

- namespace gdcm

13.368 gdcmPDFCodec.h

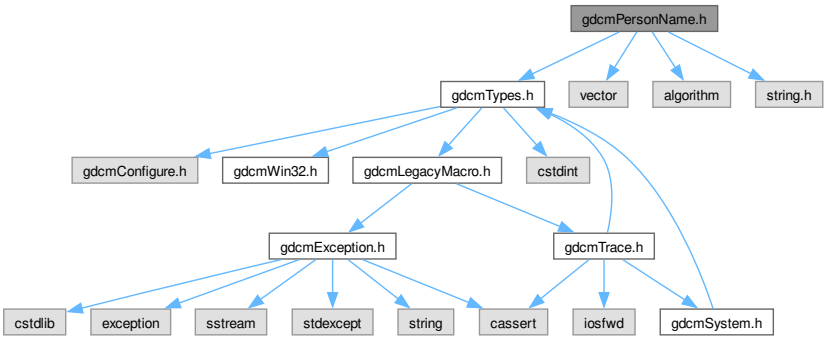
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPDFCODEC_H
#define GDCMPDFCODEC_H

#include "gdcmCodec.h"

namespace gdcm
{

class GDCM_EXPORT PDFCodec : public Codec
{
public:
  PDFCodec();
  ~PDFCodec() override;
  bool CanCode(TransferSyntax const &) const override { return false; }
  bool CanDecode(TransferSyntax const &) const override { return false; }
  bool Decode(DataElement const &is, DataElement &os) override;
};

} // end namespace gdcm

#endif //GDCMPDFCODEC_H
```

13.369 gdcmPersonName.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <algorithm>
#include <string.h>
```

Include dependency graph for gdcmPersonName.h:



Classes

- class gdcm::PersonName

  PersonName class.

Namespaces

- namespace gdcm

13.370 gdcmPersonName.h

Go to the documentation of this file.

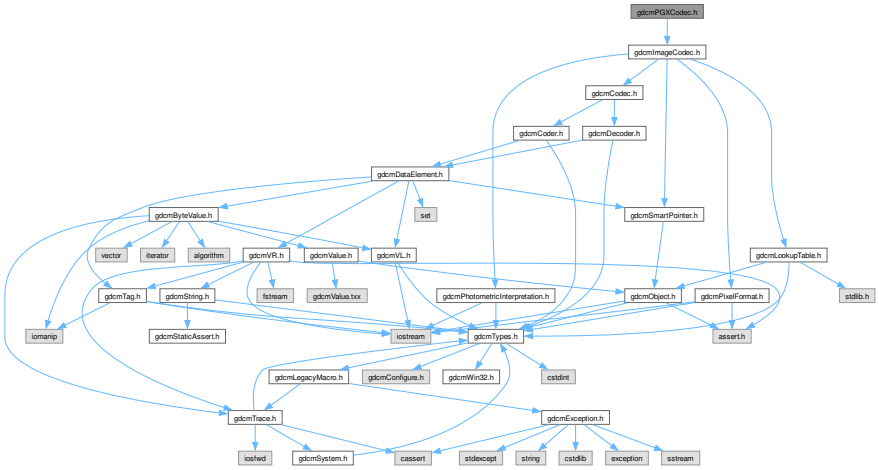
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014
00015 #ifndef GDCMPERSONNAME_H
00016 #define GDCMPERSONNAME_H
00017
00018 #include "gdcmTypes.h"
00019 #include <vector>
00020 #include <algorithm> // std::min
00021 #include <string.h> // strlen
00022
00023 namespace gdcm
00024 {
00025
00029 class GDCM_EXPORT PersonName
00030 {
00031 public:
00032   static const unsigned int MaxNumberOfComponents = 5;
00033   static const unsigned int MaxLength = 64;
00034   char Component[MaxNumberOfComponents][MaxLength+1];
00035   static const char Separator = '^';
00036   static const char Padding   = ' ';
00037
00038   unsigned int GetNumberOfComponents() const {
00039     unsigned int r = 0;
00040     for(unsigned int i = 0; i < 5; ++i) {
00041       if( *Component[i] != '\0' ) r = i;
00042       }
00043     return r+1;
00044   }
00045   unsigned int GetMaxLength() const { return MaxLength; }
00046   void SetBlob(const std::vector<char>& v) {
00047   (void)v;
00048     //assert(0); //TODO
00049   }
00050   void SetComponents(const char *comp1 = "",
00051     const char *comp2 = "",
00052     const char *comp3 = "",
00053     const char *comp4 = "",
00054     const char *comp5 = "") {
00055     const char *components[5] = { comp1, comp2, comp3, comp4, comp5 };
00056     SetComponents( components );
00057   }
00058   void SetComponents(const char *components[]) {
00059     if( components )
00060       for(unsigned int i = 0; i < 5; ++i) {
```

```
        if( components[i] && strlen(components[i]) < GetMaxLength() )
          strcpy(Component[i], components[i]);
        assert( strlen(Component[i]) < GetMaxLength() );
      }
  }
  void Print(std::ostream &os) const
  {
    //os « "Family Name Complex: " « Component[0] « std::endl;
    //os « "Given  Name Complex: " « Component[1] « std::endl;
    //os « "Middle Name        : " « Component[2] « std::endl;
    //os « "Name Suffix        : " « Component[3] « std::endl;
    //os « "Name Prefix        : " « Component[4] « std::endl;
    os « Component[0] « '^';
    os « Component[1] « '^';
    os « Component[2] « '^';
    os « Component[3] « '^';
    os « Component[4];
  }
};

} // end namespace gdcm

#endif //GDCMPERSONNAME_H
```

13.371 gdcmPGXCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmPGXCodec.h:



Classes

- class gdcm::PGXCodec

  Class to do PGX.

Namespaces

- namespace gdcm

13.372 gdcmPGXCodec.h

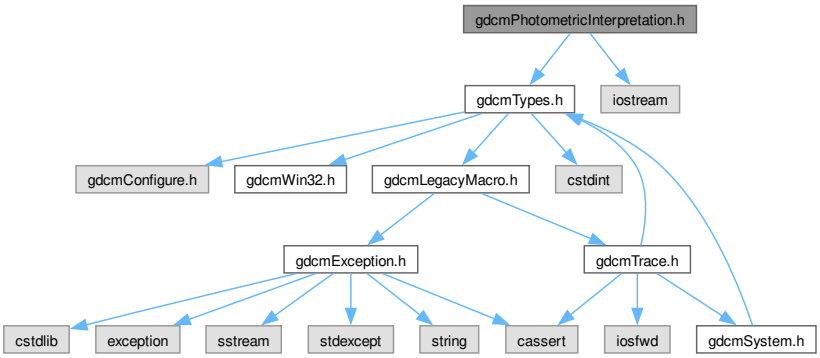
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPGXCODEC_H
#define GDCMPGXCODEC_H

#include "gdcmImageCodec.h"

namespace gdcm
{

class GDCM_EXPORT PGXCodec : public ImageCodec
{
public:
  PGXCodec();
  ~PGXCodec() override;
  bool CanDecode(TransferSyntax const &ts) const override;
  bool CanCode(TransferSyntax const &ts) const override;

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
  ImageCodec * Clone() const override;

  bool Read(const char *filename, DataElement &out) const;
  bool Write(const char *filename, const DataElement &out) const;
private:
};

} // end namespace gdcm

#endif //GDCMPGXCODEC_H
```

13.373 gdcmPhotometricInterpretation.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmPhotometricInterpretation.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::PhotometricInterpretation

  Class to represent an PhotometricInterpretation.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)

13.374 gdcmPhotometricInterpretation.h

Go to the documentation of this file.

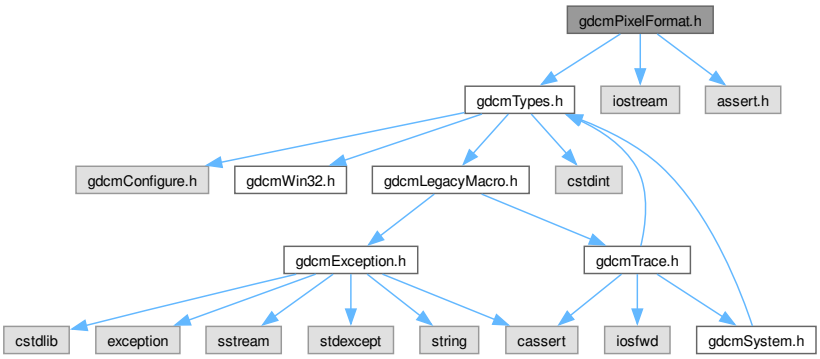
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMPHOTOMETRICINTERPRETATION_H
#define GDCMPHOTOMETRICINTERPRETATION_H

#include "gdcmTypes.h"
#include <iostream>

namespace gdcm
{

class TransferSyntax;
class GDCM_EXPORT PhotometricInterpretation
{
public:
  typedef enum {
    UNKNOWN = 0,
    MONOCHROME1,
    MONOCHROME2,
    PALETTE_COLOR,
    RGB,
```

```
    HSV,
    ARGB, // retired
    CMYK,
    YBR_FULL,
    YBR_FULL_422,
    YBR_PARTIAL_422,
    YBR_PARTIAL_420,
    YBR_ICT,
    YBR_RCT,
    // PALETTE_COLOR ?
    //MONOCHROME = MONOCHROME1 | MONOCHROME2,
    //COLOR      = RGB | HSV | ARGB | CMYK | YBR_FULL | YBR_FULL_422 | YBR_PARTIAL_422 | YBR_PARTIAL_420 | YBR_ICT | YBR_RCT,
    PI_END  // Helpful for internal implementation
  } PIType; // PhotometricInterpretationType

  PhotometricInterpretation(PIType pi = UNKNOWN):PIField(pi) {}

  static const char *GetPIString(PIType pi);

  const char *GetString() const;

  // You need to make sure end of string is \0
  static PIType GetPIType(const char *pi);

  static bool IsRetired(PIType pi);

  bool IsLossy() const;
  bool IsLossless() const;

  unsigned short GetSamplesPerPixel() const;

  // TODO
  // not all PhotometricInterpretation are allowed for compressed Transfer
  // syntax
  // static bool IsAllowedForCompressedTS(PIType pi);

  friend std::ostream& operator«(std::ostream& os, const PhotometricInterpretation& pi);

  operator PIType () const { return PIField; }

  PIType GetType () const { return PIField; }

  // Will return whether current PhotometricInterpretation is the same Color Space as input:
  // eg. RGB and YBR_RCT are
  bool IsSameColorSpace( PhotometricInterpretation const &pi ) const;

  //static PIType GetEquivalent(TransferSyntax const &ts);

private:
  PIType PIField;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream& os, const PhotometricInterpretation &val)
{
  const char *s = PhotometricInterpretation::GetPIString(val.PIField);
  os « (s ? s : "");
  return os;
}


} // end namespace gdcm

#endif //GDCMPHOTOMETRICINTERPRETATION_H
```

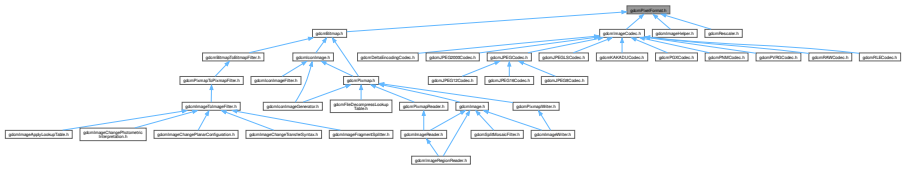
13.375 gdcmPixelFormat.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
#include <assert.h>
```

Include dependency graph for gdcmPixelFormat.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PixelFormat

  PixelFormat.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)

13.376 gdcmPixelFormat.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#ifndef GDCMPIXELFORMAT_H
#define GDCMPIXELFORMAT_H

#include "gdcmTypes.h"
#include <iostream>
#include <assert.h>

namespace gdcm
{

class TransferSyntax;

class GDCM_EXPORT PixelFormat
{
  friend class Bitmap;
  friend std::ostream& operator«(std::ostream &_os, const PixelFormat &pf);
public:
  // When adding a type please add its dual type (its unsigned counterpart)
  typedef enum {
    UINT8,
    INT8,
    UINT12,
    INT12,
    UINT16,
    INT16,
    UINT32,  // For some DICOM files (RT or SC)
    INT32,   //                          "   "
    UINT64,  // Needed when input is 32bits + intercept/slope (incomplete support)
    INT64,   //                          "   "
    FLOAT16, // sure why not...
    FLOAT32, // good ol' 'float'
    FLOAT64, // aka 'double'
    SINGLEBIT, // bool / monochrome
    UNKNOWN // aka BitsAllocated == 0 && PixelRepresentation == 0
  } ScalarType;

  // default cstor:
  PixelFormat () : PixelFormat(1, 8, 8, 7, 0) {}

  explicit PixelFormat (
    unsigned short samplesperpixel,
    unsigned short bitsallocated = 8,
    unsigned short bitsstored = 8,
    unsigned short highbit = 7,
    unsigned short pixelrepresentation = 0 ) :
  SamplesPerPixel(samplesperpixel),
  BitsAllocated(bitsallocated),
  BitsStored(bitsstored),
  HighBit(highbit),
  PixelRepresentation(pixelrepresentation) {}
  // helper, for the common case
  PixelFormat(ScalarType st);

  // For transparency of use
  operator ScalarType() const { return GetScalarType(); }

  unsigned short GetSamplesPerPixel() const;
  void SetSamplesPerPixel(unsigned short spp)
    {
    gdcmAssertMacro( spp <= 4 );
    SamplesPerPixel = spp;
```

```
00096     assert( SamplesPerPixel == 1 || SamplesPerPixel == 3 || SamplesPerPixel == 4 );
00097     }
00098
00100   unsigned short GetBitsAllocated() const
00101     {
00102     return BitsAllocated;
00103     }
00104   void SetBitsAllocated(unsigned short ba)
00105     {
00106     if( ba )
00107       {
00108       switch( ba )
00109         {
00110         /* some devices (FUJIFILM CR + MONO1) incorrectly set BitsAllocated/BitsStored
00111          * as bitmask instead of value. Do what they mean instead of what they say.
00112          */
00113         case 0xffff: ba = 16; break;
00114         case 0x0fff: ba = 12; break;
00115         case 0x00ff: ba =  8; break;
00116         }
00117       BitsAllocated = ba;
00118       BitsStored = ba;
00119       HighBit = (unsigned short)(ba - 1);
00120       }
00121     else // Make the PixelFormat as UNKNOWN
00122       {
00123       BitsAllocated = 0;
00124       PixelRepresentation = 0;
00125       }
00126     }
00127
00129   unsigned short GetBitsStored() const
00130     {
00131     assert( BitsStored <= BitsAllocated );
00132     return BitsStored;
00133     }
00134   void SetBitsStored(unsigned short bs)
00135     {
00136     switch( bs )
00137       {
00138       /* see SetBitsAllocated for explanation
00139        */
00140       case 0xffff: bs = 16; break;
00141       case 0x0fff: bs = 12; break;
00142       case 0x00ff: bs =  8; break;
00143       }
00144     if( bs <= BitsAllocated && bs )
00145       {
00146       BitsStored = bs;
00147       SetHighBit( (unsigned short) (bs - 1) );
00148       }
00149     }
00150
00152   unsigned short GetHighBit() const
00153     {
00154     assert( HighBit < BitsStored );
00155     return HighBit;
00156     }
00157   void SetHighBit(unsigned short hb)
00158     {
00159     switch( hb )
00160       {
00161       /* broken implementations that use bitmask for BitsAllocated/Stored
00162        * nonetheless use (BitsStored-1) for HighBit. correct for this here.
00163        */
00164       case 0xfffe: hb = 15; break;
00165       case 0x0ffe: hb = 11; break;
00166       case 0x00fe: hb =  7; break;
00167       }
00168     if( hb < BitsStored )
00169       HighBit = hb;
00170     }
00171
00173   unsigned short GetPixelRepresentation() const
00174     {
00175     return (unsigned short)(PixelRepresentation ? 1 : 0);
00176     }
00177   void SetPixelRepresentation(unsigned short pr)
00178     {
00179     PixelRepresentation = (unsigned short)(pr ? 1 : 0);
00180     }
```

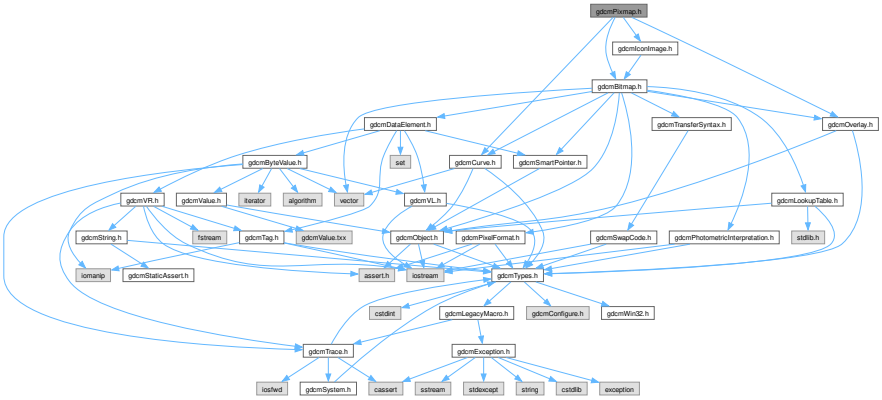
```
  ScalarType GetScalarType() const;

  void SetScalarType(ScalarType st);
  const char *GetScalarTypeAsString() const;

  uint8_t GetPixelSize() const;

  void Print(std::ostream &os) const;

  int64_t GetMin() const;

  int64_t GetMax() const;

  bool IsValid() const;

  bool operator==(ScalarType st) const
    {
    return GetScalarType() == st;
    }
  bool operator!=(ScalarType st) const
    {
    return GetScalarType() != st;
    }
  bool operator==(const PixelFormat &pf) const
    {
    return
      SamplesPerPixel     == pf.SamplesPerPixel &&
      BitsAllocated       == pf.BitsAllocated &&
      BitsStored          == pf.BitsStored &&
      HighBit             == pf.HighBit &&
      PixelRepresentation == pf.PixelRepresentation;
    }
  bool operator!=(const PixelFormat &pf) const
    {
    return
      SamplesPerPixel     != pf.SamplesPerPixel ||
      BitsAllocated       != pf.BitsAllocated ||
      BitsStored          != pf.BitsStored ||
      HighBit             != pf.HighBit ||
      PixelRepresentation != pf.PixelRepresentation;
    }

  bool IsCompatible(const TransferSyntax & ts ) const;
protected:
  bool Validate();

private:
  // D 0028|0002 [US] [Samples per Pixel] [1]
  unsigned short SamplesPerPixel;
  // D 0028|0100 [US] [Bits Allocated] [8]
  unsigned short BitsAllocated;
  // D 0028|0101 [US] [Bits Stored] [8]
  unsigned short BitsStored;
  // D 0028|0102 [US] [High Bit] [7]
  unsigned short HighBit;
  // D 0028|0103 [US] [Pixel Representation] [0]
  unsigned short PixelRepresentation;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const PixelFormat &pf)
{
  pf.Print( os );
  return os;
}

} // end namespace gdcm

#endif //GDCMPIXELFORMAT_H
```

13.377 gdcmPixmap.h File Reference

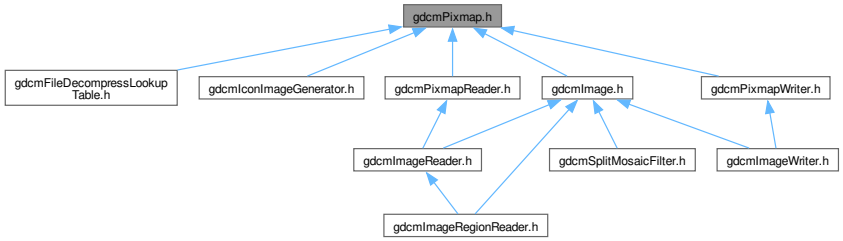
```
#include "gdcmBitmap.h"
#include "gdcmCurve.h"
```

```
#include "gdcmIconImage.h"
#include "gdcmOverlay.h"
```

Include dependency graph for gdcmPixmap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Pixmap

  Pixmap class.

Namespaces

- namespace gdcm

13.378 gdcmPixmap.h

Go to the documentation of this file.

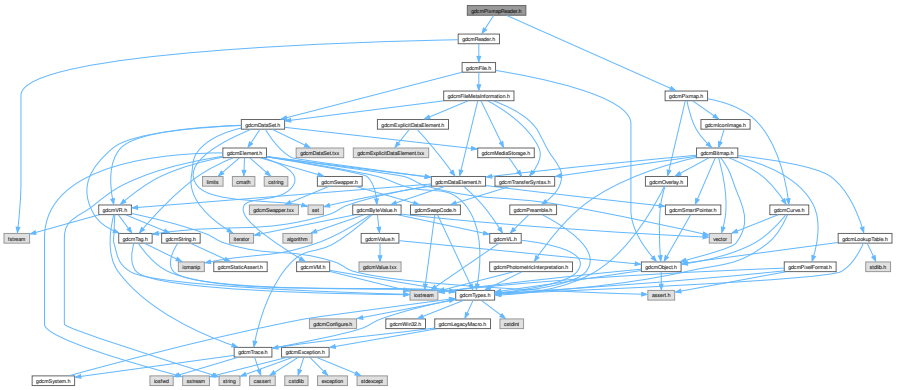
```
00001 /*=========================================================================
00002
```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPIXMAP_H
#define GDCMPIXMAP_H

#include "gdcmBitmap.h"
#include "gdcmCurve.h"
#include "gdcmIconImage.h"
#include "gdcmOverlay.h"

namespace gdcm
{

class GDCM_EXPORT Pixmap : public Bitmap
{
public:
  Pixmap();
  ~Pixmap() override;
  void Print(std::ostream &) const override;

  bool AreOverlaysInPixelData() const override;
  bool UnusedBitsPresentInPixelData() const override;

  Curve& GetCurve(size_t i = 0) {
    assert( i < Curves.size() );
    return Curves[i];
  }
  const Curve& GetCurve(size_t i = 0) const {
    assert( i < Curves.size() );
    return Curves[i];
  }
  size_t GetNumberOfCurves() const { return Curves.size(); }
  void SetNumberOfCurves(size_t n) { Curves.resize(n); }

  Overlay& GetOverlay(size_t i = 0) {
    assert( i < Overlays.size() );
    return Overlays[i];
  }
  const Overlay& GetOverlay(size_t i = 0) const {
    assert( i < Overlays.size() );
    return Overlays[i];
  }
  size_t GetNumberOfOverlays() const { return Overlays.size(); }
  void SetNumberOfOverlays(size_t n) { Overlays.resize(n); }
  void RemoveOverlay(size_t i) {
    assert( i < Overlays.size() );
    Overlays.erase( Overlays.begin() + i );
  }

  const IconImage &GetIconImage() const { return *Icon; }
  IconImage &GetIconImage() { return *Icon; }
  void SetIconImage(IconImage const &ii) { Icon = ii; }

//private:
protected:
  std::vector<Overlay>  Overlays;
  std::vector<Curve>  Curves;
  SmartPointer<IconImage> Icon;
};

} // end namespace gdcm

#endif //GDCMPIXMAP_H
```

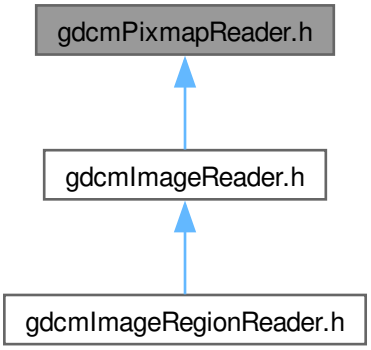
13.379 gdcmPixmapReader.h File Reference

```
#include "gdcmReader.h"
#include "gdcmPixmap.h"
```

Include dependency graph for gdcmPixmapReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PixmapReader

  PixmapReader.

Namespaces

- namespace gdcm

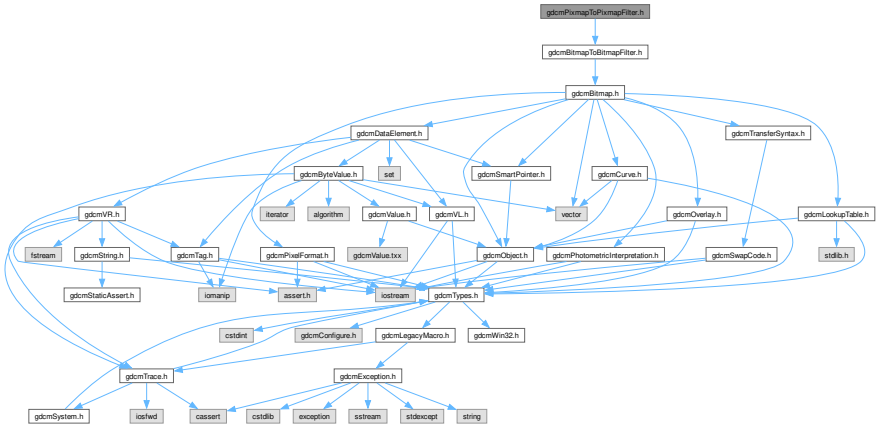
13.380 gdcmPixmapReader.h

Go to the documentation of this file.

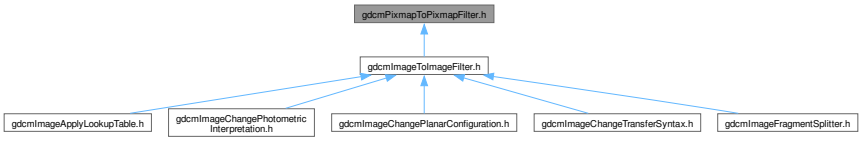
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPIXMAPREADER_H
#define GDCMPIXMAPREADER_H

#include "gdcmReader.h"
#include "gdcmPixmap.h"

namespace gdcm
{

class ByteValue;
class MediaStorage;
class GDCM_EXPORT PixmapReader : public Reader
{
public:
  PixmapReader();
  ~PixmapReader() override; //needs to be virtual to ensure lack of memory leaks


  bool Read() override;

  // Following methods are valid only after a call to 'Read'

  const Pixmap& GetPixmap() const;
  Pixmap& GetPixmap();
  //void SetPixamp(Pixmap const &pix);

protected:
  bool ReadImageInternal(MediaStorage const &ms, bool handlepixeldata = true);
  virtual bool ReadImage(MediaStorage const &ms);
  virtual bool ReadACRNEMAImage();

  SmartPointer<Pixmap> PixelData;
};


} // end namespace gdcm

#endif //GDCMPIXMAPREADER_H
```

13.381 gdcmPixmapToPixmapFilter.h File Reference

`#include "gdcmBitmapToBitmapFilter.h"`
Include dependency graph for gdcmPixmapToPixmapFilter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PixmapToPixmapFilter

  PixmapToPixmapFilter class.

Namespaces

- namespace gdcm

13.382 gdcmPixmapToPixmapFilter.h

Go to the documentation of this file.

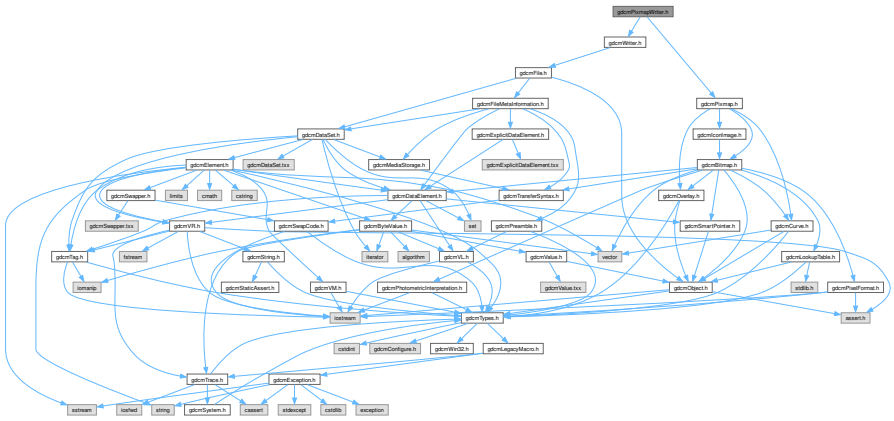
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
```

```
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 ==========================================================================*/
00014 #ifndef GDCMPIXMAPTOPIXMAPFILTER_H
00015 #define GDCMPIXMAPTOPIXMAPFILTER_H
00016
00017 #include "gdcmBitmapToBitmapFilter.h"
00018
00019 namespace gdcm
00020 {
00021
00022 class Pixmap;
00027 class GDCM_EXPORT PixmapToPixmapFilter : public BitmapToBitmapFilter
00028 {
00029 public:
00030   PixmapToPixmapFilter();
00031   ~PixmapToPixmapFilter() = default;
00032
00033   Pixmap &GetInput();
00034
00036   const Pixmap &GetOutput() const;
00037
00038   // SWIG/Java hack:
00039   const Pixmap &GetOutputAsPixmap() const;
00040 };
00041
00042 } // end namespace gdcm
00043
00044 #endif //GDCMPIXMAPTOPIXMAPFILTER_H
```

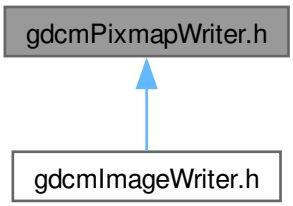
13.383 gdcmPixmapWriter.h File Reference

```
#include "gdcmWriter.h"
#include "gdcmPixmap.h"
```

Include dependency graph for gdcmPixmapWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PixmapWriter

  PixmapWriter.

Namespaces

- namespace gdcm

13.384 gdcmPixmapWriter.h

Go to the documentation of this file.

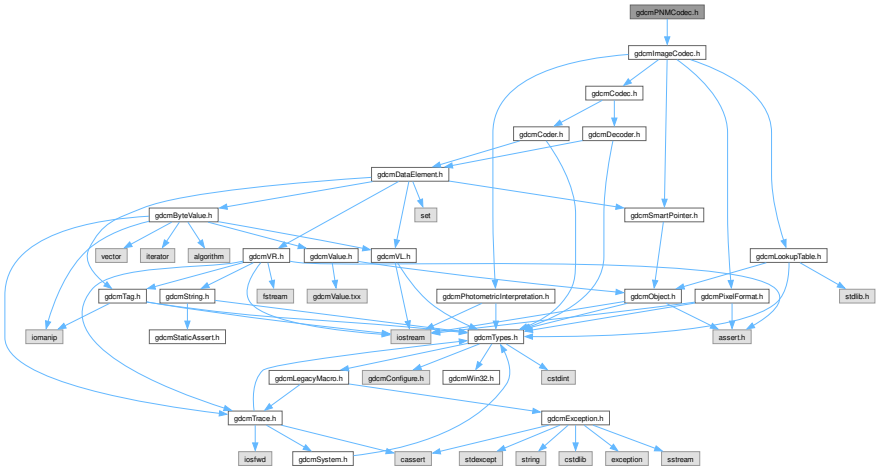
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMPIXMAPWRITER_H
00015 #define GDCMPIXMAPWRITER_H
00016
00017 #include "gdcmWriter.h"
00018 #include "gdcmPixmap.h"
00019
00020 namespace gdcm
00021 {
00022
00023 class StreamImageWriter;
00024 class Pixmap;
00036 class GDCM_EXPORT PixmapWriter : public Writer
00037 {
00038 public:
00039   PixmapWriter();
00040   ~PixmapWriter() override;
00041
```

```
  const Pixmap& GetPixmap() const { return *PixelData; }
  Pixmap& GetPixmap() { return *PixelData; } // FIXME
  void SetPixmap(Pixmap const &img);

  virtual const Pixmap& GetImage() const { return *PixelData; }
  virtual Pixmap& GetImage() { return *PixelData; } // FIXME
  virtual void SetImage(Pixmap const &img);

  bool Write() override; // Execute()

protected:
  void DoIconImage(DataSet & ds, Pixmap const & image);
  bool PrepareWrite( MediaStorage const & refms );

  SmartPointer<Pixmap> PixelData;
};


} // end namespace gdcm

#endif //GDCMPIXMAPWRITER_H
```

13.385 gdcmPNMCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmPNMCodec.h:



Classes

- class gdcm::PNMCodec

  Class to do PNM.

Namespaces

- namespace gdcm

13.386 gdcmPNMCodec.h

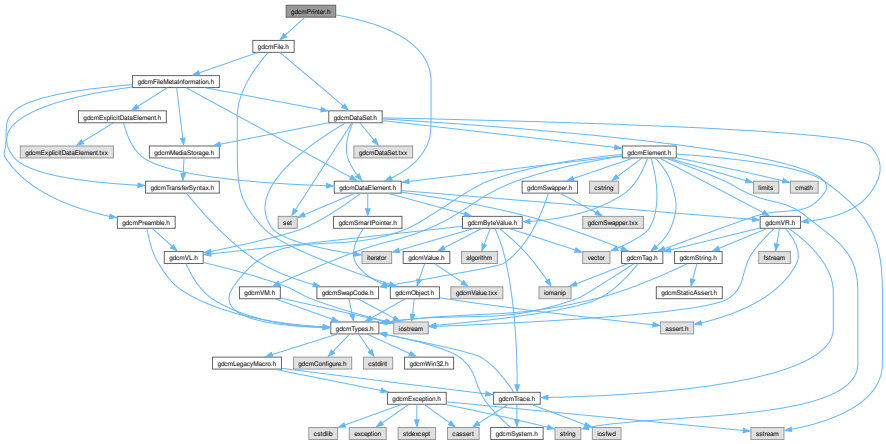
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPNMCODEC_H
#define GDCMPNMCODEC_H

#include "gdcmImageCodec.h"

namespace gdcm
{

class GDCM_EXPORT PNMCodec : public ImageCodec
{
public:
  PNMCodec();
  ~PNMCodec() override;
  bool CanDecode(TransferSyntax const &ts) const override;
  bool CanCode(TransferSyntax const &ts) const override;

  unsigned long GetBufferLength() const { return BufferLength; }
  void SetBufferLength(unsigned long l) { BufferLength = l; }

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
  ImageCodec * Clone() const override;

  bool Read(const char *filename, DataElement &out) const;
  bool Write(const char *filename, const DataElement &out) const;
  //bool Write(const char *filename);
private:
  unsigned long BufferLength;
};

} // end namespace gdcm

#endif //GDCMPNMCODEC_H
```

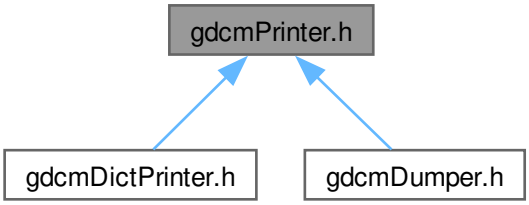
13.387 gdcmPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmPrinter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Printer

  Printer class.

Namespaces

- namespace gdcm

13.388 gdcmPrinter.h

Go to the documentation of this file.

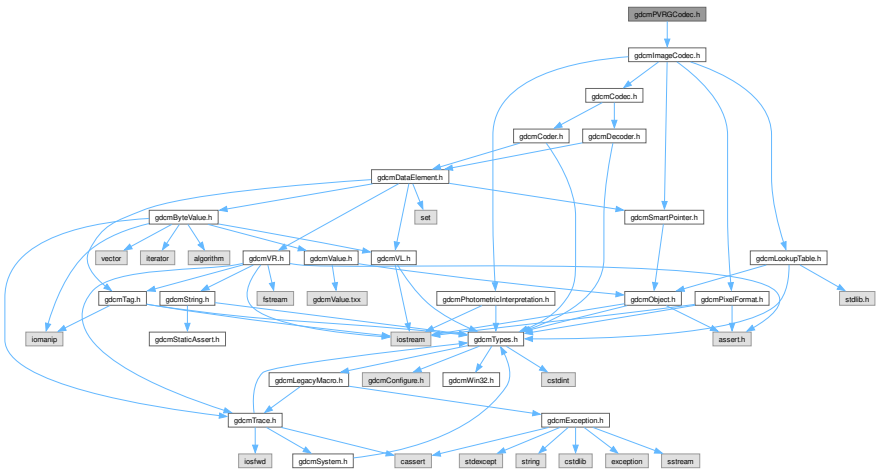
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPRINTER_H
#define GDCMPRINTER_H

// TODO Class to implement printing
// Since DICOM does printing ?
// Also I would like to encapsulate the IsCharacterPrintable thing
// (to avoid printing \0 and other weird characters)
// \todo I still need to implement skipping of group (shadow)
// need to implement longer field to read

/*
 * Output:
 * For ASCII:
 * Typically will look like:
 * [ORIGINAL\PRIMARY\OTHER]
 * If a non printable character is found: RED and INVERSE is used:
 * [                .]
 *
 * when the VR is not found (file or dict), we check if we can print the output:
 * on success ASCII mode is used, on failure the output is printed a series of bytes
 *
 * Special case when the data element is empty:
 * INVERSE « (no value)
 *
 * retired public element are printed in red and underline
 * unknown private element are printed in RED followed by 'UNKNOWN'
 *
 * Correct VR is printed in green just after the found VR
 *
 * length of data element is printed in bytes, followed by the VM, a green VM is appended
 * if this is not compatible
 */
#include "gdcmFile.h"
#include "gdcmDataElement.h"

namespace gdcm
{

class DataSet;
class DictEntry;
class Dicts;
// It's a sink there is no output
class GDCM_EXPORT Printer
{
public:
  Printer();
  ~Printer() = default;

  void SetFile(File const &f) { F = &f; }

  void SetColor(bool c);

  typedef enum {
    VERBOSE_STYLE = 0, // GDCM Legacy VERBOSE one
    CONDENSED_STYLE, //
    // Ok I am missing voc here ...better naming would be nice
    XML, //
    CXX
  } PrintStyles;

  void SetStyle(PrintStyles ps) {
    PrintStyle = ps;
```

```
  }
  PrintStyles GetPrintStyle() const {
    return PrintStyle;
    }

  void Print(std::ostream& os);

  void PrintDataSet(const DataSet &ds, std::ostream& os, const std::string &s = "");

protected:
  VR PrintDataElement(std::ostringstream & os, const Dicts &dicts, const DataSet & ds, const DataElement
&de, std::ostream &out, std::string const & indent );
void PrintSQ(const SequenceOfItems *sqi, std::ostream & os, std::string const & indent);

  PrintStyles PrintStyle;
  const File *F;
  VL MaxPrintLength;
};

} // end namespace gdcm

#endif //GDCMPRINTER_H
```

13.389 gdcmPVRGCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmPVRGCodec.h:



Classes

- class gdcm::PVRGCodec

  PVRGCodec.

Namespaces

- namespace gdcm

13.390 gdcmPVRGCodec.h

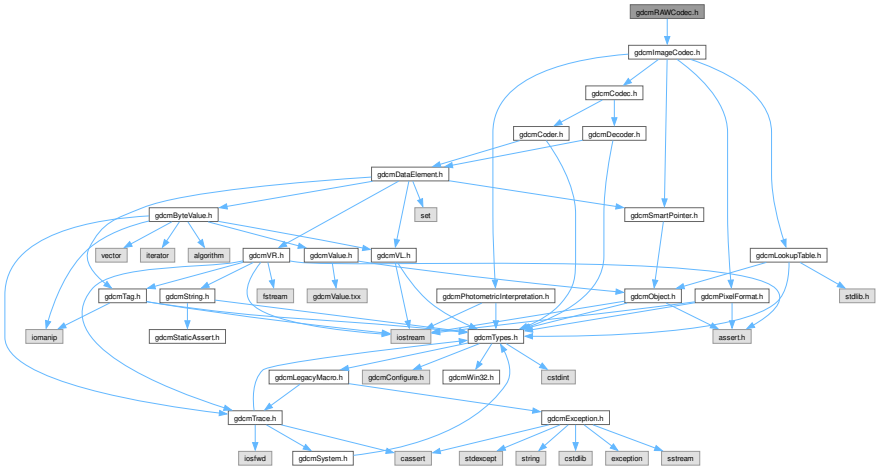
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPVRGCODEC_H
#define GDCMPVRGCODEC_H

#include "gdcmImageCodec.h"

namespace gdcm
{

class PVRGCodec : public ImageCodec
{
public:
  PVRGCodec();
  ~PVRGCodec() override;
  bool CanDecode(TransferSyntax const &ts) const override;
  bool CanCode(TransferSyntax const &ts) const override;

  bool Decode(DataElement const &is, DataElement &os) override;
  bool Code(DataElement const &in, DataElement &out) override;
  void SetLossyFlag( bool l );

  ImageCodec * Clone() const override;
private:
};

} // end namespace gdcm

#endif //GDCMPVRGCODEC_H
```

13.391 gdcmRAWCodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmRAWCodec.h:



Classes

- class gdcm::RAWCodec

  RAWCodec class.

Namespaces

- namespace gdcm

13.392 gdcmRAWCodec.h

Go to the documentation of this file.

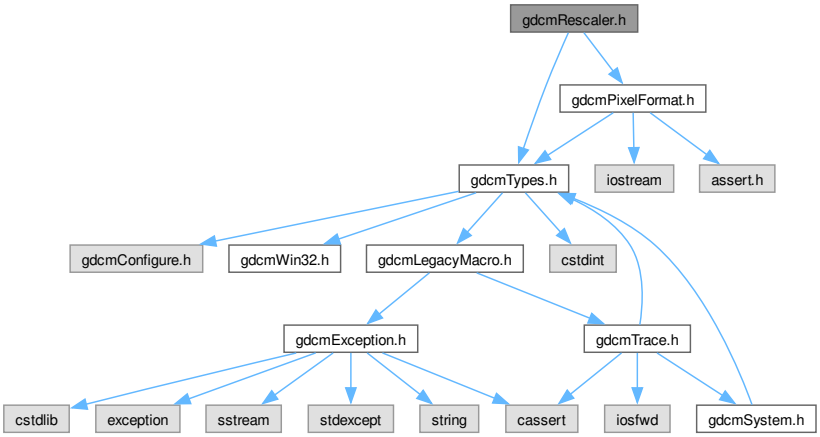
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMRAWCODEC_H
00015 #define GDCMRAWCODEC_H
00016
00017 #include "gdcmImageCodec.h"
00018
00019 namespace gdcm
```

```
{

class RAWInternals;
class GDCM_EXPORT RAWCodec : public ImageCodec
{
public:
  RAWCodec();
  ~RAWCodec() override;
  bool CanCode(TransferSyntax const &ts) const override;
  bool CanDecode(TransferSyntax const &ts) const override;
  bool Decode(DataElement const &is, DataElement &os) override;
  bool Code(DataElement const &in, DataElement &out) override;

  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
  ImageCodec * Clone() const override;

  bool DecodeBytes(const char* inBytes, size_t inBufferLength,
    char* outBytes, size_t inOutBufferLength);

protected:
  bool DecodeByStreams(std::istream &is, std::ostream &os) override;

private:
  RAWInternals *Internals;
};

} // end namespace gdcm

#endif // GDCMRAWCODEC_H
```

13.393 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmRescaler.h:



Classes

- class gdcm::Rescaler

  Rescale class.

Namespaces

- namespace gdcm

13.394 gdcmRescaler.h

Go to the documentation of this file.

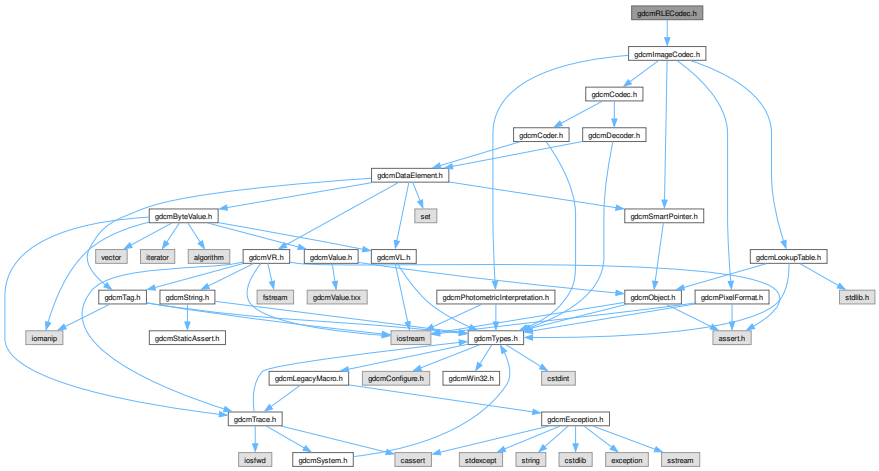
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMRESCALER_H
#define GDCMRESCALER_H

#include "gdcmTypes.h"
#include "gdcmPixelFormat.h"

namespace gdcm
{

class GDCM_EXPORT Rescaler
{
public:
  Rescaler():Intercept(0),Slope(1),PF(PixelFormat::UNKNOWN),TargetScalarType(PixelFormat::UNKNOWN),
ScalarRangeMin(0), ScalarRangeMax(0), UseTargetPixelType(false) {}
  ~Rescaler() = default;

  bool Rescale(char *out, const char *in, size_t n);

  bool InverseRescale(char *out, const char *in, size_t n);

  void SetIntercept(double i) { Intercept = i; }
  double GetIntercept() const { return Intercept; }

  void SetSlope(double s) { Slope = s; }
  double GetSlope() const { return Slope; }

  void SetTargetPixelType( PixelFormat const & targetst );

  void SetUseTargetPixelType(bool b);

  void SetPixelFormat(PixelFormat const & pf) { PF = pf; }

  PixelFormat::ScalarType ComputeInterceptSlopePixelType();

  void SetMinMaxForPixelType(double min, double max);

  PixelFormat ComputePixelTypeFromMinMax();

protected:
  template <typename TIn>
    void RescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n);
  template <typename TIn>
    void InverseRescaleFunctionIntoBestFit(char *out, const TIn *in, size_t n);

private:
  double Intercept; // 0028,1052
  double Slope;     // 0028,1053
  PixelFormat PF;
  PixelFormat::ScalarType TargetScalarType;
  double ScalarRangeMin;
  double ScalarRangeMax;
  bool UseTargetPixelType;
};
```

```
00127
00128 } // end namespace gdcm
00129
00130 #endif //GDCMRESCALER_H
```

13.395 gdcmRLECodec.h File Reference

`#include "gdcmImageCodec.h"`

Include dependency graph for gdcmRLECodec.h:



Classes

- class gdcm::RLECodec

  Class to do RLE.

Namespaces

- namespace gdcm

13.396 gdcmRLECodec.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

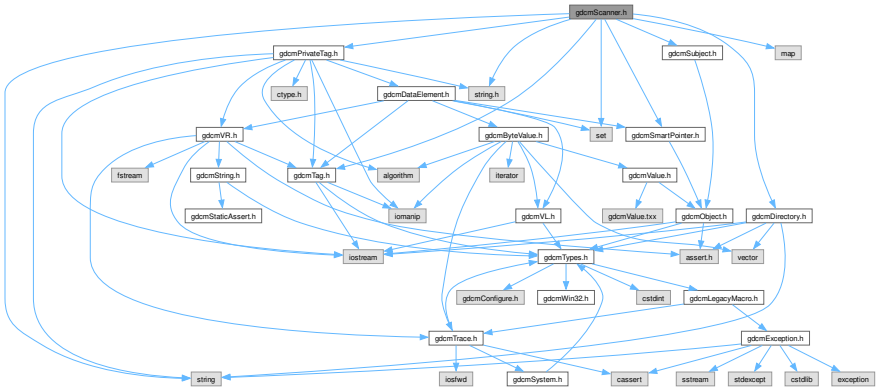
```
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMRLECODEC_H
#define GDCMRLECODEC_H

#include "gdcmImageCodec.h"

namespace gdcm
{

class Fragment;
class RLEInternals;
class GDCM_EXPORT RLECodec : public ImageCodec
{
friend class ImageRegionReader;
public:
  RLECodec();
  ~RLECodec() override;
  bool CanCode(TransferSyntax const &ts) const override;
  bool CanDecode(TransferSyntax const &ts) const override;
  bool Decode(DataElement const &is, DataElement &os) override;
  unsigned long GetBufferLength() const { return BufferLength; }
  void SetBufferLength(unsigned long l) { BufferLength = l; }

  bool Code(DataElement const &in, DataElement &out) override;
  bool GetHeaderInfo(std::istream &is, TransferSyntax &ts) override;
  ImageCodec * Clone() const override;

protected:
  bool DecodeExtent(
    char *buffer,
    unsigned int XMin, unsigned int XMax,
    unsigned int YMin, unsigned int YMax,
    unsigned int ZMin, unsigned int ZMax,
    std::istream & is
  );

  bool DecodeByStreams(std::istream &is, std::ostream &os) override;
public:

  void SetLength(unsigned long l)
    {
    Length = l;
    }

protected:
  bool StartEncode( std::ostream & ) override;
  bool IsRowEncoder() override;
  bool IsFrameEncoder() override;
  bool AppendRowEncode( std::ostream & out, const char * data, size_t datalen ) override;
  bool AppendFrameEncode( std::ostream & out, const char * data, size_t datalen ) override;
  bool StopEncode( std::ostream & ) override;

private:
  bool DecodeByStreamsCommon(std::istream &is, std::ostream &os);
  RLEInternals *Internals;
  unsigned long Length;
  unsigned long BufferLength;
  size_t DecodeFragment(Fragment const & frag, char *buffer, size_t llen);
};

} // end namespace gdcm

#endif //GDCMRLECODEC_H
```

13.397 gdcmScanner.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
```

```
#include <map>
#include <set>
#include <string>
#include <string.h>
```

Include dependency graph for gdcmScanner.h:



Classes

- struct gdcm::Scanner::ltstr
- class gdcm::Scanner

  Scanner.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)

13.398 gdcmScanner.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
```

```
#ifndef GDCMSCANNER_H
#define GDCMSCANNER_H

#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"

#include <map>
#include <set>
#include <string>

#include <string.h> // strcmp

namespace gdcm
{
class StringFilter;

class GDCM_EXPORT Scanner : public Subject
{
  friend std::ostream& operator«(std::ostream &_os, const Scanner &s);
public:
  Scanner():Values(),Filenames(),Mappings() {}
  ~Scanner() override;

  typedef std::map<Tag, const char*> TagToValue;
  //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
  //typedef TagToStringMap TagToValue;
  typedef TagToValue::value_type TagToValueValueType;

  void AddTag( Tag const & t );
  void ClearTags();

  // Work in progress do not use:
  void AddPrivateTag( PrivateTag const & t );

  void AddSkipTag( Tag const & t );
  void ClearSkipTags();

  bool Scan( Directory::FilenamesType const & filenames );

  Directory::FilenamesType const &GetFilenames() const { return Filenames; }

  void Print( std::ostream & os ) const override;

  void PrintTable( std::ostream & os ) const;

  bool IsKey( const char * filename ) const;

  Directory::FilenamesType GetKeys() const;

  // struct to store all the values found:
  typedef std::set< std::string > ValuesType;

  ValuesType const & GetValues() const { return Values; }

  ValuesType GetValues(Tag const &t) const;

  Directory::FilenamesType GetOrderedValues(Tag const &t) const;

  /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
  struct ltstr
    {
    bool operator()(const char* s1, const char* s2) const
      {
      assert( s1 && s2 );
      return strcmp(s1, s2) < 0;
      }
    };
  typedef std::map<const char *,TagToValue, ltstr> MappingType;
  typedef MappingType::const_iterator ConstIterator;
  ConstIterator Begin() const { return Mappings.begin(); }
  ConstIterator End() const { return Mappings.end(); }

  MappingType const & GetMappings() const { return Mappings; }

  TagToValue const & GetMapping(const char *filename) const;

  const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;

```

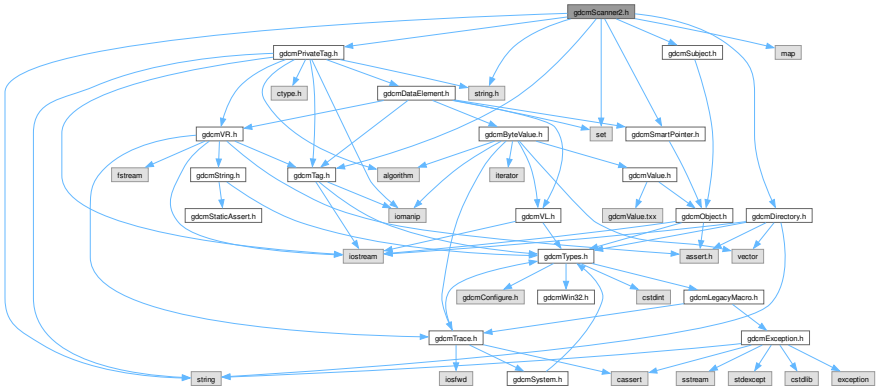
```
  Directory::FilenamesType GetAllFilenamesFromTagToValue(Tag const &t, const char *valueref) const;

  // by a call to GetMapping()
  TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;

  const char* GetValue(const char *filename, Tag const &t) const;

  static SmartPointer<Scanner> New() { return new Scanner; }

protected:
  void ProcessPublicTag(StringFilter &sf, const char *filename);
private:
  // struct to store all uniq tags in ascending order:
  typedef std::set< Tag > TagsType;
  typedef std::set< PrivateTag > PrivateTagsType;
  std::set< Tag > Tags;
  std::set< PrivateTag > PrivateTags;
  std::set< Tag > SkipTags;
  ValuesType Values;
  Directory::FilenamesType Filenames;

  // Main struct that will hold all mapping:
  MappingType Mappings;

  double Progress;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Scanner &s)
{
  s.Print( os );
  return os;
}

#if defined(SWIGPYTHON) || defined(SWIGCSHARP) || defined(SWIGJAVA) || defined(SWIGPHP)
/*
 * HACK: I need this temp class to be able to manipulate a std::map from python,
 * swig does not support wrapping of simple class like std::map...
 */
class SWIGTagToValue
{
public:
  SWIGTagToValue(Scanner::TagToValue const &t2v):Internal(t2v),it(t2v.begin()) {}
  const Scanner::TagToValueValueType& GetCurrent() const { return *it; }
  const Tag& GetCurrentTag() const { return it->first; }
  const char *GetCurrentValue() const { return it->second; }
  void Start() { it = Internal.begin(); }
  bool IsAtEnd() const { return it == Internal.end(); }
  void Next() { ++it; }
private:
  const Scanner::TagToValue& Internal;
  Scanner::TagToValue::const_iterator it;
};
#endif /* SWIG */


} // end namespace gdcm

#endif //GDCMSCANNER_H
```

13.399 gdcmScanner2.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>
```

`#include <string.h>`

Include dependency graph for gdcmScanner2.h:



Classes

- struct gdcm::Scanner2::ltstr
- class gdcm::Scanner2

  Scanner2.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner2 &s)

13.400 gdcmScanner2.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMSCANNER2_H
00015 #define GDCMSCANNER2_H
00016
00017 #include "gdcmDirectory.h"
```

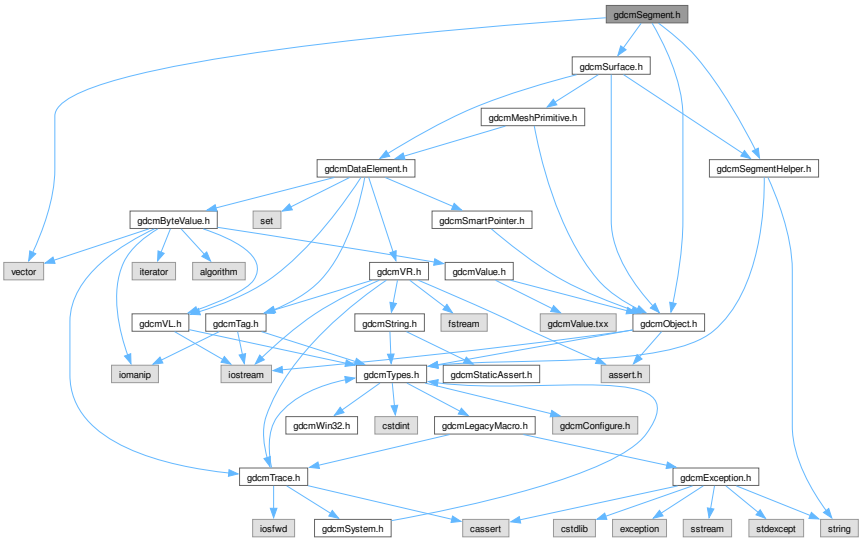
```
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"

#include <map>
#include <set>
#include <string>

#include <string.h> // strcmp

namespace gdcm
{
class StringFilter;

class GDCM_EXPORT Scanner2 : public Subject
{
  friend std::ostream& operator«(std::ostream &_os, const Scanner2 &s);
public:
  Scanner2():Values(),Filenames(),PublicMappings(),PrivateMappings(),Progress(0.0) {}
  ~Scanner2() override;

  typedef std::map<Tag, const char*> PublicTagToValue;
  typedef PublicTagToValue::value_type PublicTagToValueValueType;

  typedef std::map<PrivateTag, const char*> PrivateTagToValue;
  typedef PrivateTagToValue::value_type PrivateTagToValueValueType;

  bool AddPublicTag( Tag const & t );
  void ClearPublicTags();

  // Work in progress do not use:
  bool AddPrivateTag( PrivateTag const & pt );
  void ClearPrivateTags();

  bool AddSkipTag( Tag const & t );
  void ClearSkipTags();

  bool Scan( Directory::FilenamesType const & filenames );

  Directory::FilenamesType const &GetFilenames() const { return Filenames; }

  void Print( std::ostream & os ) const override;

  void PrintTable( std::ostream & os, bool header = false ) const;

  bool IsKey( const char * filename ) const;

  Directory::FilenamesType GetKeys() const;

  // struct to store all the values found:
  typedef std::set< std::string > ValuesType;

  ValuesType const & GetValues() const { return Values; }

  ValuesType GetPublicValues(Tag const &t) const;

  ValuesType GetPrivateValues(PrivateTag const &pt) const;

  Directory::FilenamesType GetPublicOrderedValues(Tag const &t) const;

  Directory::FilenamesType GetPrivateOrderedValues(PrivateTag const &pt) const;

  /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
  struct ltstr
    {
    bool operator()(const char* s1, const char* s2) const
      {
      assert( s1 && s2 );
      return strcmp(s1, s2) < 0;
      }
    };
  typedef std::map<const char *,PublicTagToValue, ltstr> PublicMappingType;
  typedef PublicMappingType::const_iterator PublicConstIterator;
  PublicConstIterator Begin() const { return PublicMappings.begin(); }
  PublicConstIterator End() const { return PublicMappings.end(); }

  typedef std::map<const char *,PrivateTagToValue, ltstr> PrivateMappingType;
  typedef PrivateMappingType::const_iterator PrivateConstIterator;
  PrivateConstIterator PrivateBegin() const { return PrivateMappings.begin(); }
  PrivateConstIterator PrivateEnd() const { return PrivateMappings.end(); }
```

```
00144
00146   PublicMappingType const & GetPublicMappings() const { return PublicMappings; }
00147   PrivateMappingType const & GetPrivateMappings() const { return PrivateMappings; }
00148
00150   PublicTagToValue const & GetPublicMapping(const char *filename) const;
00151   PrivateTagToValue const & GetPrivateMapping(const char *filename) const;
00152
00155   const char *GetFilenameFromPublicTagToValue(Tag const &t, const char *valueref) const;
00156   const char *GetFilenameFromPrivateTagToValue(PrivateTag const &pt, const char *valueref) const;
00157
00160   Directory::FilenamesType GetAllFilenamesFromPublicTagToValue(Tag const &t, const char *valueref) const;
00161   Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue(PrivateTag const &pt, const char
      *valueref) const;
00162
00164   // by a call to GetMapping()
00165   PublicTagToValue const & GetMappingFromPublicTagToValue(Tag const &t, const char *value) const;
00166   PrivateTagToValue const & GetMappingFromPrivateTagToValue(PrivateTag const &pt, const char *value)
      const;
00167
00173   const char* GetPublicValue(const char *filename, Tag const &t) const;
00174   const char* GetPrivateValue(const char *filename, PrivateTag const &t) const;
00175
00177   static SmartPointer<Scanner2> New() { return new Scanner2; }
00178
00179 protected:
00180   void ProcessPublicTag(StringFilter &sf, const char *filename);
00181   void ProcessPrivateTag(StringFilter &sf, const char *filename);
00182 private:
00183   // struct to store all uniq tags in ascending order:
00184   typedef std::set< Tag > PublicTagsType;
00185   typedef std::set< PrivateTag > PrivateTagsType;
00186   std::set< Tag > PublicTags; // Public and Private Creator
00187   std::set< PrivateTag > PrivateTags; // Only Private (no Private Creator)
00188   std::set< Tag > SkipTags;
00189   ValuesType Values;
00190   Directory::FilenamesType Filenames;
00191
00192   // Main struct that will hold all public mapping:
00193   PublicMappingType PublicMappings;
00194   // Main struct that will hold all private mapping:
00195   PrivateMappingType PrivateMappings;
00196
00197   double Progress;
00198 };
00199 //-----------------------------------------------------------------------------
00200 inline std::ostream& operator«(std::ostream &os, const Scanner2 &s)
00201 {
00202   s.Print( os );
00203   return os;
00204 }
00205
00206 } // end namespace gdcm
00207
00208 #endif //GDCMSCANNER2_H
```

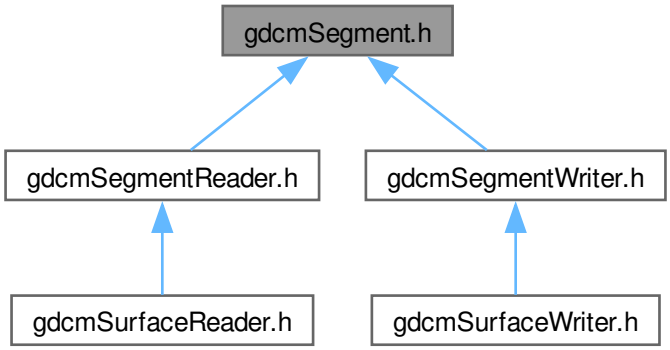
13.401 gdcmSegment.h File Reference

```
#include <vector>
#include <gdcmObject.h>
#include <gdcmSurface.h>
#include "gdcmSegmentHelper.h"
```

Include dependency graph for gdcmSegment.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Segment

  This class defines a segment.

Namespaces

- namespace gdcm

13.402 gdcmSegment.h

Go to the documentation of this file.

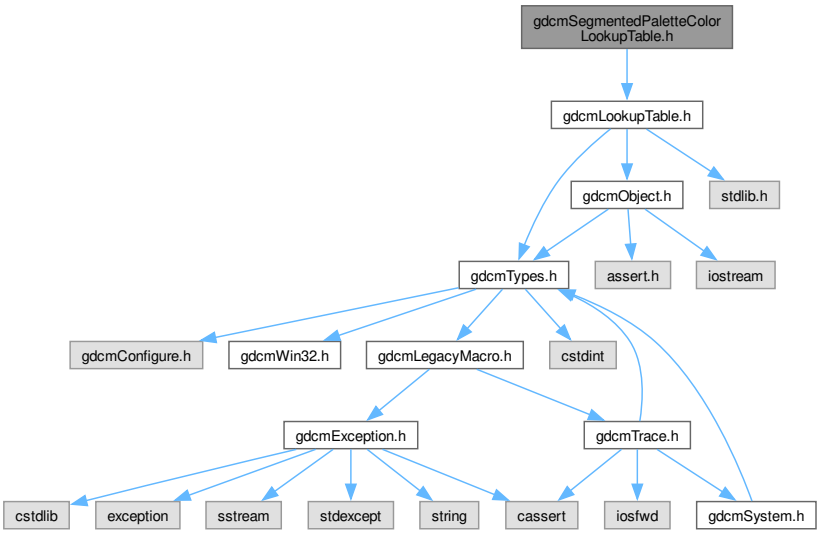
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSEGMENT_H
#define GDCMSEGMENT_H

#include <vector>

#include <gdcmObject.h>
#include <gdcmSurface.h>
#include "gdcmSegmentHelper.h"

namespace gdcm
{

class GDCM_EXPORT Segment : public Object
{
public:

  typedef std::vector< SmartPointer< Surface > > SurfaceVector;
  typedef std::vector< SegmentHelper::BasicCodedEntry > BasicCodedEntryVector;

  typedef enum {
    AUTOMATIC = 0,
    SEMIAUTOMATIC,
    MANUAL,
    ALGOType_END
  } ALGOType;

  static const char * GetALGOTypeString(ALGOType type);
  static ALGOType GetALGOType(const char * type);


  Segment();

  ~Segment() override;

  //**        Segment getters/setters     **//
  unsigned short GetSegmentNumber() const;
  void SetSegmentNumber(const unsigned short num);

  const char * GetSegmentLabel() const;
  void SetSegmentLabel(const char * label);

  const char * GetSegmentDescription() const;
  void SetSegmentDescription(const char * description);

  SegmentHelper::BasicCodedEntry const & GetAnatomicRegion() const;
  SegmentHelper::BasicCodedEntry & GetAnatomicRegion();
  void SetAnatomicRegion(SegmentHelper::BasicCodedEntry const & BSE);

  BasicCodedEntryVector const & GetAnatomicRegionModifiers() const;
  BasicCodedEntryVector & GetAnatomicRegionModifiers();
  void SetAnatomicRegionModifiers(BasicCodedEntryVector const & BSEV);

  SegmentHelper::BasicCodedEntry const & GetPropertyCategory() const;
  SegmentHelper::BasicCodedEntry & GetPropertyCategory();
  void SetPropertyCategory(SegmentHelper::BasicCodedEntry const & BSE);

  SegmentHelper::BasicCodedEntry const & GetPropertyType() const;
  SegmentHelper::BasicCodedEntry & GetPropertyType();
  void SetPropertyType(SegmentHelper::BasicCodedEntry const & BSE);

  BasicCodedEntryVector const & GetPropertyTypeModifiers() const;
  BasicCodedEntryVector & GetPropertyTypeModifiers();
```

```
  void SetPropertyTypeModifiers(BasicCodedEntryVector const & BSEV);

  ALGOType GetSegmentAlgorithmType() const;
  void SetSegmentAlgorithmType(ALGOType type);
  void SetSegmentAlgorithmType(const char * typeStr);

  const char * GetSegmentAlgorithmName() const;
  void SetSegmentAlgorithmName(const char * name);

  //**        Surface getters/setters     **//
  unsigned long GetSurfaceCount();
  void SetSurfaceCount(const unsigned long nb);

  SurfaceVector const & GetSurfaces() const;
  SurfaceVector & GetSurfaces();

  SmartPointer< Surface > GetSurface(const unsigned int idx = 0) const;

  void AddSurface(SmartPointer< Surface > surface);

protected :
  //**        Segment members     **//
  //0062 0004 US 1 Segment Number
  unsigned short  SegmentNumber;
  //0062 0005 LO 1 Segment Label
  std::string     SegmentLabel;
  //0062 0006 ST 1 Segment Description
  std::string     SegmentDescription;

  // General Anatomic Region
  SegmentHelper::BasicCodedEntry AnatomicRegion;
  // General Anatomic Region Modifier
  BasicCodedEntryVector AnatomicRegionModifiers;
  // Property Category Code
  SegmentHelper::BasicCodedEntry PropertyCategory;
  // Property Type Code
  SegmentHelper::BasicCodedEntry PropertyType;
  // Property Type Modifier Code
  BasicCodedEntryVector PropertyTypeModifiers;

  //0062 0008 CS 1 Segment Algorithm Type
  ALGOType        SegmentAlgorithmType;
  //0062 0009 LO 1 Segment Algorithm Name
  std::string     SegmentAlgorithmName;

  //**        Surface members     **//
  //0066 002a UL 1 Surface Count
  unsigned long   SurfaceCount;

  SurfaceVector   Surfaces;

private :
  void ComputeSurfaceCount();
};

}

#endif // GDCMSEGMENT_H
```

13.403 gdcmSegmentedPaletteColorLookupTable.h File Reference

`#include "gdcmLookupTable.h"`

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



Classes

- class gdcm::SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.

Namespaces

- namespace gdcm

13.404 gdcmSegmentedPaletteColorLookupTable.h

Go to the documentation of this file.

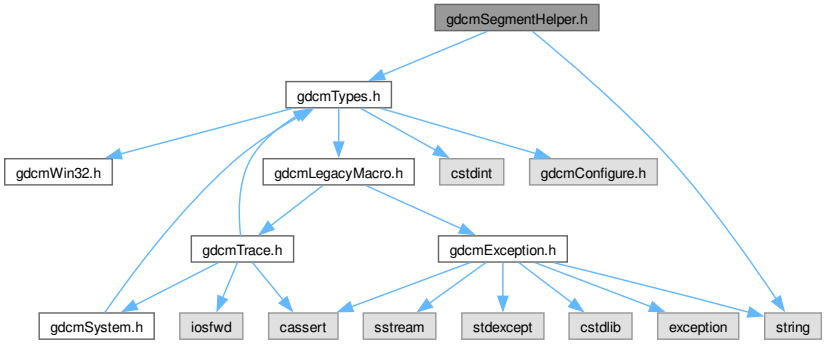
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
```

```
#ifndef GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H
#define GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H

#include "gdcmLookupTable.h"

namespace gdcm
{

class GDCM_EXPORT SegmentedPaletteColorLookupTable : public LookupTable
{
public:
  SegmentedPaletteColorLookupTable();
  ~SegmentedPaletteColorLookupTable() override;
  void Print(std::ostream &) const override {}

  void SetLUT(LookupTableType type, const unsigned char *array,
    unsigned int length) override;

};

} // end namespace gdcm

#endif //GDCMSEGMENTEDPALETTECOLORLOOKUPTABLE_H
```

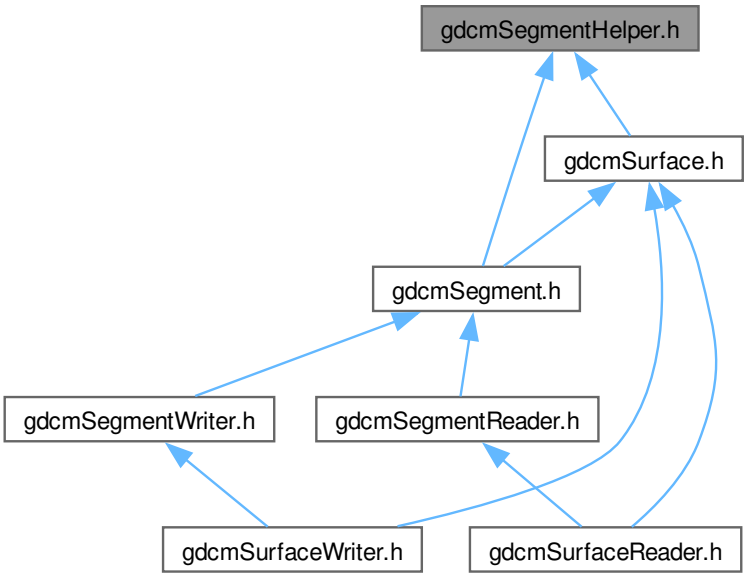
13.405 gdcmSegmentHelper.h File Reference

```
#include "gdcmTypes.h"
#include <string>
```

Include dependency graph for gdcmSegmentHelper.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct gdcm::SegmentHelper::BasicCodedEntry

  This structure defines a basic coded entry with all of its attributes.

Namespaces

- namespace gdcm
- namespace gdcm::SegmentHelper

13.406 gdcmSegmentHelper.h

Go to the documentation of this file.

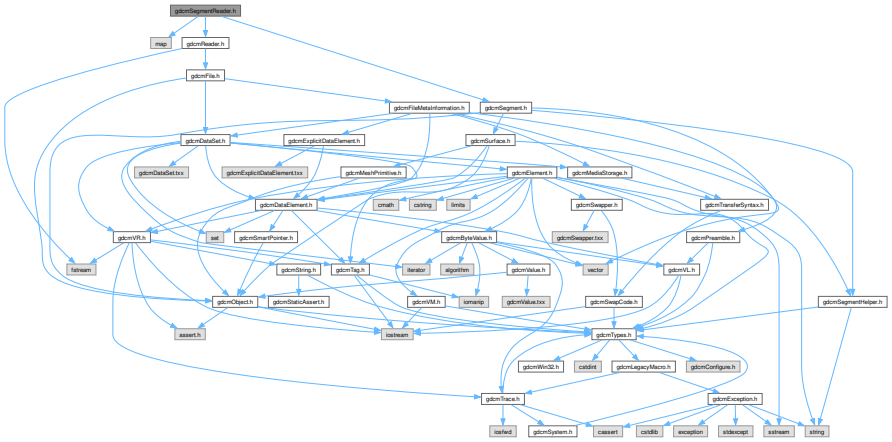
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

```

```
==========================================================================*/
#ifndef GDCMSEGMENTHELPER_H
#define GDCMSEGMENTHELPER_H

#include "gdcmTypes.h"

#include <string>

namespace gdcm
{

namespace SegmentHelper
{

struct GDCM_EXPORT BasicCodedEntry
{
  BasicCodedEntry():
    CV(""),
    CSD(""),
    CSV(""),
    CM("")
  {}

  BasicCodedEntry(const char * a_CV,
                  const char * a_CSD,
                  const char * a_CM):
    CV(a_CV),
    CSD(a_CSD),
    CSV(""),
    CM(a_CM)
  {}

  BasicCodedEntry(const char * a_CV,
                  const char * a_CSD,
                  const char * a_CSV,
                  const char * a_CM):
    CV(a_CV),
    CSD(a_CSD),
    CSV(a_CSV),
    CM(a_CM)
  {}

  bool IsEmpty(const bool checkOptionalAttributes = false) const;


  //**      Members     **//
  // 0008 0100 1   Code Value
  std::string CV;
  // 0008 0102 1   Coding Scheme Designator
  std::string CSD;
  // 0008 0103 1C  Coding Scheme Version
  std::string CSV;
  // 0008 0104 1   Code Meaning
  std::string CM;
};

} // end of SegmentHelper namespace

} // end of gdcm namespace

#endif // GDCMSEGMENTHELPER_H
```

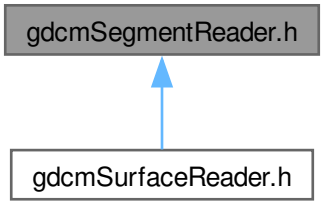
13.407 gdcmSegmentReader.h File Reference

```
#include <map>
#include <gdcmReader.h>
#include <gdcmSegment.h>
```

Include dependency graph for gdcmSegmentReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::SegmentReader

  This class defines a segment reader.

Namespaces

- namespace gdcm

13.408 gdcmSegmentReader.h

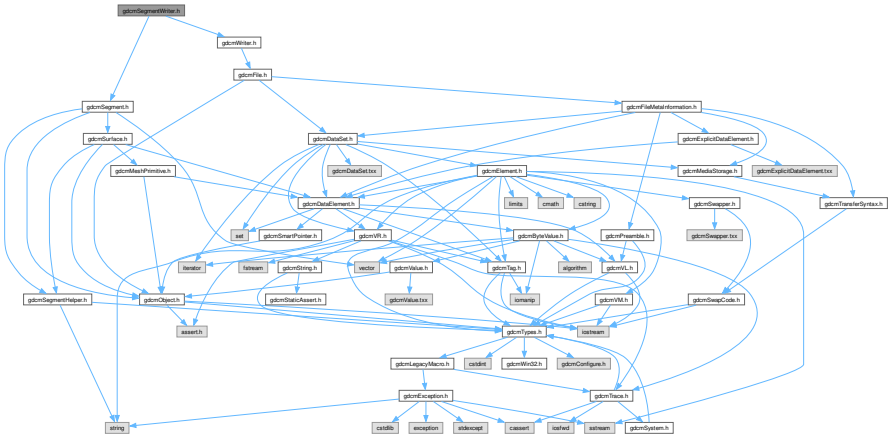
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSEGMENTREADER_H
#define GDCMSEGMENTREADER_H

#include <map>

#include <gdcmReader.h>
#include <gdcmSegment.h>

namespace gdcm
{

class GDCM_EXPORT SegmentReader : public Reader
{
public:
  typedef std::vector< SmartPointer< Segment > > SegmentVector;

  SegmentReader();

  ~SegmentReader() override;

  bool Read() override; // Set to protected ?

  //**        Segment getters/setters     **//
  SegmentVector GetSegments() const;
  SegmentVector GetSegments();

//  unsigned int GetNumberOfSegments();

protected:

  typedef std::map< unsigned long, SmartPointer< Segment > > SegmentMap;

  bool ReadSegments();

  bool ReadSegment(const Item & segmentItem, const unsigned int idx);


  SegmentMap Segments;  // The key value is item number (in segment sequence)
                        // or the surface number (for a surface segmentation).

};

}

#endif // GDCMSEGMENTREADER_H
```

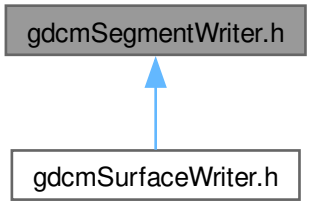
13.409 gdcmSegmentWriter.h File Reference

```
#include <gdcmWriter.h>
#include <gdcmSegment.h>
```

Include dependency graph for gdcmSegmentWriter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::SegmentWriter

  This class defines a segment writer.

Namespaces

- namespace gdcm

13.410 gdcmSegmentWriter.h

Go to the documentation of this file.

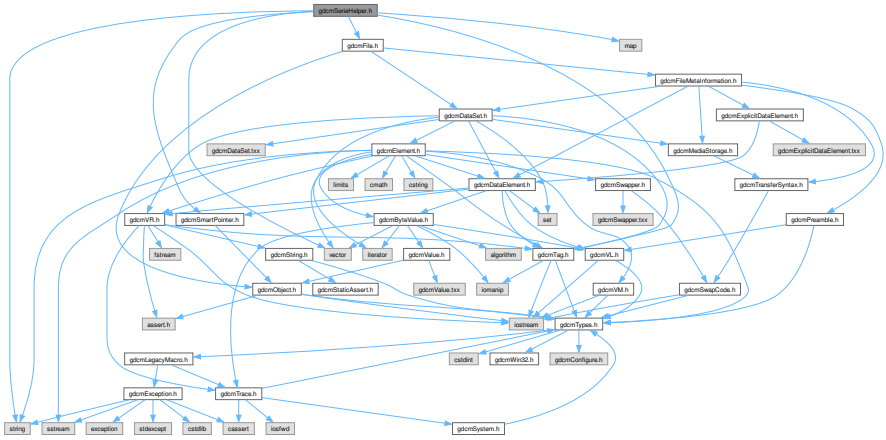
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSEGMENTWRITER_H
#define GDCMSEGMENTWRITER_H

#include <gdcmWriter.h>
#include <gdcmSegment.h>

namespace gdcm
{

class GDCM_EXPORT SegmentWriter : public Writer
{
public:
  typedef std::vector< SmartPointer< Segment > > SegmentVector;

  SegmentWriter();

  ~SegmentWriter() override;

  bool Write() override; // Set to protected ?

  //**        Segment getters/setters     **//
  unsigned int GetNumberOfSegments() const;
  void SetNumberOfSegments(const unsigned int size);

  const SegmentVector & GetSegments() const;
  SegmentVector & GetSegments();
  SmartPointer< Segment > GetSegment(const unsigned int idx = 0) const;

  void AddSegment(SmartPointer< Segment > segment);

  void SetSegments(SegmentVector & segments);

protected:

  bool PrepareWrite();


  SegmentVector Segments;
};

}

#endif // GDCMSEGMENTWRITER_H
```

13.411 gdcmSerieHelper.h File Reference

```
#include "gdcmTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmFile.h"
#include <vector>
#include <string>
```

```
#include <map>
```

Include dependency graph for gdcmSerieHelper.h:



Classes

- class gdcm::FileWithName

  FileWithName.
- class gdcm::SerieHelper

  SerieHelper DO NOT USE this class, it is only a temporary solution for ITK migration from GDCM 1.x to GDCM 2.x It will disappear soon, you've been warned.

Namespaces

- namespace gdcm

Typedefs

- typedef bool(∗ gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER) (File ∗, File ∗)
- typedef std::vector< SmartPointer< FileWithName > > gdcm::FileList

Enumerations

- enum gdcm::CompOperators {
  gdcm::GDCM_EQUAL = 0 ,
  gdcm::GDCM_DIFFERENT ,
  gdcm::GDCM_GREATER ,
  gdcm::GDCM_GREATEROREQUAL ,
  gdcm::GDCM_LESS ,
  gdcm::GDCM_LESSOREQUAL }
- enum gdcm::LodModeType {
  gdcm::LD_ALL = 0x00000000 ,
  gdcm::LD_NOSEQ = 0x00000001 ,
  gdcm::LD_NOSHADOW = 0x00000002 ,
  gdcm::LD_NOSHADOWSEQ = 0x00000004 }

13.412 gdcmSerieHelper.h

Go to the documentation of this file.

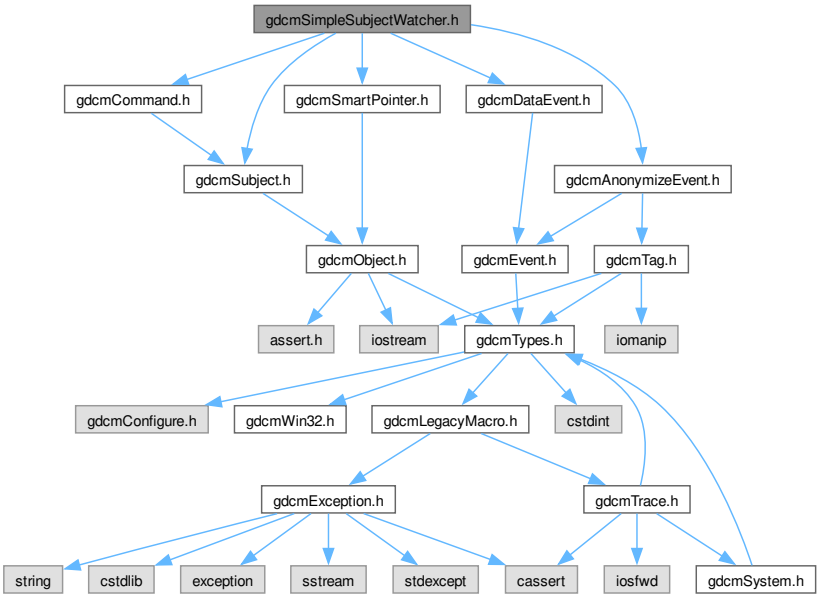
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSERIEHELPER_H
#define GDCMSERIEHELPER_H

#include "gdcmTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmFile.h"
#include <vector>
#include <string>
#include <map>

namespace gdcm
{

enum CompOperators {
   GDCM_EQUAL = 0,
   GDCM_DIFFERENT,
   GDCM_GREATER,
   GDCM_GREATEROREQUAL,
   GDCM_LESS,
   GDCM_LESSOREQUAL
};
enum LodModeType
{
   LD_ALL         = 0x00000000,
   LD_NOSEQ       = 0x00000001,
   LD_NOSHADOW    = 0x00000002,
   LD_NOSHADOWSEQ = 0x00000004
};


class GDCM_EXPORT FileWithName : public File
{
public:
  FileWithName(File &f):File(f),filename(){}
  std::string filename;
};

typedef std::vector< SmartPointer<FileWithName> > FileList;
typedef bool (*BOOL_FUNCTION_PFILE_PFILE_POINTER)(File *, File *);
class Scanner;

class GDCM_EXPORT SerieHelper
{
public:
  SerieHelper();
  ~SerieHelper();

  void Clear();
  void SetLoadMode (int ) {}
  void SetDirectory(std::string const &dir, bool recursive=false);

  void AddRestriction(const std::string & tag);
  void SetUseSeriesDetails( bool useSeriesDetails );
  void CreateDefaultUniqueSeriesIdentifier();
  FileList *GetFirstSingleSerieUIDFileSet();
  FileList *GetNextSingleSerieUIDFileSet();
  std::string CreateUniqueSeriesIdentifier( File * inFile );
  void OrderFileList(FileList *fileSet);
  void AddRestriction(uint16_t group, uint16_t elem, std::string const &value, int op);

protected:
  bool UserOrdering(FileList *fileSet);
```

```
  void AddFileName(std::string const &filename);
  bool AddFile(FileWithName &header);
  void AddRestriction(const Tag& tag);
  bool ImagePositionPatientOrdering(FileList *fileSet);
  bool ImageNumberOrdering( FileList *fileList );
  bool FileNameOrdering( FileList *fileList );

  using Rule = struct RuleStructure{
    uint16_t group;
    uint16_t elem;
    std::string value;
    int op;
  };
  typedef std::vector<Rule> SerieRestrictions;

  typedef std::map<std::string, FileList *> SingleSerieUIDFileSetmap;
  SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT;
  SingleSerieUIDFileSetmap::iterator ItFileSetHt;

private:
  SerieRestrictions Restrictions;
  SerieRestrictions Refine;

  bool UseSeriesDetails;
  bool DirectOrder;

  BOOL_FUNCTION_PFILE_PFILE_POINTER UserLessThanFunction;
};

// backward compat
} // end namespace gdcm


#endif //GDCMSERIEHELPER_H
```

13.413 gdcmSimpleSubjectWatcher.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmCommand.h"
#include "gdcmSmartPointer.h"
#include "gdcmAnonymizeEvent.h"
#include "gdcmDataEvent.h"
```

Include dependency graph for gdcmSimpleSubjectWatcher.h:



Classes

- class gdcm::SimpleSubjectWatcher

  SimpleSubjectWatcher.

Namespaces

- namespace gdcm

13.414 gdcmSimpleSubjectWatcher.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSIMPLESUBJECTWATCHER_H
#define GDCMSIMPLESUBJECTWATCHER_H

```

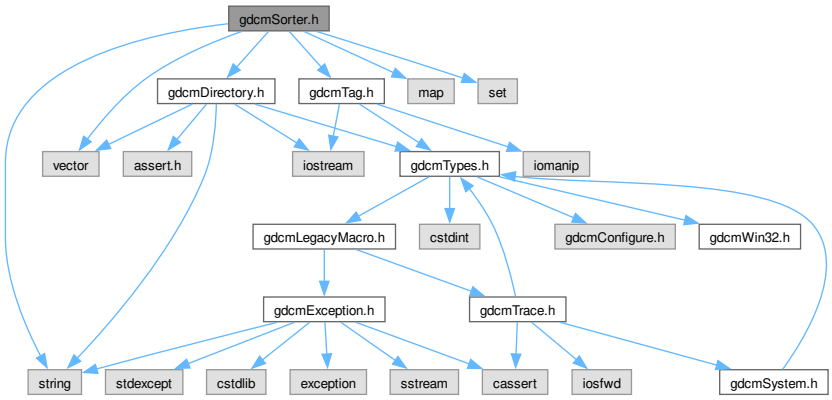
```
#include "gdcmSubject.h"
#include "gdcmCommand.h"
#include "gdcmSmartPointer.h"
#include "gdcmAnonymizeEvent.h"
#include "gdcmDataEvent.h"

namespace gdcm
{
//-----------------------------------------------------------------------------
class Event;
class GDCM_EXPORT SimpleSubjectWatcher
{
public:
  SimpleSubjectWatcher(Subject * s, const char *comment = "");
  virtual ~SimpleSubjectWatcher();
  SimpleSubjectWatcher(const SimpleSubjectWatcher&) = delete;
  void operator=(const SimpleSubjectWatcher&) = delete;

protected:
  virtual void StartFilter();
  virtual void EndFilter();
  virtual void ShowProgress(Subject *caller, const Event &evt);
  virtual void ShowFileName(Subject *caller, const Event &evt);
  virtual void ShowIteration();
  virtual void ShowAnonymization(Subject *caller, const Event &evt);
  virtual void ShowDataSet(Subject *caller, const Event &evt);
  virtual void ShowData(Subject *caller, const Event &evt);
  virtual void ShowAbort();

protected:
  // Custom API used for internal Testing do not use !
  void TestAbortOn();
  void TestAbortOff();

private:
  SmartPointer<Subject> m_Subject;
  std::string m_Comment;

  typedef SimpleMemberCommand<SimpleSubjectWatcher> SimpleCommandType;
  typedef MemberCommand<SimpleSubjectWatcher> CommandType;

  SmartPointer<SimpleCommandType> m_StartFilterCommand;
  SmartPointer<SimpleCommandType> m_EndFilterCommand;
  SmartPointer<CommandType> m_ProgressFilterCommand;
  SmartPointer<CommandType> m_FileNameFilterCommand;
  SmartPointer<SimpleCommandType> m_IterationFilterCommand;
  SmartPointer<SimpleCommandType> m_AbortFilterCommand;
  SmartPointer<CommandType> m_AnonymizeFilterCommand;
  SmartPointer<CommandType> m_DataFilterCommand;
  SmartPointer<CommandType> m_DataSetFilterCommand;

  unsigned long m_StartTag;
  unsigned long m_EndTag;
  unsigned long m_ProgressTag;
  unsigned long m_FileNameTag;
  unsigned long m_IterationTag;
  unsigned long m_AbortTag;
  unsigned long m_AnonymizeTag;
  unsigned long m_DataTag;
  unsigned long m_DataSetTag;

  bool m_TestAbort;

};
} // end namespace gdcm
//-----------------------------------------------------------------------------
#endif //GDCMSIMPLESUBJECTWATCHER_H
```

13.415 gdcmSorter.h File Reference

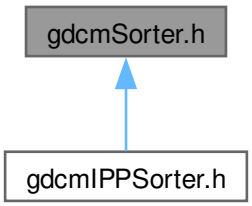
```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
```

```
#include <map>
#include <set>
```

Include dependency graph for gdcmSorter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Sorter

  Sorter.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)

13.416 gdcmSorter.h

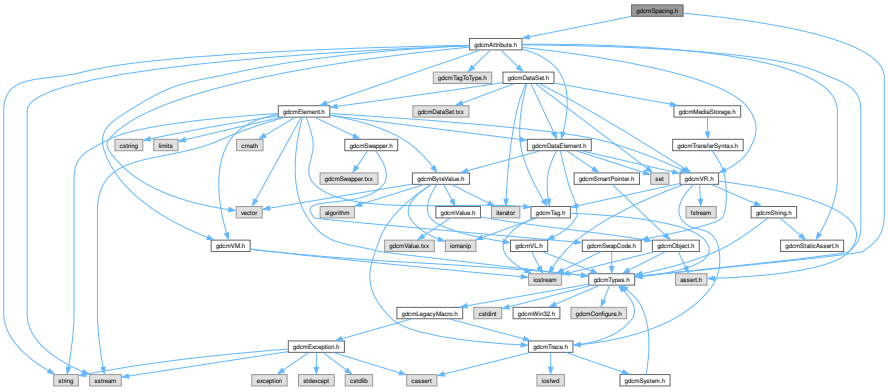
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSORTER_H
#define GDCMSORTER_H

#include "gdcmDirectory.h"
#include "gdcmTag.h"

#include <vector>
#include <string>
#include <map>
#include <set>

namespace gdcm
{
class DataSet;

class GDCM_EXPORT Sorter
{
  friend std::ostream& operator«(std::ostream &_os, const Sorter &s);
public:
  Sorter();
  virtual ~Sorter();

  virtual bool Sort(std::vector<std::string> const & filenames);

  const std::vector<std::string> &GetFilenames() const { return Filenames; }

  void Print(std::ostream &os) const;

  bool AddSelect( Tag const &tag, const char *value );

  void SetTagsToRead( std::set<Tag> const & tags );

  typedef bool (*SortFunction)(DataSet const &, DataSet const &);
  void SetSortFunction( SortFunction f );

  virtual bool StableSort(std::vector<std::string> const & filenames);

protected:
  std::vector<std::string> Filenames;
  typedef std::map<Tag,std::string> SelectionMap;
  std::map<Tag,std::string> Selection;
  SortFunction SortFunc;
  std::set<Tag> TagsToRead;
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const Sorter &s)
{
  s.Print( os );
  return os;
}


} // end namespace gdcm

#endif //GDCMSORTER_H
```

13.417 gdcmSpacing.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmAttribute.h"
```

Include dependency graph for gdcmSpacing.h:



Classes

- class gdcm::Spacing

  Class for Spacing.

Namespaces

- namespace gdcm

13.418 gdcmSpacing.h

Go to the documentation of this file.

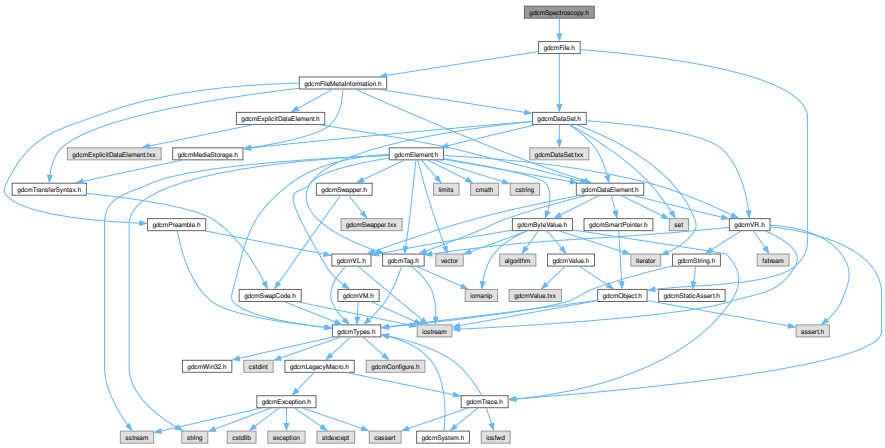
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMSPACING_H
00015 #define GDCMSPACING_H
00016
00017 #include "gdcmTypes.h"
00018 #include "gdcmAttribute.h"
00019
00020 namespace gdcm
00021 {
```

```
00056
00085 class GDCM_EXPORT Spacing
00086 {
00087 public :
00088   Spacing();
00089   ~Spacing() = default;
00090
00091   // Here are the list of spacing we support:
00092   // (0018,0088) DS [1.500000]                                    # 8,1 Spacing Between Slices
00093   // (0018,1164) DS [0.5\0.5 ]                                    # 8,2 Imager Pixel Spacing
00094   // (0018,2010) DS [0.664062\0.664062 ]                          # 18,2 Nominal Scanned Pixel
      Spacing
00095   // (0018,7022) DS [0.125\0.125 ]                                # 12,2 Detector Element Spacing
00096   // (0028,0030) DS [0.25\0.25 ]                                  # 10,2 Pixel Spacing
00097   // > (0028,0a02) CS [FIDUCIAL]                                    # 8,1 Pixel Spacing Calibration
      Type
00098   // > (0028,0a04) LO [Used fiducial ]                              # 14,1 Pixel Spacing Calibration
      Description
00099   // (0028,0034) IS [4\3 ]                                        # 4,2 Pixel Aspect Ratio
00100   // (3002,0011) DS [0.8\0.8 ]                                    # 8,2 Image Plane Pixel Spacing
00101
00102   // Here is the list of Spacing we do not support:
00103   // <entry group="0018" element="7041" vr="LT" vm="1" name="Grid Spacing Material"/>
00104   // <entry group="0018" element="9030" vr="FD" vm="1" name="Tag Spacing First Dimension"/>
00105   // <entry group="0018" element="9218" vr="FD" vm="1" name="Tag Spacing Second Dimension"/>
00106   // <entry group="0018" element="9322" vr="FD" vm="2" name="Reconstruction Pixel Spacing"/>
00107   // <entry group="0018" element="9404" vr="FL" vm="2" name="Object Pixel Spacing in Center of Beam"/>
00108   // <entry group="0040" element="08d8" vr="SQ" vm="1" name="Pixel Spacing Sequence"/>
00109   // <entry group="0070" element="0101" vr="DS" vm="2" name="Presentation Pixel Spacing"/>
00110   // <entry group="2010" element="0376" vr="DS" vm="2" name="Printer Pixel Spacing"/>
00111   // <entry group="300a" element="00e9" vr="DS" vm="2" name="Compensator Pixel Spacing"/>
00112
00113   typedef enum {
00114     DETECTOR = 0, // (0018,1164) Imager Pixel Spacing
00115     MAGNIFIED,    // (0018,1114) (IHE Mammo)
00116     CALIBRATED,   // (0028,0030) Pixel Spacing -> (0028,0a04) Pixel Spacing Calibration Description
00117     UNKNOWN
00118   } SpacingType;
00119
00120   static Attribute<0x28,0x34> ComputePixelAspectRatioFromPixelSpacing(const Attribute<0x28,0x30>&
      pixelspacing);
00121 };
00122 } // end namespace gdcm
00123 //----------------------------------------------------------------------------
00124 #endif //GDCMSPACING_H
```

13.419 gdcmSpectroscopy.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmSpectroscopy.h:



Classes

- class gdcm::Spectroscopy

  Spectroscopy class.

Namespaces

- namespace gdcm

13.420 gdcmSpectroscopy.h

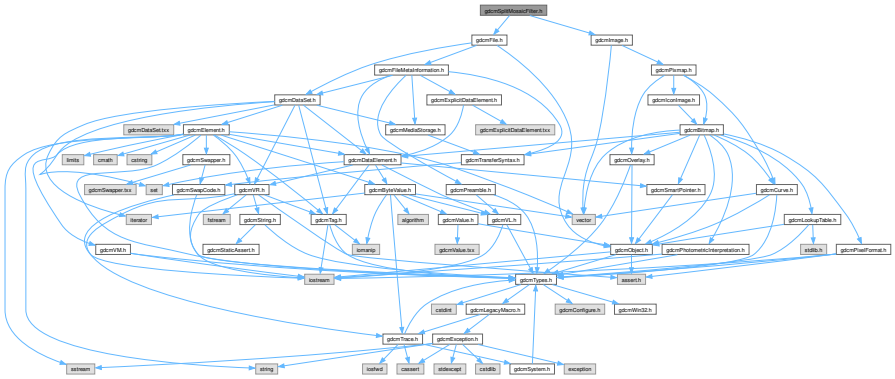
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSPECTROSCOPY_H
#define GDCMSPECTROSCOPY_H

#include "gdcmFile.h"

namespace gdcm
{
class GDCM_EXPORT Spectroscopy
{
public:
  Spectroscopy() = default;

private:
};

} // end namespace gdcm

#endif //GDCMSPECTROSCOPY_H
```

13.421 gdcmSplitMosaicFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmSplitMosaicFilter.h:



Classes

- class gdcm::SplitMosaicFilter

  SplitMosaicFilter class.

Namespaces

- namespace gdcm

13.422 gdcmSplitMosaicFilter.h

Go to the documentation of this file.

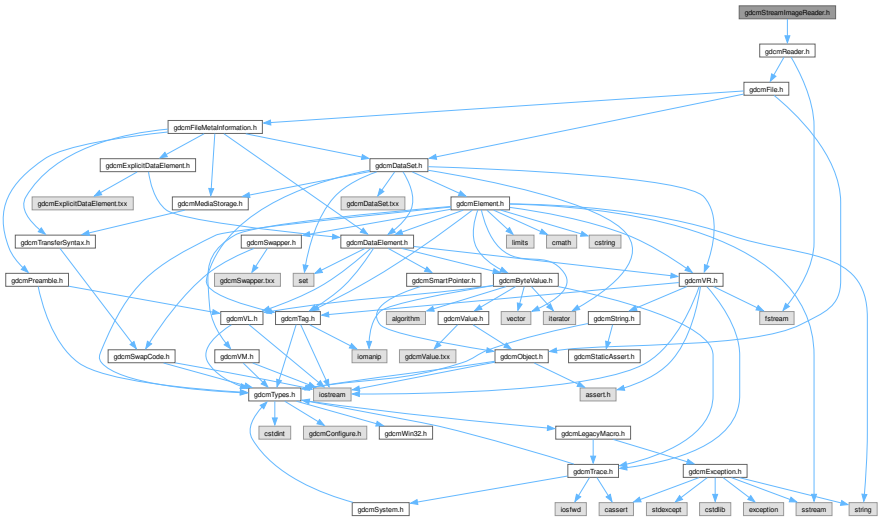
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMSPLITMOSAICFILTER_H
00015 #define GDCMSPLITMOSAICFILTER_H
00016
00017 #include "gdcmFile.h"
00018 #include "gdcmImage.h"
00019
00020 namespace gdcm
00021 {
00022
00023 /*
00024  * Everything done in this code is for the sole purpose of writing interoperable
00025  * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
00026  * If you believe anything in this code violates any law or any of your rights,
00027  * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
00028  * find a solution.
00029  */
```

```
class GDCM_EXPORT SplitMosaicFilter
{
public:
  SplitMosaicFilter();
  ~SplitMosaicFilter();

  bool Split();

  bool ComputeMOSAICDimensions(unsigned int dims[3]);

  bool ComputeMOSAICSliceNormal( double dims[3], bool & inverted );

  bool ComputeMOSAICSlicePosition( double pos[3], bool inverted );

  bool ComputeMOSAICImagePositionPatient( double pos[3],
    const double ipp[6],
    const double dircos[6],
    const double pixelspacing[3],
    const unsigned int image_dims[3] ,
    const unsigned int mosaic_dims[3], bool inverted );

  void SetImage(const Image& image);
  const Image &GetImage() const { return *I; }
  Image &GetImage() { return *I; }

  void SetFile(const File& f) { F = f; }
  File &GetFile() { return *F; }
  const File &GetFile() const { return *F; }

  static bool GetAcquisitionSize(unsigned int size[2], DataSet const & ds);

  static unsigned int GetNumberOfImagesInMosaic( File const & file );

protected:

private:
  SmartPointer<File> F;
  SmartPointer<Image> I;
};

} // end namespace gdcm

#endif //GDCMSPLITMOSAICFILTER_H
```

13.423 gdcmStreamImageReader.h File Reference

`#include "gdcmReader.h"`

Include dependency graph for gdcmStreamImageReader.h:



Classes

- class gdcm::StreamImageReader

  StreamImageReader.

Namespaces

- namespace gdcm

13.424 gdcmStreamImageReader.h

Go to the documentation of this file.

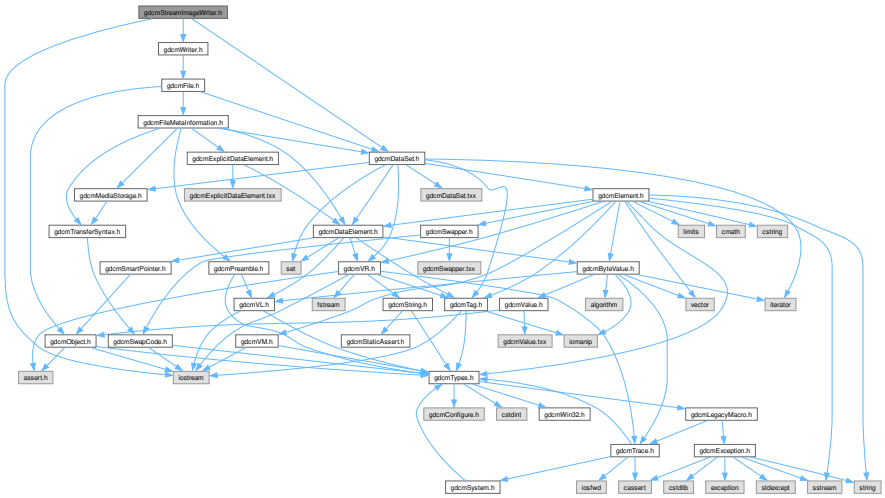
```
00001 /*=========================================================================
00002  *
00003  *  Copyright NumFOCUS
00004  *
00005  *  Licensed under the Apache License, Version 2.0 (the "License");
00006  *  you may not use this file except in compliance with the License.
00007  *  You may obtain a copy of the License at
00008  *
00009  *         http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *  Unless required by applicable law or agreed to in writing, software
00012  *  distributed under the License is distributed on an "AS IS" BASIS,
00013  *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *  See the License for the specific language governing permissions and
00015  *  limitations under the License.
00016  *
```

```
 *=============================================================================*/
#ifndef GDCMSTREAMIMAGEREADER_H
#define GDCMSTREAMIMAGEREADER_H

#include "gdcmReader.h"

namespace gdcm
{

class MediaStorage;
class GDCM_EXPORT StreamImageReader
{

public:
  StreamImageReader();
  virtual ~StreamImageReader();

  void SetFileName(const char* inFileName);
  void SetStream(std::istream& inStream);

  std::vector<unsigned int> GetDimensionsValueForResolution( unsigned int  );

  void DefinePixelExtent(uint16_t inXMin, uint16_t inXMax,
    uint16_t inYMin, uint16_t inYMax, uint16_t inZMin = 0, uint16_t inZMax = 1);

  uint32_t DefineProperBufferLength() const;

  bool Read(char* inReadBuffer, const std::size_t& inBufferLength);

  bool CanReadImage() const;

  virtual bool ReadImageInformation();

  File const & GetFile() const;

protected:
private:
  //contains a reader for being able to ReadUpToTag
  //however, we don't want the user to be able to call Read
  //either directly or via a parent class call, so we hide the reader in here.
  Reader mReader;

  std::streamoff mFileOffset; //the file offset for getting header information
#if 0
  std::streamoff mFileOffset1;
#endif
  DataSet mHeaderInformation; //all the non-pixel information

  //for thread safety, these should not be stored here, but should be used
  //for every read subregion operation.
  uint16_t mXMin, mYMin, mXMax, mYMax, mZMin, mZMax;

  bool ReadImageSubregionRAW(char* inReadBuffer, const std::size_t& inBufferLength);

  bool ReadImageSubregionJpegLS(char* inReadBuffer, const std::size_t& inBufferLength);
};

} // end namespace gdcm

#endif //GDCMSTREAMIMAGEREADER_H

```

13.425 gdcmStreamImageWriter.h File Reference

```
#include "gdcmWriter.h"
#include <iostream>
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmStreamImageWriter.h:



Classes

- class gdcm::StreamImageWriter

  StreamImageReader.

Namespaces

- namespace gdcm

13.426 gdcmStreamImageWriter.h

Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/

#ifndef GDCMSTREAMIMAGEWRITER_H
#define GDCMSTREAMIMAGEWRITER_H

#include "gdcmWriter.h"
#include <iostream>
```

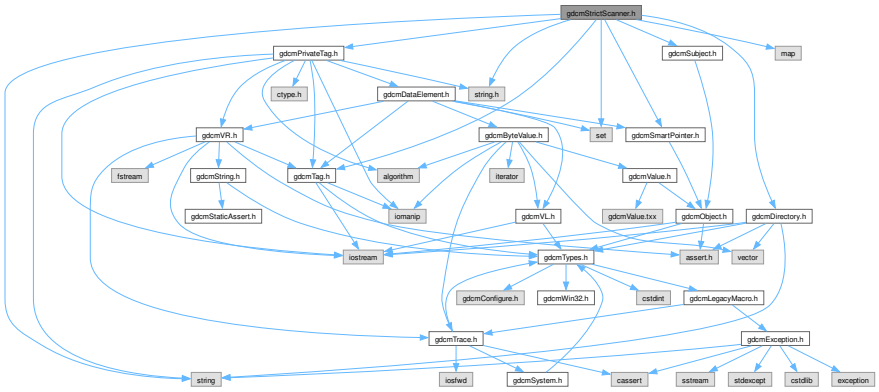
```
#include "gdcmDataSet.h"

namespace gdcm
{

class MediaStorage;
class RAWCodec;
class GDCM_EXPORT StreamImageWriter
{

public:
  StreamImageWriter();
  virtual ~StreamImageWriter();


  void SetFileName(const char* inFileName);
  void SetStream(std::ostream& inStream);

  void DefinePixelExtent(uint16_t inXMin, uint16_t inXMax,
    uint16_t inYMin, uint16_t inYMax, uint16_t inZMin = 0, uint16_t inZMax = 1);


  uint32_t DefineProperBufferLength();

  bool Write(void* inWriteBuffer, const std::size_t& inBufferLength);

  virtual bool WriteImageInformation();

  bool CanWriteFile() const;


  void SetFile(const File& inFile);

protected:

  //contains the PrepareWrite function, which will get the given dataset ready
  //for writing to disk by manufacturing the header information.
  //note that if there is a pixel element in the given dataset, that will be removed
  //during the copy, so that the imagewriter can write everything else out
  Writer mWriter;

  //is the offset necessary if we always append?
  //std::streamoff mFileOffset; //the fileoffset for getting header information
  SmartPointer<File> mspFile; //all the non-pixel information

  //for thread safety, these should not be stored here, but should be used
  //for every read subregion operation.
  uint16_t mXMin, mYMin, mXMax, mYMax, mZMin, mZMax;

  //virtual bool ReadImageSubregionRAW(std::ostream& os);
  virtual bool WriteImageSubregionRAW(char* inWriteBuffer, const std::size_t& inBufferLength);

  int WriteRawHeader(RAWCodec* inCodec, std::ostream* inStream);

  int mElementOffsets;
  int mElementOffsets1;

};


} // end namespace gdcm

#endif //GDCMSTREAMIMAGEWRITER_H
```

13.427 gdcmStrictScanner.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
```

```
#include <set>
#include <string>
#include <string.h>
```

Include dependency graph for gdcmStrictScanner.h:



Classes

- struct gdcm::StrictScanner::ltstr
- class gdcm::StrictScanner

  StrictScanner.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const StrictScanner &s)

13.428 gdcmStrictScanner.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMSTRICTSCANNER_H
```

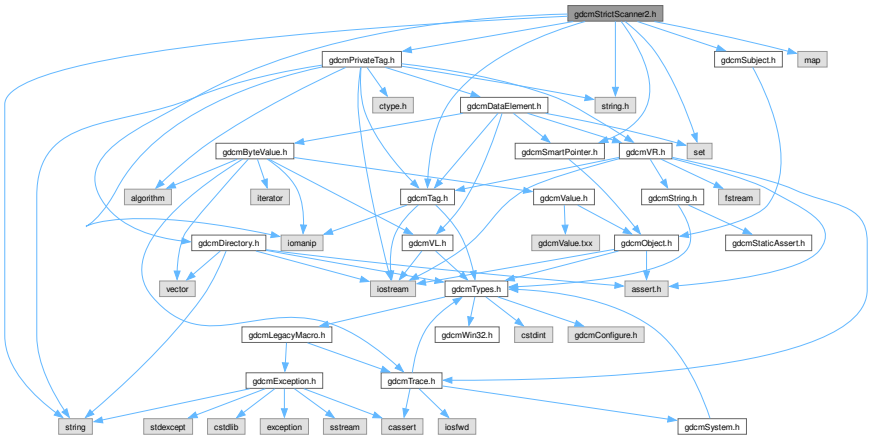
```
#define GDCMSTRICTSCANNER_H

#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"

#include <map>
#include <set>
#include <string>

#include <string.h> // strcmp

namespace gdcm
{
class StringFilter;

class GDCM_EXPORT StrictScanner : public Subject
{
  friend std::ostream& operator«(std::ostream &_os, const StrictScanner &s);
public:
  StrictScanner():Values(),Filenames(),Mappings() {}
  ~StrictScanner() override;

  typedef std::map<Tag, const char*> TagToValue;
  //typedef std::map<Tag, ConstCharWrapper> TagToValue; //StringMap;
  //typedef TagToStringMap TagToValue;
  typedef TagToValue::value_type TagToValueValueType;

  void AddTag( Tag const & t );
  void ClearTags();

  // Work in progress do not use:
  void AddPrivateTag( PrivateTag const & t );

  void AddSkipTag( Tag const & t );
  void ClearSkipTags();

  bool Scan( Directory::FilenamesType const & filenames );

  Directory::FilenamesType const &GetFilenames() const { return Filenames; }

  void Print( std::ostream & os ) const override;

  void PrintTable( std::ostream & os ) const;

  bool IsKey( const char * filename ) const;

  Directory::FilenamesType GetKeys() const;

  // struct to store all the values found:
  typedef std::set< std::string > ValuesType;

  ValuesType const & GetValues() const { return Values; }

  ValuesType GetValues(Tag const &t) const;

  Directory::FilenamesType GetOrderedValues(Tag const &t) const;

  /* ltstr is CRITICAL, otherwise pointers value are used to do the key comparison */
  struct ltstr
    {
    bool operator()(const char* s1, const char* s2) const
      {
      assert( s1 && s2 );
      return strcmp(s1, s2) < 0;
      }
    };
  typedef std::map<const char *,TagToValue, ltstr> MappingType;
  typedef MappingType::const_iterator ConstIterator;
  ConstIterator Begin() const { return Mappings.begin(); }
  ConstIterator End() const { return Mappings.end(); }

  MappingType const & GetMappings() const { return Mappings; }

  TagToValue const & GetMapping(const char *filename) const;

  const char *GetFilenameFromTagToValue(Tag const &t, const char *valueref) const;

  Directory::FilenamesType GetAllFilenamesFromTagToValue(Tag const &t, const char *valueref) const;
```

```
00144
00146   // by a call to GetMapping()
00147   TagToValue const & GetMappingFromTagToValue(Tag const &t, const char *value) const;
00148
00154   const char* GetValue(const char *filename, Tag const &t) const;
00155
00157   static SmartPointer<StrictScanner> New() { return new StrictScanner; }
00158
00159 protected:
00160   void ProcessPublicTag(StringFilter &sf, const char *filename);
00161 private:
00162   // struct to store all uniq tags in ascending order:
00163   typedef std::set< Tag > TagsType;
00164   typedef std::set< PrivateTag > PrivateTagsType;
00165   std::set< Tag > Tags;
00166   std::set< PrivateTag > PrivateTags;
00167   std::set< Tag > SkipTags;
00168   ValuesType Values;
00169   Directory::FilenamesType Filenames;
00170
00171   // Main struct that will hold all mapping:
00172   MappingType Mappings;
00173
00174   double Progress;
00175 };
00176 //-----------------------------------------------------------------------------
00177 inline std::ostream& operator«(std::ostream &os, const StrictScanner &s)
00178 {
00179   s.Print( os );
00180   return os;
00181 }
00182
00183 } // end namespace gdcm
00184
00185 #endif //GDCMSTRICTSCANNER_H
```

13.429 gdcmStrictScanner2.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include <map>
#include <set>
#include <string>
#include <string.h>
```

Include dependency graph for gdcmStrictScanner2.h:



Classes

- struct gdcm::StrictScanner2::ltstr
- class gdcm::StrictScanner2

  StrictScanner2.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const StrictScanner2 &s)

13.430 gdcmStrictScanner2.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSTRICTSCANNER2_H
#define GDCMSTRICTSCANNER2_H

```

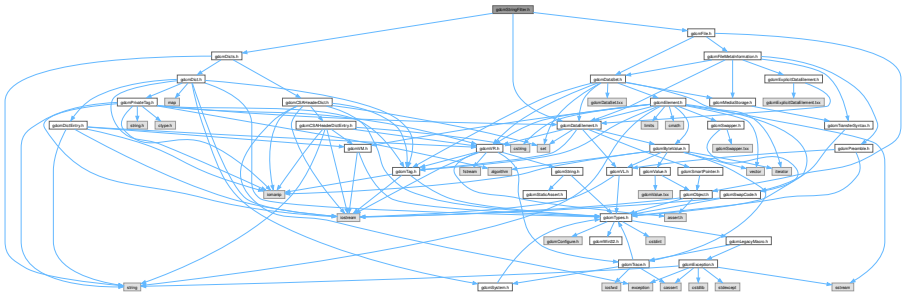
```
#include "gdcmDirectory.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"

#include <map>
#include <set>
#include <string>

#include <string.h>  // strcmp

namespace gdcm {
class StringFilter;

class GDCM_EXPORT StrictScanner2 : public Subject {
  friend std::ostream &operator«(std::ostream &_os, const StrictScanner2 &s);

 public:
  StrictScanner2() : Values(), Filenames(), PublicMappings(), PrivateMappings(), Progress(0.0) {}
  ~StrictScanner2() override;

  typedef std::map<Tag, const char *> PublicTagToValue;
  typedef PublicTagToValue::value_type PublicTagToValueValueType;

  typedef std::map<PrivateTag, const char *> PrivateTagToValue;
  typedef PrivateTagToValue::value_type PrivateTagToValueValueType;

  bool AddPublicTag(Tag const &t);
  void ClearPublicTags();

  // Work in progress do not use:
  bool AddPrivateTag(PrivateTag const &pt);
  void ClearPrivateTags();

  bool AddSkipTag(Tag const &t);
  void ClearSkipTags();

  bool Scan(Directory::FilenamesType const &filenames);

  Directory::FilenamesType const &GetFilenames() const { return Filenames; }

  void Print(std::ostream &os) const override;

  void PrintTable(std::ostream &os, bool header = false) const;

  bool IsKey(const char *filename) const;

  Directory::FilenamesType GetKeys() const;

  // struct to store all the values found:
  typedef std::set<std::string> ValuesType;

  ValuesType const &GetValues() const { return Values; }

  ValuesType GetPublicValues(Tag const &t) const;

  ValuesType GetPrivateValues(PrivateTag const &pt) const;

  Directory::FilenamesType GetPublicOrderedValues(Tag const &t) const;

  Directory::FilenamesType GetPrivateOrderedValues(PrivateTag const &pt) const;

  /* ltstr is CRITICAL, otherwise pointers value are used to do the key
   * comparison */
  struct ltstr {
    bool operator()(const char *s1, const char *s2) const {
      assert(s1 && s2);
      return strcmp(s1, s2) < 0;
    }
  };
  typedef std::map<const char *, PublicTagToValue, ltstr> PublicMappingType;
  typedef PublicMappingType::const_iterator PublicConstIterator;
  PublicConstIterator Begin() const { return PublicMappings.begin(); }
  PublicConstIterator End() const { return PublicMappings.end(); }

  typedef std::map<const char *, PrivateTagToValue, ltstr> PrivateMappingType;
  typedef PrivateMappingType::const_iterator PrivateConstIterator;
  PrivateConstIterator PrivateBegin() const { return PrivateMappings.begin(); }
  PrivateConstIterator PrivateEnd() const { return PrivateMappings.end(); }

```

```
  PublicMappingType const &GetPublicMappings() const { return PublicMappings; }
  PrivateMappingType const &GetPrivateMappings() const {
    return PrivateMappings;
  }

  PublicTagToValue const &GetPublicMapping(const char *filename) const;
  PrivateTagToValue const &GetPrivateMapping(const char *filename) const;

  const char *GetFilenameFromPublicTagToValue(Tag const &t,
                                              const char *valueref) const;
  const char *GetFilenameFromPrivateTagToValue(PrivateTag const &pt,
                                               const char *valueref) const;

  Directory::FilenamesType GetAllFilenamesFromPublicTagToValue(
      Tag const &t, const char *valueref) const;
  Directory::FilenamesType GetAllFilenamesFromPrivateTagToValue(
      PrivateTag const &pt, const char *valueref) const;

  // by a call to GetMapping()
  PublicTagToValue const &GetMappingFromPublicTagToValue(
      Tag const &t, const char *value) const;
  PrivateTagToValue const &GetMappingFromPrivateTagToValue(
      PrivateTag const &pt, const char *value) const;

  const char *GetPublicValue(const char *filename, Tag const &t) const;
  const char *GetPrivateValue(const char *filename, PrivateTag const &t) const;

  static SmartPointer<StrictScanner2> New() { return new StrictScanner2; }

 protected:
  void ProcessPublicTag(StringFilter &sf, const char *filename);
  void ProcessPrivateTag(StringFilter &sf, const char *filename);

 private:
  // struct to store all uniq tags in ascending order:
  typedef std::set<Tag> PublicTagsType;
  typedef std::set<PrivateTag> PrivateTagsType;
  std::set<Tag> PublicTags;          // Public and Private Creator
  std::set<PrivateTag> PrivateTags;  // Only Private (no Private Creator)
  std::set<Tag> SkipTags;
  ValuesType Values;
  Directory::FilenamesType Filenames;

  // Main struct that will hold all public mapping:
  PublicMappingType PublicMappings;
  // Main struct that will hold all private mapping:
  PrivateMappingType PrivateMappings;

  double Progress;
};
//-----------------------------------------------------------------------------
inline std::ostream &operator«(std::ostream &os, const StrictScanner2 &s) {
  s.Print(os);
  return os;
}

}  // end namespace gdcm

#endif  // GDCMSTRICTSCANNER2_H
```

13.431 gdcmStringFilter.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"
```

Include dependency graph for gdcmStringFilter.h:



Classes

- class gdcm::StringFilter

  StringFilter.

Namespaces

- namespace gdcm

13.432 gdcmStringFilter.h

Go to the documentation of this file.

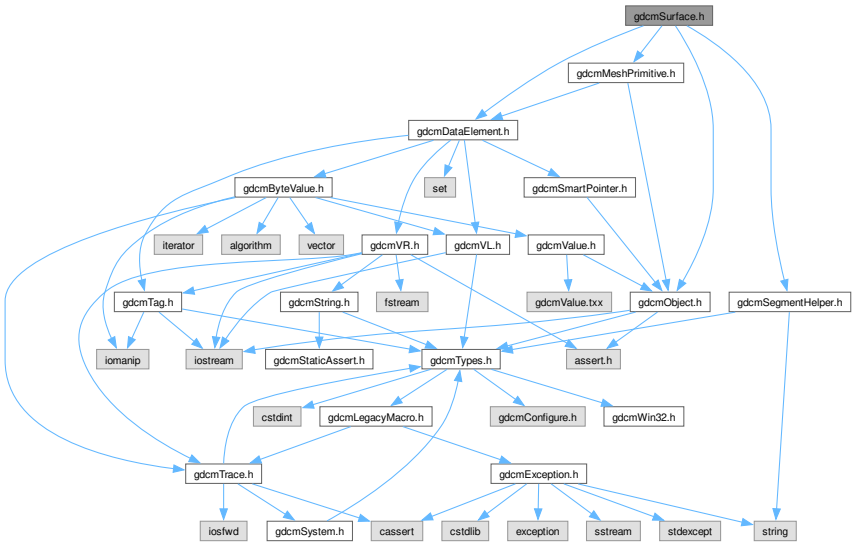
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMSTRINGFILTER_H
00015 #define GDCMSTRINGFILTER_H
00016
00017 #include "gdcmDataElement.h"
00018 #include "gdcmDicts.h"
00019 #include "gdcmFile.h"
00020
00021 namespace gdcm
00022 {
00023
00029 class GDCM_EXPORT StringFilter
00030 {
00031 public:
00032   StringFilter();
00033   ~StringFilter();
00034
00036   void UseDictAlways(bool) {}
00037
00039   void SetDicts(const Dicts &dicts);
00040
00044   std::string ToString(const DataElement& de) const;
```

```
  std::string ToString(const Tag& t) const;

  std::string ToString(const PrivateTag& t) const;

  std::pair<std::string, std::string> ToStringPair(const DataElement& de) const;
  std::pair<std::string, std::string> ToStringPair(const Tag& t) const;

  std::string FromString(const Tag&t, const char * value, size_t len);

  void SetFile(const File& f) { F = f; }
  File &GetFile() { return *F; }
  const File &GetFile() const { return *F; }

  bool ExecuteQuery(std::string const &query, std::string & value) const;

protected:
  std::pair<std::string, std::string> ToStringPair(const Tag& t, DataSet const &ds) const;
  bool ExecuteQuery(std::string const &query, DataSet const &ds, std::string & value) const;

private:
  std::pair<std::string, std::string> ToStringPairInternal(const DataElement& de, DataSet const &ds) const;
  SmartPointer<File> F;
};

} // end namespace gdcm

#endif //GDCMSTRINGFILTER_H
```

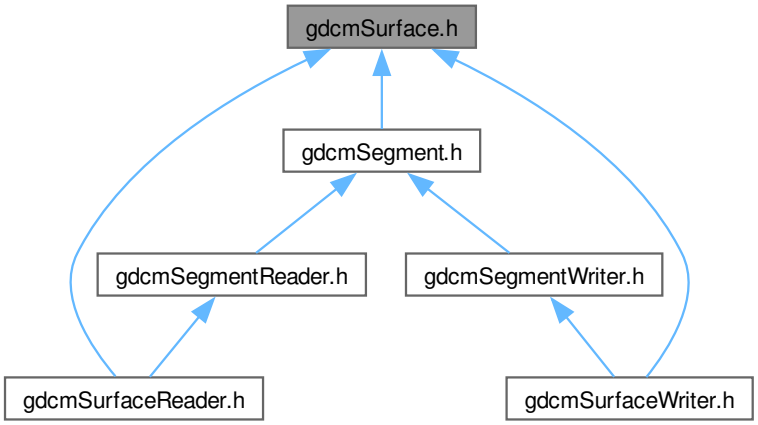
13.433 gdcmSurface.h File Reference

```
#include <gdcmObject.h>
#include <gdcmDataElement.h>
#include <gdcmMeshPrimitive.h>
#include "gdcmSegmentHelper.h"
```

Include dependency graph for gdcmSurface.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Surface

  This class defines a SURFACE IE.

Namespaces

- namespace gdcm

13.434 gdcmSurface.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSURFACE_H
#define GDCMSURFACE_H

#include <gdcmObject.h>
#include <gdcmDataElement.h>
#include <gdcmMeshPrimitive.h>
#include "gdcmSegmentHelper.h"  // for BasicCodedEntry

```

```
namespace gdcm
{

class GDCM_EXPORT Surface : public Object
{
public:

  typedef enum {
    NO = 0,
    YES,
    UNKNOWN,
    STATES_END
  } STATES;

  static const char * GetSTATESString(STATES state);
  static STATES GetSTATES(const char * state);

  typedef enum {
    SURFACE = 0,
    WIREFRAME,
    POINTS,
    VIEWType_END
  } VIEWType;

  static const char * GetVIEWTypeString(VIEWType type);
  static VIEWType GetVIEWType(const char * type);

  Surface();

  ~Surface() override;

  //**    Common getters/setters      **//
  unsigned long GetSurfaceNumber() const;
  void SetSurfaceNumber(const unsigned long nb);

  const char * GetSurfaceComments() const;
  void SetSurfaceComments(const char * comment);

  bool GetSurfaceProcessing() const;
  void SetSurfaceProcessing(bool b);

  float GetSurfaceProcessingRatio() const;
  void SetSurfaceProcessingRatio(const float ratio);

  const char * GetSurfaceProcessingDescription() const;
  void SetSurfaceProcessingDescription(const char * description);

  SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm() const;
  SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm();
  void SetProcessingAlgorithm(SegmentHelper::BasicCodedEntry const & BSE);

  unsigned short GetRecommendedDisplayGrayscaleValue() const;
  void SetRecommendedDisplayGrayscaleValue(const unsigned short vl);

  const unsigned short * GetRecommendedDisplayCIELabValue() const;
  unsigned short GetRecommendedDisplayCIELabValue(const unsigned int idx) const;
  void SetRecommendedDisplayCIELabValue(const unsigned short vl[3]);
  void SetRecommendedDisplayCIELabValue(const unsigned short vl, const unsigned int idx = 0);
  void SetRecommendedDisplayCIELabValue(const std::vector< unsigned short > & vl);

  float GetRecommendedPresentationOpacity() const;
  void SetRecommendedPresentationOpacity(const float opacity);

  VIEWType GetRecommendedPresentationType() const;
  void SetRecommendedPresentationType(VIEWType type);

  STATES GetFiniteVolume() const;
  void SetFiniteVolume(STATES state);

  STATES GetManifold() const;
  void SetManifold(STATES state);

  SegmentHelper::BasicCodedEntry const & GetAlgorithmFamily() const;
  SegmentHelper::BasicCodedEntry & GetAlgorithmFamily();
  void SetAlgorithmFamily(SegmentHelper::BasicCodedEntry const & BSE);

  const char * GetAlgorithmVersion() const;
  void SetAlgorithmVersion(const char * str);

  const char * GetAlgorithmName() const;
  void SetAlgorithmName(const char * str);
```

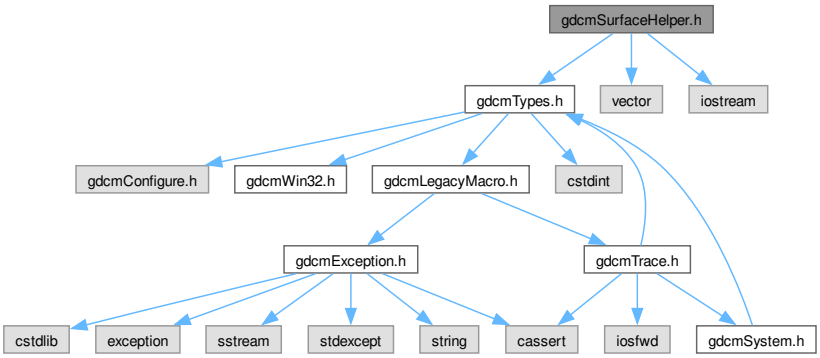
```
00114
00115   //**    Points getters/setters      **//
00116   unsigned long GetNumberOfSurfacePoints() const;
00117   void SetNumberOfSurfacePoints(const unsigned long nb);
00118
00119   const DataElement & GetPointCoordinatesData() const;
00120   DataElement & GetPointCoordinatesData();
00121
00122   void SetPointCoordinatesData(DataElement const & de);
00123
00127   const float * GetPointPositionAccuracy() const;
00128   void SetPointPositionAccuracy(const float * accuracies);
00129
00130   float GetMeanPointDistance() const;
00131   void SetMeanPointDistance(float average);
00132
00133   float GetMaximumPointDistance() const;
00134   void SetMaximumPointDistance(float maximum);
00135
00139   const float * GetPointsBoundingBoxCoordinates() const;
00140   void SetPointsBoundingBoxCoordinates(const float * coordinates);
00141
00145   const float * GetAxisOfRotation() const;
00146   void SetAxisOfRotation(const float * axis);
00147
00151   const float * GetCenterOfRotation() const;
00152   void SetCenterOfRotation(const float * center);
00153
00154   //**    Vectors getters/setters      **//
00155   unsigned long GetNumberOfVectors() const;
00156   void SetNumberOfVectors(const unsigned long nb);
00157
00158   unsigned short GetVectorDimensionality() const;
00159   void SetVectorDimensionality(const unsigned short dim);
00160
00161   const float * GetVectorAccuracy() const;
00162   void SetVectorAccuracy(const float * accuracy);
00163
00164   const DataElement & GetVectorCoordinateData() const;
00165   DataElement & GetVectorCoordinateData();
00166
00167   void SetVectorCoordinateData(DataElement const & de);
00168
00169   //**    Primitive getters/setters      **//
00170   MeshPrimitive const & GetMeshPrimitive() const;
00171   MeshPrimitive & GetMeshPrimitive();
00172
00173   void SetMeshPrimitive(MeshPrimitive & mp);
00174
00175 private:
00176
00177   //**        Common members      **//
00178
00179   //0066 0003 UL 1 Surface Number
00180   unsigned long SurfaceNumber;
00181   //0066 0004 LT 1 Surface Comments
00182   std::string   SurfaceComments;
00183
00184   //0066 0009 CS 1 Surface Processing
00185   bool          SurfaceProcessing;
00186   //0066 000a FL 1 Surface Processing Ratio
00187   float         SurfaceProcessingRatio;
00188   //0066 000b LO 1 Surface Processing Description
00189   std::string   SurfaceProcessingDescription;
00190   // Processing Algorithm Code
00191   SegmentHelper::BasicCodedEntry ProcessingAlgorithm;
00192
00193   //0062 000c US 1 Recommended Display Grayscale Value
00194   unsigned short  RecommendedDisplayGrayscaleValue;
00195   //0062 000d US 3 Recommended Display CIELab Value
00196   unsigned short  RecommendedDisplayCIELabValue[3];
00197
00198   // 0066 000c FL 1 Recommended Presentation Opacity
00199   float           RecommendedPresentationOpacity;
00200   // 0066 000d CS 1 Recommended Presentation Type
00201   VIEWType        RecommendedPresentationType;
00202
00203   //0066 000e CS 1 Finite Volume
00204   STATES        FiniteVolume;
00205   //0066 0010 CS 1 Manifold
00206   STATES        Manifold;
```

```
  // Algorithm Family Code
  SegmentHelper::BasicCodedEntry AlgorithmFamily;

  //0066 0031 LO 1 Algorithm Version
  std::string   AlgorithmVersion;
  //0066 0032 LT 1 Algorithm Parameters
  //0066 0036 LO 1 Algorithm Name
  std::string   AlgorithmName;


  //**        Point members      **//

  //0066 0015 UL 1 Number of Surface Points
  unsigned long NumberOfSurfacePoints;
  //0066 0016 OF 1 Point Coordinates Data
  DataElement   PointCoordinatesData;
  //0066 0017 FL 3 Point Position Accuracy
  float *       PointPositionAccuracy;
  //0066 0018 FL 1 Mean Point Distance
  float         MeanPointDistance;
  //0066 0019 FL 1 Maximum Point Distance
  float         MaximumPointDistance;
  //0066 001a FL 6 Points Bounding Box Coordinates
  float *       PointsBoundingBoxCoordinates;
  //0066 001b FL 3 Axis of Rotation
  float *       AxisOfRotation;
  //0066 001c FL 3 Center of Rotation
  float *       CenterOfRotation;


  //**        Normal members      **//

  //0066 001e UL 1 Number of Vectors
  unsigned long NumberOfVectors;
  //0066 001f US 1 Vector Dimensionality
  unsigned short VectorDimensionality;
  //0066 0020 FL 1-n Vector Accuracy
  float *       VectorAccuracy;
  //0066 0021 OF 1 Vector Coordinate Data
  DataElement   VectorCoordinateData;


  //**        Primitive members      **//
  SmartPointer< MeshPrimitive > Primitive;
};

}

#endif // GDCMSURFACE_H
```

13.435 gdcmSurfaceHelper.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <iostream>
```

Include dependency graph for gdcmSurfaceHelper.h:



Classes

- class gdcm::SurfaceHelper

  SurfaceHelper.

Namespaces

- namespace gdcm

13.436 gdcmSurfaceHelper.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2017 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSURFACEHELPER_H
#define GDCMSURFACEHELPER_H

#include "gdcmTypes.h"  // for GDCM_EXPORT

#include <vector>
#include <iostream>

namespace gdcm
{

class GDCM_EXPORT SurfaceHelper
{
public:
```

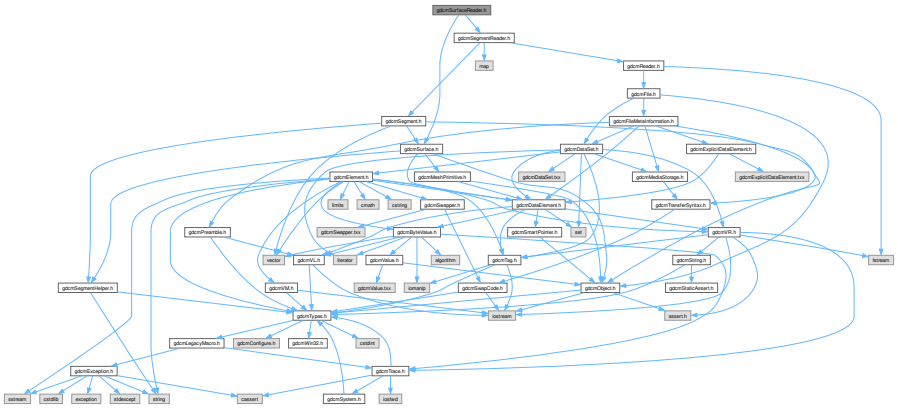
```
  typedef std::vector< unsigned short > ColorArray;

  template <typename T, typename U>
  static unsigned short RGBToRecommendedDisplayGrayscale(const std::vector<T> & RGB,
                                                         const U rangeMax = 255);
  template <typename T, typename U>
  static ColorArray RGBToRecommendedDisplayCIELab(const std::vector<T> & RGB,
                                                  const U rangeMax = 255);
  template <typename T, typename U>
  static std::vector<T> RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
                                                      const U rangeMax = 255);
  template <typename U>
  static std::vector<float> RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
                                                      const U rangeMax = 255);

private:

  static std::vector< float > RGBToXYZ(const std::vector<float> & RGB);

  static std::vector< float > XYZToRGB(const std::vector<float> & XYZ);

  static std::vector< float > XYZToCIELab(const std::vector<float> & XYZ);

  static std::vector< float > CIELabToXYZ(const std::vector<float> & CIELab);
};

template <typename T, typename U>
unsigned short SurfaceHelper::RGBToRecommendedDisplayGrayscale(const std::vector<T> & RGB,
                                                               const U rangeMax/* = 255*/)
{
  assert(RGB.size() > 2);

  unsigned short Grayscale = 0;

  const float inverseRangeMax = 1.0f / (float) rangeMax;

  // 0xFFFF "=" 255 "=" white
  Grayscale = (unsigned short) ((0.2989 * RGB[0] + 0.5870 * RGB[1] + 0.1140 * RGB[2])
                                * inverseRangeMax // Convert to range 0-1
                                * 0xFFFF);        // Convert to range 0x0000-0xFFFF

  return Grayscale;
}

template <typename T, typename U>
SurfaceHelper::ColorArray SurfaceHelper::RGBToRecommendedDisplayCIELab(const std::vector<T> & RGB,
                                                                       const U rangeMax/* = 255*/)
{
  assert(RGB.size() > 2);

  ColorArray CIELab(3);
  std::vector<float> tmp(3);

  // Convert to range 0-1
  const float inverseRangeMax = 1.0f / (float) rangeMax;
  tmp[0] = (float) (RGB[0] * inverseRangeMax);
  tmp[1] = (float) (RGB[1] * inverseRangeMax);
  tmp[2] = (float) (RGB[2] * inverseRangeMax);

  tmp = SurfaceHelper::XYZToCIELab( SurfaceHelper::RGBToXYZ( tmp ) );

  // Convert to range 0x0000-0xFFFF
  // 0xFFFF "=" 127, 0x8080 "=" 0, 0x0000 "=" -128
  CIELab[0] = (unsigned short) (          0xFFFF           * (tmp[0]*0.01f));
  if(tmp[1] >= -128 && tmp[1] <= 0)
  {
      CIELab[1] = (unsigned short)(((float)(0x8080)/128.0f)*tmp[1] + ((float)0x8080));
  }
  else if(tmp[1] <= 127 && tmp[1] > 0)
  {
      CIELab[1] = (unsigned short)(((float)(0xFFFF - 0x8080)/127.0f)*tmp[1] + (float)(0x8080));
  }
  if(tmp[2] >= -128 && tmp[2] <= 0)
  {
      CIELab[2] = (unsigned short)(((float)0x8080/128.0f)*tmp[2] + ((float)0x8080));
  }
  else if(tmp[2] <= 127 && tmp[2] > 0)
  {
      CIELab[2] = (unsigned short)(((float)(0xFFFF - 0x8080)/127.0f)*tmp[2] + (float)(0x8080));
  }
```

```

  return CIELab;
}

template <typename T, typename U>
std::vector<T> SurfaceHelper::RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
                                                            const U rangeMax/* = 255*/)
{
  assert(CIELab.size() > 2);

  std::vector<T> RGB(3);
  std::vector<float> tmp(3);

  // Convert to range 0-1

  tmp[0] = 100.0f*CIELab[0] /(float)(0xFFFF);
  if(CIELab[1] <= 0x8080)
  {
      tmp[1] = (float)(((CIELab[1] - 0x8080) * 128.0f)/(float)0x8080);
  }
  else
  {
      tmp[1] = (float)((CIELab[1]-0x8080)*127.0f / (float)(0xFFFF - 0x8080));
  }
  if(CIELab[2] <= 0x8080)
  {
      tmp[2] = (float)(((CIELab[2] - 0x8080) * 128.0f)/(float)0x8080);
  }
  else
  {
      tmp[2] = (float)((CIELab[2]-0x8080)*127.0f / (float)(0XFFFF - 0x8080));
  }

  tmp = SurfaceHelper::XYZToRGB( SurfaceHelper::CIELabToXYZ( tmp ) );

  // Convert to range 0-rangeMax
  RGB[0] = (T) (tmp[0] * rangeMax);
  RGB[1] = (T) (tmp[1] * rangeMax);
  RGB[2] = (T) (tmp[2] * rangeMax);

  return RGB;
}

template <typename U>
std::vector<float> SurfaceHelper::RecommendedDisplayCIELabToRGB(const ColorArray & CIELab,
                                                            const U rangeMax/* = 255*/)
{
  return RecommendedDisplayCIELabToRGB<float>(CIELab, rangeMax);
}

} // end namespace gdcm

#endif // GDCMSURFACEHELPER_H
```

13.437 gdcmSurfaceReader.h File Reference

```
#include <gdcmSegmentReader.h>
#include <gdcmSurface.h>
```

Include dependency graph for gdcmSurfaceReader.h:



Classes

- class gdcm::SurfaceReader

  This class defines a SURFACE IE reader.

Namespaces

- namespace gdcm

13.438 gdcmSurfaceReader.h

Go to the documentation of this file.

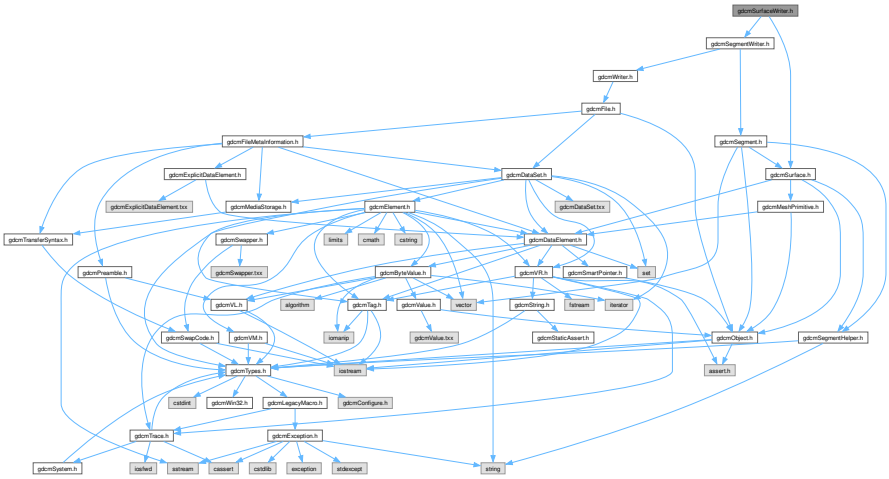
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMSURFACEREADER_H
00015 #define GDCMSURFACEREADER_H
00016
00017 #include <gdcmSegmentReader.h>
00018 #include <gdcmSurface.h>
00019
00020 namespace gdcm
00021 {
00022
00029 class GDCM_EXPORT SurfaceReader : public SegmentReader
00030 {
00031 public:
00032     SurfaceReader();
00033
00034     ~SurfaceReader() override;
```

```
00035
00037     bool Read() override;
00038
00039     unsigned long GetNumberOfSurfaces() const;
00040
00041   protected:
00042
00043     bool ReadSurfaces();
00044
00045     bool ReadSurface(const Item & surfaceItem, const unsigned long idx);
00046
00047     bool ReadPointMacro(SmartPointer< Surface > surface, const DataSet & surfaceDS);
00048 };
00049
00050 }
00051
00052 #endif // GDCMSURFACEREADER_H
```

13.439 gdcmSurfaceWriter.h File Reference

```
#include <gdcmSegmentWriter.h>
#include <gdcmSurface.h>
```

Include dependency graph for gdcmSurfaceWriter.h:



Classes

- class gdcm::SurfaceWriter

  This class defines a SURFACE IE writer.

Namespaces

- namespace gdcm

13.440 gdcmSurfaceWriter.h

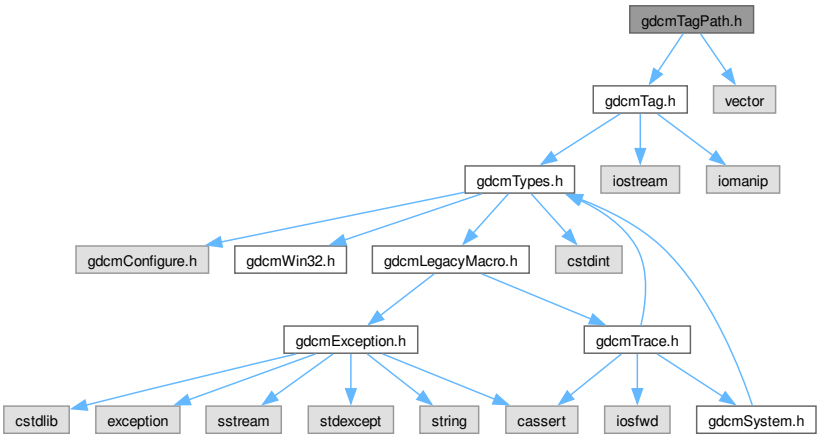
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSURFACEWRITER_H
#define GDCMSURFACEWRITER_H

#include <gdcmSegmentWriter.h>
#include <gdcmSurface.h>

namespace gdcm
{

class GDCM_EXPORT SurfaceWriter : public SegmentWriter
{
public:
    SurfaceWriter();

    ~SurfaceWriter() override;

//    const Surface & GetSurface() const { return *SurfaceData; }
//    Surface & GetSurface() { return *SurfaceData; }
//    void SetSurface(Surface const & segment);

    bool Write() override; // Execute()

    unsigned long GetNumberOfSurfaces();
    void SetNumberOfSurfaces(const unsigned long nb);

protected:

    bool PrepareWrite();

    void ComputeNumberOfSurfaces();

    bool PrepareWritePointMacro(SmartPointer< Surface > surface,
                                DataSet & surfaceDS,
                                const TransferSyntax & ts);

    //0066 0001 UL 1 Number of Surfaces
    unsigned long NumberOfSurfaces;
};

}

#endif // GDCMSURFACEWRITER_H
```

13.441 gdcmTagPath.h File Reference

```
#include "gdcmTag.h"
#include <vector>
```

Include dependency graph for gdcmTagPath.h:



Classes

- class gdcm::TagPath

  class to handle a path of tag.

Namespaces

- namespace gdcm

13.442 gdcmTagPath.h

Go to the documentation of this file.

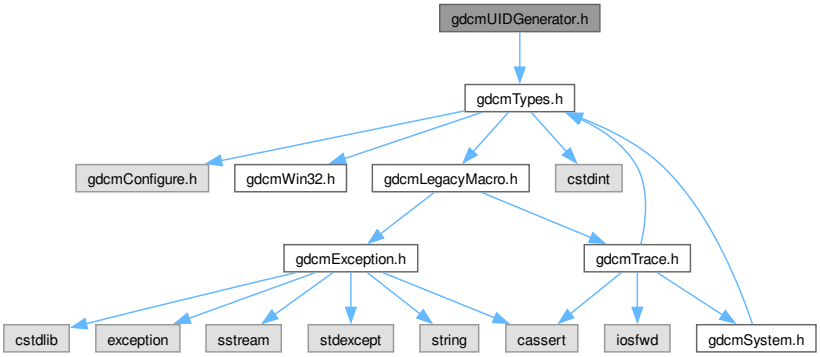
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTAGPATH_H
#define GDCMTAGPATH_H

#include "gdcmTag.h"

#include <vector>

namespace gdcm
{

```

```
00030 class GDCM_EXPORT TagPath
00031 {
00032 public:
00033   TagPath();
00034   ~TagPath();
00035   void Print(std::ostream &) const;
00036
00041   bool ConstructFromString(const char *path);
00042
00044   static bool IsValid(const char *path);
00045
00047   bool ConstructFromTagList(Tag const *l, unsigned int n);
00048
00049   bool Push(Tag const & t);
00050   bool Push(unsigned int itemnum);
00051
00052 private:
00053   std::vector<Tag> Path;
00054 };
00055
00056 } // end namespace gdcm
00057
00058 #endif //GDCMTAGPATH_H
```

13.443 gdcmUIDGenerator.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmUIDGenerator.h:



Classes

- class gdcm::UIDGenerator

  Class for generating unique UID.

Namespaces

- namespace gdcm

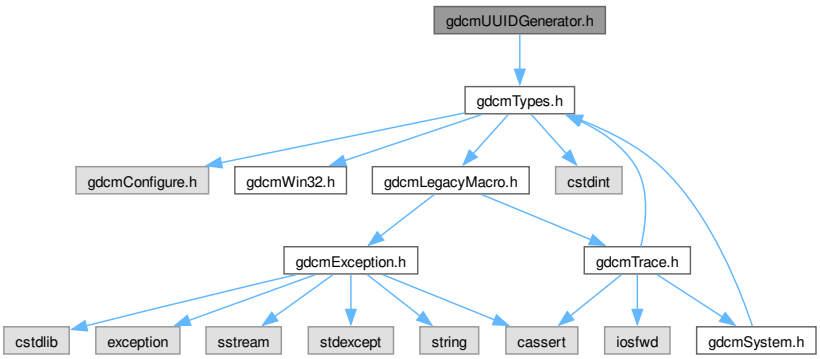
13.444 gdcmUIDGenerator.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMUIDGENERATOR_H
#define GDCMUIDGENERATOR_H

#include "gdcmTypes.h"

namespace gdcm
{

class GDCM_EXPORT UIDGenerator
{
public:
  UIDGenerator():Unique() {}

  // Function to override the GDCM root with a user one:
  // WARNING: This need to be a valid root, otherwise call will fail
  // Implementation note. According to DICOM standard PS 3.5, Section 9 :
  // Unique Identifiers (UIDs), we have:
  /*
  ...
  The <org root> portion of the UID uniquely identifies an organization, (i.e., manufacturer, research
  organization, NEMA, etc.), and is composed of a number of numeric components as defined by ISO 8824.
  The <suffix> portion of the UID is also composed of a number of numeric components, and shall be
  unique within the scope of the <org root>. This implies that the organization identified in the <org
root> is
  responsible for guaranteeing <suffix> uniqueness by providing registration policies. These policies
shall
  guarantee <suffix> uniqueness for all UID's created by that organization. Unlike the <org root>, which
may
  be common for UID's in an organization, the <suffix> shall take different unique values between
different
  UID's that identify different objects.
  ...
   */
  static void SetRoot(const char * root);
  static const char *GetRoot();

  const char* Generate();

  static bool IsValid(const char *uid);

  static const char *GetGDCMUID(); // who would want that in the public API ??

protected:
  static bool GenerateUUID(unsigned char *uuid_data);

private:
  static const char GDCM_UID[];
  static std::string Root;
  static std::string EncodedHardwareAddress;
  std::string Unique; // Buffer
};


} // end namespace gdcm

#endif //GDCMUIDGENERATOR_H
```

13.445 gdcmUUIDGenerator.h File Reference

`#include "gdcmTypes.h"`
Include dependency graph for gdcmUUIDGenerator.h:



Classes

- class gdcm::UUIDGenerator

  Class for generating unique UUID.

Namespaces

- namespace gdcm

13.446 gdcmUUIDGenerator.h

Go to the documentation of this file.

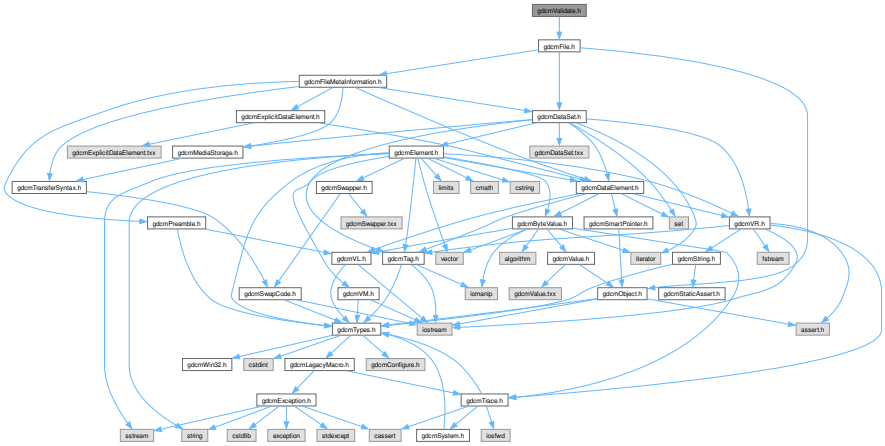
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMUUIDGENERATOR_H
#define GDCMUUIDGENERATOR_H

#include "gdcmTypes.h"

namespace gdcm
{

```

```
class GDCM_EXPORT UUIDGenerator
{
public:
  const char* Generate();

  static bool IsValid(const char *uid);

private:
  std::string Unique; // Buffer
};

} // end namespace gdcm

#endif //GDCMUUIDGENERATOR_H
```

13.447 gdcmValidate.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmValidate.h:



Classes

- class gdcm::Validate

  Validate class.

Namespaces

- namespace gdcm

13.448 gdcmValidate.h

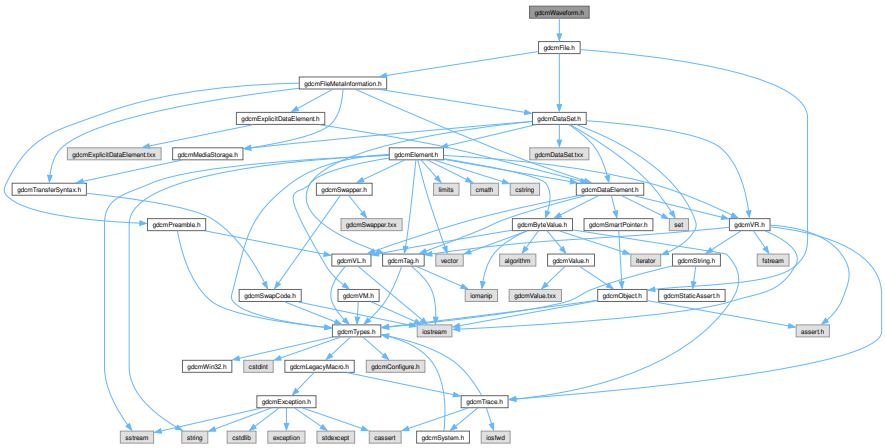
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMVALIDATE_H
#define GDCMVALIDATE_H

#include "gdcmFile.h"

namespace gdcm
{

class GDCM_EXPORT Validate
{
public:
  Validate();
  ~Validate();

  void SetFile(File const &f) { F = &f; }
  const File& GetValidatedFile() { return V; }

  void Validation();

protected:
  const File *F;
  File V; // Validated file
};

} // end namespace gdcm

#endif //GDCMVALIDATE_H
```

13.449 gdcmWaveform.h File Reference

`#include "gdcmFile.h"`

Include dependency graph for gdcmWaveform.h:



Classes

- class gdcm::Waveform

  Waveform class.

Namespaces

- namespace gdcm

13.450 gdcmWaveform.h

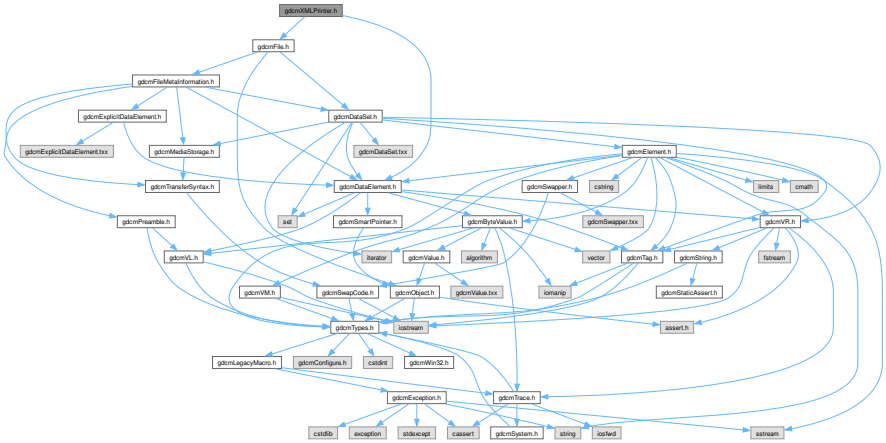
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMWAVEFORM_H
#define GDCMWAVEFORM_H

#include "gdcmFile.h"

namespace gdcm
{
class GDCM_EXPORT Waveform
{
public:
  Waveform() = default;

private:
};

} // end namespace gdcm

#endif //GDCMWAVEFORM_H
```

13.451 gdcmXMLPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmXMLPrinter.h:



Classes

- class gdcm::XMLPrinter

Namespaces

- namespace gdcm

13.452 gdcmXMLPrinter.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMXMLPRINTER_H
#define GDCMXMLPRINTER_H

/*

The Normative version of the XML Schema for the Native DICOM Model follows:



start = element NativeDicomModel { DicomDataSet }

# A DICOM Data Set is as defined in PS3.5. It does not appear
# as an XML Element, since it does not appear in the binary encoded
# DICOM objects. It exists here merely as a documentation aid.

```

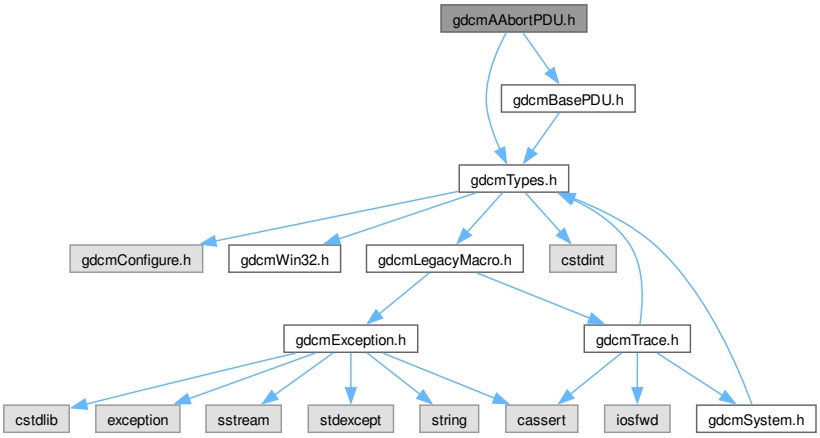
```
DicomDataSet = DicomAttribute*
DicomAttribute = element DicomAttribute {
Tag, VR, Keyword?, PrivateCreator?,
( BulkData | Value+ | Item+ | PersonName+ )?
}

BulkData = element BulkData{ UUID }
Value = element Value { Number, xsd:string }
Item = element Item { Number, DicomDataSet }
PersonName = element PersonName {
Number,
element SingleByte { NameComponents }?,
element Ideographic { NameComponents }?,
element Phonetic
{ NameComponents }?
}

NameComponents =
element FamilyName {xsd:string}?,
element GivenName {xsd:string}?,
element MiddleName {xsd:string}?,
element NamePrefix {xsd:string}?,
element NameSuffix {xsd:string}?

# keyword is the attribute tag from PS3.6
# (derived from the DICOM Attribute's name)
Keyword = attribute keyword { xsd:token }
# canonical XML definition of Hex, with lowercase letters disallowed
Tag = attribute tag { xsd:string{ minLength="8" maxLength="8" pattern="[0-9A-F]{8}" } }
VR = attribute vr { "AE" | "AS" | "AT"| "CS" | "DA" | "DS" | "DT" | "FL" | "FD"
| "IS" | "LO" | "LT" | "OB" | "OF" | "OW" | "PN" | "SH" | "SL"
| "SQ" | "SS" | "ST" | "TM" | "UI" | "UL" | "UN" | "US" | "UT" }
PrivateCreator = attribute privateCreator{ xsd:string }
UUID = attribute uuid { xsd:string }
Number = attribute number { xsd:positiveInteger }


*/

#include "gdcmFile.h"
#include "gdcmDataElement.h"

namespace gdcm
{

class DataSet;
class DictEntry;
class Dicts;

class GDCM_EXPORT XMLPrinter
{
public:
  XMLPrinter();
  virtual ~XMLPrinter();

  // Set file
  void SetFile(File const &f) { F = &f; }



  typedef enum {

    OnlyUUID = 0 ,
    LOADBULKDATA = 1

  } PrintStyles;

  // Set PrintStyle value
  void SetStyle(PrintStyles ps)
  {
    PrintStyle = ps;
  }

  // Get PrintStyle value
  PrintStyles GetPrintStyle() const
  {
    return PrintStyle;
  }

  // Print
  void Print(std::ostream& os);
```

```
  // Print an individual dataset
  void PrintDataSet(const DataSet &ds, const TransferSyntax & ts, std::ostream& os);

  //void PrintUID(std::ostream &os);

  virtual void HandleBulkData(const char *uuid, const TransferSyntax &ts,
    const char *bulkdata, size_t bulklen);

protected:

  VR PrintDataElement(std::ostream &os, const Dicts &dicts, const DataSet & ds, const DataElement &de,
const TransferSyntax & ts);

  void PrintSQ(const SequenceOfItems *sqi, const TransferSyntax & ts, std::ostream &os);

  PrintStyles PrintStyle;

  const File *F;

};

} // end namespace gdcm

#endif //GDCMXMLPRINTER_H
```

13.453 gdcmAAbortPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAbortPDU.h:



Classes

- class gdcm::network::AAbortPDU

  AAbortPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.454 gdcmAAbortPDU.h

Go to the documentation of this file.

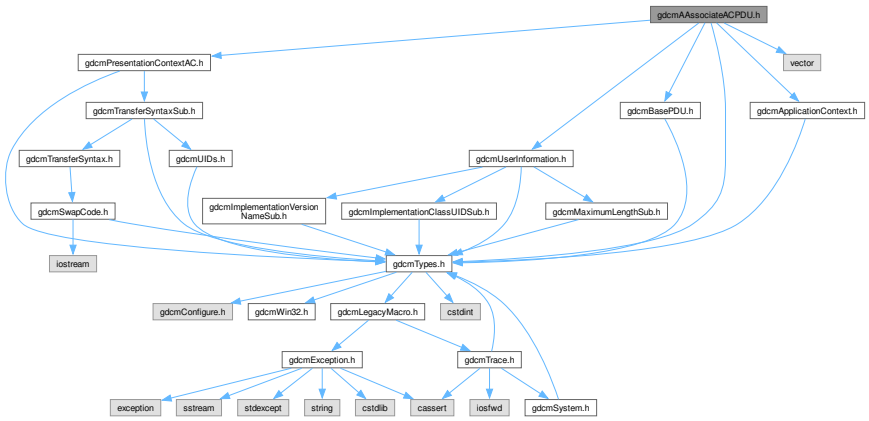
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMAABORTPDU_H
#define GDCMAABORTPDU_H

#include "gdcmTypes.h"
#include "gdcmBasePDU.h"

namespace gdcm
{

namespace network
{

class GDCM_EXPORT AAbortPDU : public BasePDU
{
public:
  AAbortPDU();
  std::istream &Read(std::istream &is) override;
  const std::ostream &Write(std::ostream &os) const override;

  size_t Size() const override;
  void Print(std::ostream &os) const override;

  bool IsLastFragment() const override { return true; }

  void SetSource(const uint8_t s);
  void SetReason(const uint8_t r);

private:
  static const uint8_t ItemType; // PDUType ?
  static const uint8_t Reserved2;
  uint32_t ItemLength; // PDU Length
  static const uint8_t Reserved7;
  static const uint8_t Reserved8;
  uint8_t Source;
  uint8_t Reason; // diag
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMAABORTPDU_H
```

13.455 gdcmAAssociateACPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmApplicationContext.h"
```

```
#include "gdcmPresentationContextAC.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"
#include <vector>
```

Include dependency graph for gdcmAAssociateACPDU.h:



Classes

- class gdcm::network::AAssociateACPDU

  AAssociateACPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.456 gdcmAAssociateACPDU.h

Go to the documentation of this file.

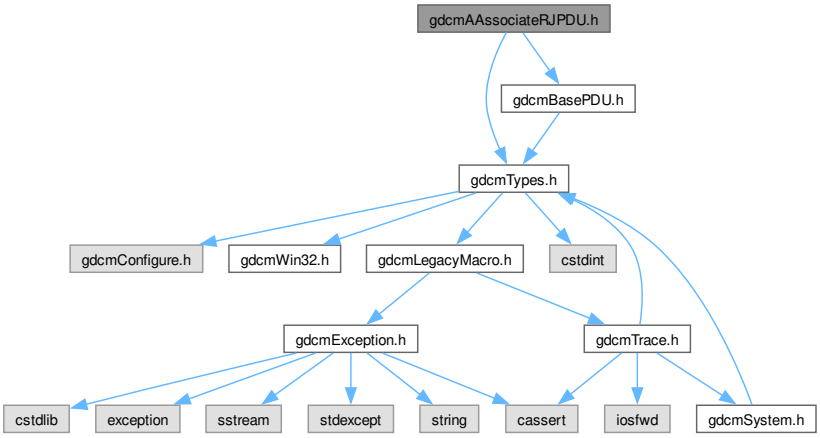
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMAASSOCIATEACPDU_H
#define GDCMAASSOCIATEACPDU_H

#include "gdcmTypes.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextAC.h"
```

```
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"

#include <vector>

namespace gdcm
{

namespace network
{
class AAssociateRQPDU;

class AAssociateACPDU : public BasePDU
{
public:
  AAssociateACPDU();
  std::istream &Read(std::istream &is) override;
  const std::ostream &Write(std::ostream &os) const override;

  void AddPresentationContextAC( PresentationContextAC const &pcac );

  typedef std::vector<PresentationContextAC>::size_type SizeType;
  const PresentationContextAC &GetPresentationContextAC( SizeType i ) {
    assert( !PresContextAC.empty() && i < PresContextAC.size() );
    return PresContextAC[i];
  }
  SizeType GetNumberOfPresentationContextAC() const {
    return PresContextAC.size();
  }
  const UserInformation &GetUserInformation() const { return UserInfo; }

  SizeType Size() const override;

  void Print(std::ostream &os) const override;
  bool IsLastFragment() const override { return true; }

  void InitFromRQ( AAssociateRQPDU const & rqpdu );
protected:
  friend class AAssociateRQPDU;
  void SetCalledAETitle(const char calledaetitle[16]);
  void SetCallingAETitle(const char callingaetitle[16]);

private:
  void InitSimple( AAssociateRQPDU const & rqpdu );

private:
  static const uint8_t ItemType; // PDUType ?
  static const uint8_t Reserved2;
  uint32_t PDULength; // len of
  static const uint16_t ProtocolVersion;
  static const uint16_t Reserved9_10;

  // This reserved field shall be sent with a value identical to the value
  // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
  // shall not be tested when received.
  char Reserved11_26[16];
  // This reserved field shall be sent with a value identical to the value
  // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
  // shall not be tested when received.
  char Reserved27_42[16];
  // This reserved field shall be sent with a value identical to the value
  // received in the same field of the A-ASSOCIATE-RQ PDU, but its value
  // shall not be tested when received.
  char Reserved43_74[32];
  /*
  75-xxx Variable items This variable field shall contain the following items: one Application
  Context Item, one or more Presentation Context Item(s) and one User
  Information Item. For a complete description of these items see Sections
  7.1.1.2, 7.1.1.14, and 7.1.1.6.
   */
  ApplicationContext AppContext;
  std::vector<PresentationContextAC>  PresContextAC;
  UserInformation UserInfo;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMAASSOCIATEACPDU_H
```

13.457 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



Classes

- class gdcm::network::AAssociateRJPDU

  AAssociateRJPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.458 gdcmAAssociateRJPDU.h

Go to the documentation of this file.

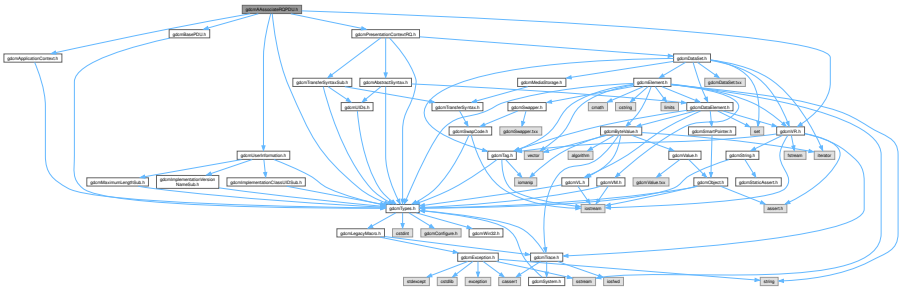
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMAASSOCIATERJPDU_H
```

```
#define GDCMAASSOCIATERJPDU_H

#include "gdcmTypes.h"
#include "gdcmBasePDU.h"

namespace gdcm
{

namespace network
{

class AAssociateRJPDU : public BasePDU
{
public:
  AAssociateRJPDU();
  std::istream &Read(std::istream &is) override;
  const std::ostream &Write(std::ostream &os) const override;
  void Print(std::ostream &os) const override;
  size_t Size() const override;
  bool IsLastFragment() const override { return true; }
private:
  static const uint8_t ItemType; // PDUType ?
  static const uint8_t Reserved2;
  uint32_t ItemLength; // PDU Length ?
  static const uint8_t Reserved8;
  uint8_t Result;
  uint8_t Source;
  uint8_t Reason; // diag ?
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMAASSOCIATERJPDU_H
```

13.459 gdcmAAssociateRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRQPDU.h:



Classes

- class gdcm::network::AAssociateRQPDU

  AAssociateRQPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.460 gdcmAAssociateRQPDU.h

Go to the documentation of this file.

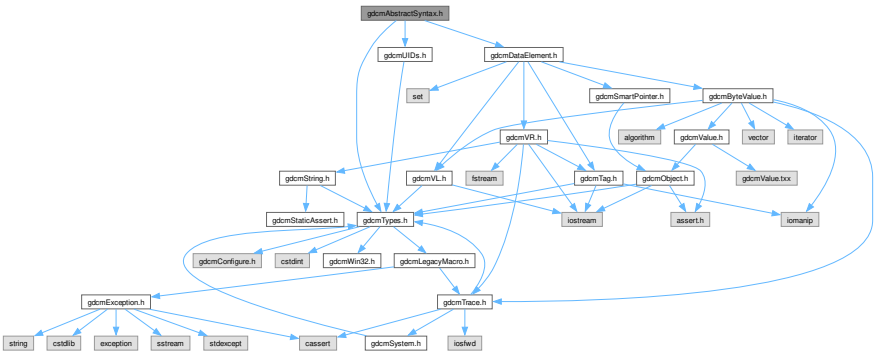
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMAASSOCIATERQPDU_H
#define GDCMAASSOCIATERQPDU_H

#include "gdcmTypes.h"
#include "gdcmVR.h" // AEComp
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmUserInformation.h"
#include "gdcmBasePDU.h"

namespace gdcm
{

namespace network
{

class AAssociateACPDU;
class AAssociateRQPDU : public BasePDU
{
public:
  AAssociateRQPDU();
  std::istream &Read(std::istream &is) override;
  const std::ostream &Write(std::ostream &os) const override;
  size_t Size() const override;
  void AddPresentationContext( PresentationContextRQ const &pc );

  void SetCalledAETitle(const char calledaetitle[16]);
  std::string GetCalledAETitle() const { return std::string(CalledAETitle,16); }

  void SetCallingAETitle(const char callingaetitle[16]);
  std::string GetCallingAETitle() const { return std::string(CallingAETitle,16); }

  static bool IsAETitleValid(const char title[16]);

  //void InitFromRQ( AAssociateACPDU & acpdu );

  void Print(std::ostream &os) const override;

  AAssociateRQPDU(const AAssociateRQPDU &pdu):BasePDU(pdu)
    {
    assert( 0 );
    }
  //this function fails to compile on windows.
//  AAssociateRQPDU &operator=(const AAssociateRQPDU &_val)
//    {
//    assert( 0 );
//    }

  typedef std::vector<PresentationContextRQ>::size_type SizeType;
  SizeType GetNumberOfPresentationContext() const {
    return PresContext.size();
  }
```

```
  PresentationContextRQ const &GetPresentationContext(SizeType i) const {
    assert( !PresContext.empty() && i < PresContext.size() );
    return PresContext[i];
  }
  typedef std::vector<PresentationContextRQ> PresentationContextArrayType;
  PresentationContextArrayType const &GetPresentationContexts() { return PresContext; }

  const PresentationContextRQ *GetPresentationContextByID(uint8_t i) const;
  const PresentationContextRQ *GetPresentationContextByAbstractSyntax(AbstractSyntax const & absyn ) const;
  bool IsLastFragment() const override { return true; }

  const UserInformation & GetUserInformation() const { return UserInfo; }
  void SetUserInformation( UserInformation const & ui );

protected:
  friend class AAssociateACPDU;
  std::string GetReserved43_74() const;

private:
  // 1 PDU-type 01H
  static const uint8_t ItemType; // PDUType ?
  // 2 Reserved This reserved field shall be sent with a value 00H but not tested to this value when received.
  static const uint8_t Reserved2;
  /* 3-6 PDU-length This PDU-length shall be the number of bytes from the first byte of the
     following field to the last byte of the variable field. It shall be encoded as
     an unsigned binary number
   */
  uint32_t ItemLength; // PDU Length
  /*
  7-8 Protocol-version This two byte field shall use one bit to identify each version of the
  DICOM UL protocol supported by the calling end-system. This is
  Version 1 and shall be identified with bit 0 set. A receiver of this PDU
  implementing only this version of the DICOM UL protocol shall only test
  that bit 0 is set.
   */
  static const uint16_t ProtocolVersion;
  /*
  9-10 Reserved This reserved field shall be sent with a value 0000H but not tested to
  this value when received.
   */
  static const uint16_t Reserved9_10;
  /*
  11-26 Called-AE-title Destination DICOM Application Name. It shall be encoded as 16
  characters as defined by the ISO 646:1990-Basic G0 Set with leading
  and trailing spaces (20H) being non-significant. The value made of 16
  spaces (20H) meaning "no Application Name specified" shall not be
  used. For a complete description of the use of this field, see Section
  7.1.1.4.
   */
  char CalledAETitle[16];
  /*
  27-42 Calling-AE-title Source DICOM Application Name. It shall be encoded as 16
  characters as defined by the ISO 646:1990-Basic G0 Set with leading
  and trailing spaces (20H) being non-significant. The value made of 16
  spaces (20H) meaning "no Application Name specified" shall not be
  used. For a complete description of the use of this field, see Section
  7.1.1.3.
   */
  char CallingAETitle[16];
  /*
  43-74 Reserved This reserved field shall be sent with a value 00H for all bytes but not
  tested to this value when received
   */
  char Reserved43_74[32]; // { 0 }
  /*
  75-xxx Variable items This variable field shall contain the following items: one Application
  Context Item, one or more Presentation Context Items and one User
  Information Item. For a complete description of the use of these items
  see Sections 7.1.1.2, 7.1.1.13, and 7.1.1.6.
   */
  ApplicationContext AppContext;
  std::vector<PresentationContextRQ> PresContext;
  UserInformation UserInfo;
};

} // end namespace network
} // end namespace gdcm

#endif //GDCMAASSOCIATERQPDU_H
```

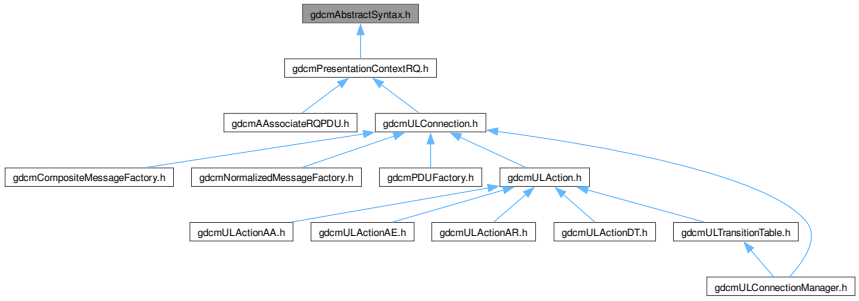
13.461 gdcmAbstractSyntax.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmAbstractSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::AbstractSyntax

  AbstractSyntax.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.462 gdcmAbstractSyntax.h

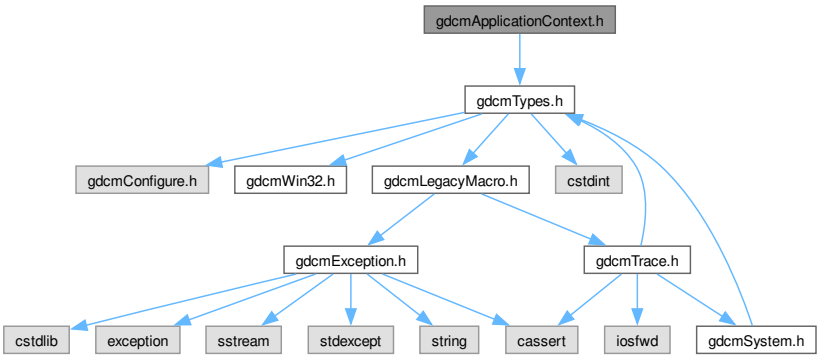
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMABSTRACTSYNTAX_H
#define GDCMABSTRACTSYNTAX_H

#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include "gdcmDataElement.h"

namespace gdcm
{

namespace network
{

class AbstractSyntax
{
public:
  AbstractSyntax();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  void SetName( const char *name ) { UpdateName( name ); }
  const char *GetName() const { return Name.c_str(); }

  // accept a UIDs::TSType also...
  void SetNameFromUID( UIDs::TSName tsname );
  //now that the PresentationContext messes around with UIDs and returns a string
  //use that string as well.
  //void SetNameFromUIDString( const std::string& inUIDName );

  size_t Size() const;

  void Print(std::ostream &os) const;

  bool operator==(const AbstractSyntax & as) const
    {
    return Name == as.Name;
    }

  DataElement GetAsDataElement() const;

private:
  void UpdateName( const char *name );
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength; // len of
  std::string /*AbstractSyntax*/ Name; // UID
};

} // end namespace network
} // end namespace gdcm

#endif //GDCMABSTRACTSYNTAX_H
```

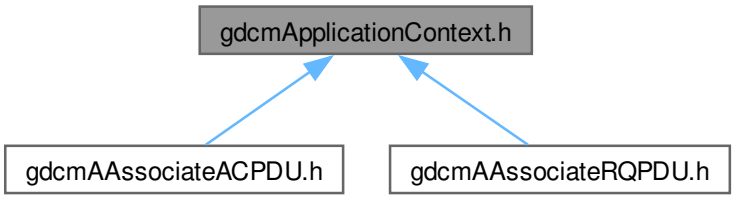
13.463 gdcmApplicationContext.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmApplicationContext.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ApplicationContext

  ApplicationContext.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.464 gdcmApplicationContext.h

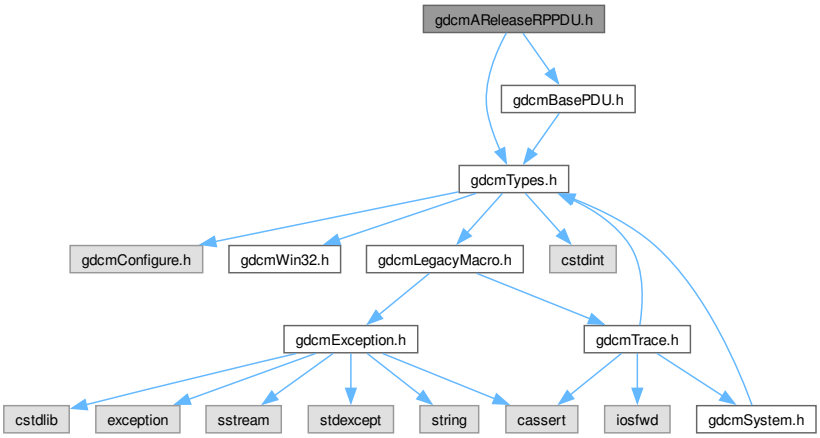
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMAPPLICATIONCONTEXT_H
#define GDCMAPPLICATIONCONTEXT_H

#include "gdcmTypes.h"

namespace gdcm
{

namespace network
{

class ApplicationContext
{
public:
  ApplicationContext();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  void SetName( const char *name ) { UpdateName( name ); }
  const char *GetName() const { return Name.c_str(); }
  size_t Size() const;

  //static const uint8_t GetItemType() { return ItemType; }
  void Print(std::ostream &os) const;

private:
  void UpdateName( const char *name );
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength; // len of application context name
  std::string /*ApplicationContext*/ Name; // UID
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMAPPLICATIONCONTEXT_H
```

13.465 gdcmAReleaseRPPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRPPDU.h:



Classes

- class gdcm::network::AReleaseRPPDU

  AReleaseRPPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.466 gdcmAReleaseRPPDU.h

Go to the documentation of this file.

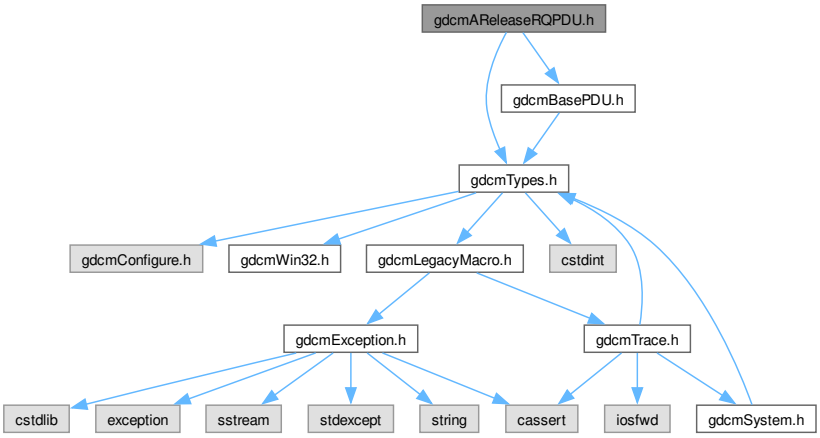
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMARELEASERPPDU_H
#define GDCMARELEASERPPDU_H

#include "gdcmTypes.h"
#include "gdcmBasePDU.h"

namespace gdcm
{

```

```
namespace network
{

class AReleaseRPPDU : public BasePDU
{
public:
  AReleaseRPPDU();
  std::istream &Read(std::istream &is) override;
  const std::ostream &Write(std::ostream &os) const override;
  size_t Size() const override;
  void Print(std::ostream &os) const override;
  bool IsLastFragment() const override { return true; }
private:
  static const uint8_t ItemType; // PDUType ?
  static const uint8_t Reserved2;
  uint32_t ItemLength; // PDU Length
  static const uint32_t Reserved7_10;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMARELEASERPPDU_H
```

13.467 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRQPDU.h:



Classes

- class gdcm::network::AReleaseRQPDU

  AReleaseRQPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

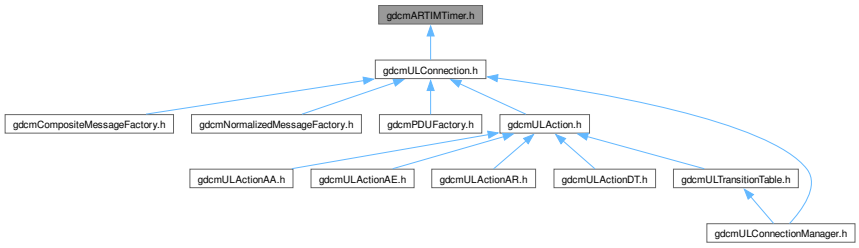
13.468 gdcmAReleaseRQPDU.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMARELEASERQPDU_H
#define GDCMARELEASERQPDU_H

#include "gdcmTypes.h"
#include "gdcmBasePDU.h"

namespace gdcm
{

namespace network
{

class AReleaseRQPDU : public BasePDU
{
public:
  AReleaseRQPDU();
  std::istream &Read(std::istream &is) override;
  const std::ostream &Write(std::ostream &os) const override;
  size_t Size() const override;
  void Print(std::ostream &os) const override;
  bool IsLastFragment() const override { return true; }
private:
  static const uint8_t ItemType; // PDUType ?
  static const uint8_t Reserved2;
  uint32_t ItemLength; // PDU Length
  static const uint32_t Reserved7_10;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMARELEASERQPDU_H
```

13.469 gdcmARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ARTIMTimer

  ARTIMTimer.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.470 gdcmARTIMTimer.h

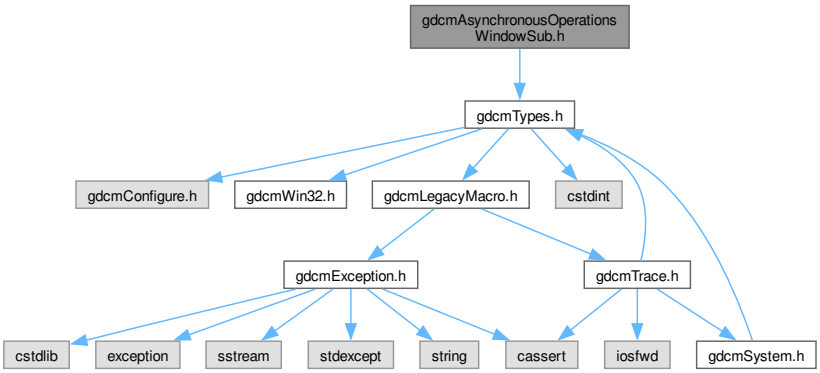
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMARTIMTIMER_H
#define GDCMARTIMTIMER_H

namespace gdcm {
  namespace network{
class ARTIMTimer
{
    private:
      double mStartTime; //ms timing should be good enough, but there are also
      //high-resolution timing options.  Those return doubles.  For now,
      //go with integer timing solutions based on milliseconds (DWORD on windows),
      //but leave as doubles to ease transitions to other timing methods.

      double mTimeOut;
      //once GetCurrentTime() -mStartTime > mTimeout, GetHasExpired returns true.

      double GetCurrentTime() const;//a platform-specific implementation of getting the
      //current time.

    public:
      ARTIMTimer(); //initiates the start and timeout at -1;
      void Start(); //'start' the timer by getting the current wall time
      void Stop();//'stop' the timer by resetting the 'start' to -1;
      void SetTimeout(double inTimeout);
      double GetTimeout() const;

      double GetElapsedTime() const;

      bool GetHasExpired() const;

    };
  }
}

#endif //GDCMARTIMTIMER_H
```

13.471 gdcmAsynchronousOperationsWindowSub.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



Classes

- class gdcm::network::AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.472 gdcmAsynchronousOperationsWindowSub.h

Go to the documentation of this file.

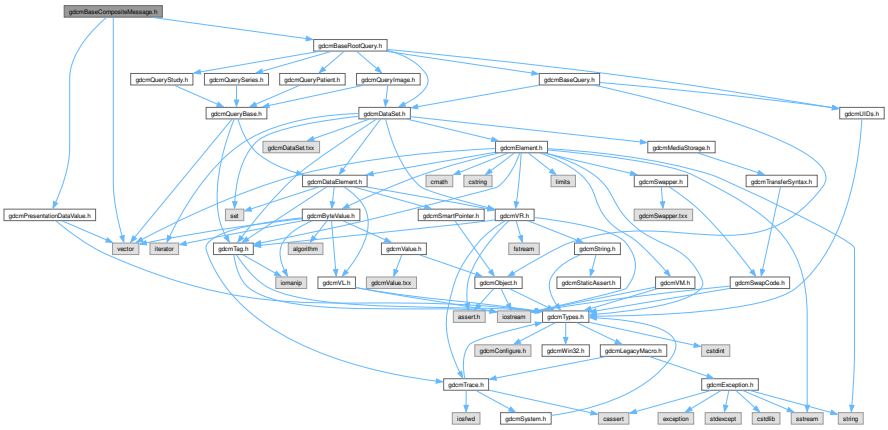
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
#define GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H

#include "gdcmTypes.h"

namespace gdcm
```

```
{

namespace network
{

class AsynchronousOperationsWindowSub
{
public:
  AsynchronousOperationsWindowSub();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;
  void Print(std::ostream &os) const;

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength;
  uint16_t MaximumNumberOperationsInvoked;
  uint16_t MaximumNumberOperationsPerformed;
};

} // end namespace network

} // end namespace gdcm

#endif // GDCMASYNCHRONOUSOPERATIONSWINDOWSUB_H
```

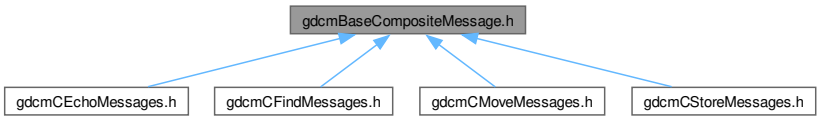
13.473 gdcmBaseCompositeMessage.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
```

Include dependency graph for gdcmBaseCompositeMessage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::BaseCompositeMessage

  BaseCompositeMessage.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.474 gdcmBaseCompositeMessage.h

Go to the documentation of this file.

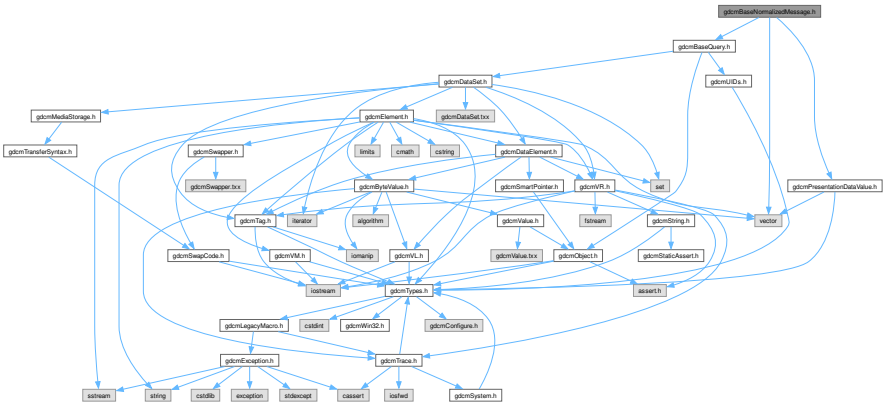
```
00001 /*=========================================================================
00002  *
00003  *  Copyright NumFOCUS
00004  *
00005  *  Licensed under the Apache License, Version 2.0 (the "License");
00006  *  you may not use this file except in compliance with the License.
00007  *  You may obtain a copy of the License at
00008  *
00009  *         http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *  Unless required by applicable law or agreed to in writing, software
00012  *  distributed under the License is distributed on an "AS IS" BASIS,
00013  *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *  See the License for the specific language governing permissions and
00015  *  limitations under the License.
00016  *
00017  *=========================================================================*/
00018 #ifndef GDCMBASECOMPOSITEMESSSAGE_H
00019 #define GDCMBASECOMPOSITEMESSSAGE_H
00020
00021 #include "gdcmPresentationDataValue.h"
00022 #include "gdcmBaseRootQuery.h"
00023
00024 #include <vector>
00025
00026 namespace gdcm
00027 {
00028   namespace network
00029     {
00030 class ULConnection;
00052 class BaseCompositeMessage
00053 {
00054     public:
00055       virtual ~BaseCompositeMessage() = default;
00056       //construct the appropriate pdv and dataset for this message
00057       //for instance, setting tag 0x0,0x100 to the appropriate value
00058       //the pdv, as described in Annex E of 3.8-2009, is the first byte
```

```
      //of the message (the MessageHeader), and then the subsequent dataset
      //that describes the operation.
      virtual std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
        const BaseRootQuery * inRootQuery) = 0;
    };
  }
}
#endif //BASECOMPOSITEMESSSAGE_H
```

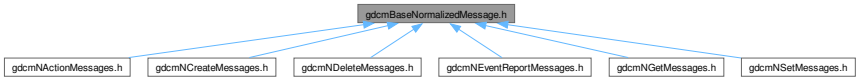
13.475 gdcmBaseNormalizedMessage.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmBaseQuery.h"
#include <vector>
```

Include dependency graph for gdcmBaseNormalizedMessage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::BaseNormalizedMessage

  BaseNormalizedMessage.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.476 gdcmBaseNormalizedMessage.h

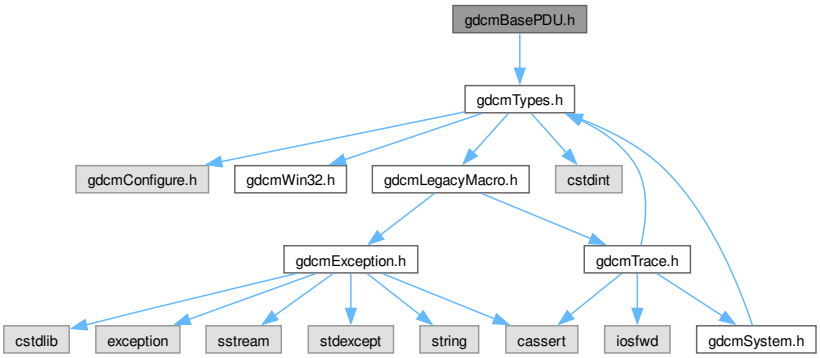
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2014 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMBASENORMALIZEDMESSSAGE_H
#define GDCMBASENORMALIZEDMESSSAGE_H

#include "gdcmPresentationDataValue.h"
#include "gdcmBaseQuery.h"

#include <vector>

namespace gdcm
{
  namespace network
    {
class ULConnection;
class BaseNormalizedMessage
{
    public:
      virtual ~BaseNormalizedMessage() = default;
      //construct the appropriate pdv and dataset for this message
      //for instance, setting tag 0x0,0x100 to the appropriate value
      //the pdv, as described in Annex E of 3.8-2009, is the first byte
      //of the message (the MessageHeader), and then the subsequent dataset
      //that describes the operation.
      virtual std::vector<PresentationDataValue> ConstructPDV(  const ULConnection &inConnection,
                                const BaseQuery * inQuery) = 0;
    };
  }
}
#endif //GDCMBASENORMALIZEDMESSSAGE_H
```

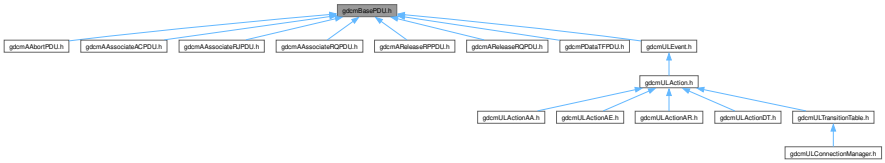
13.477 gdcmBasePDU.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmBasePDU.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::BasePDU

  BasePDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.478 gdcmBasePDU.h

Go to the documentation of this file.

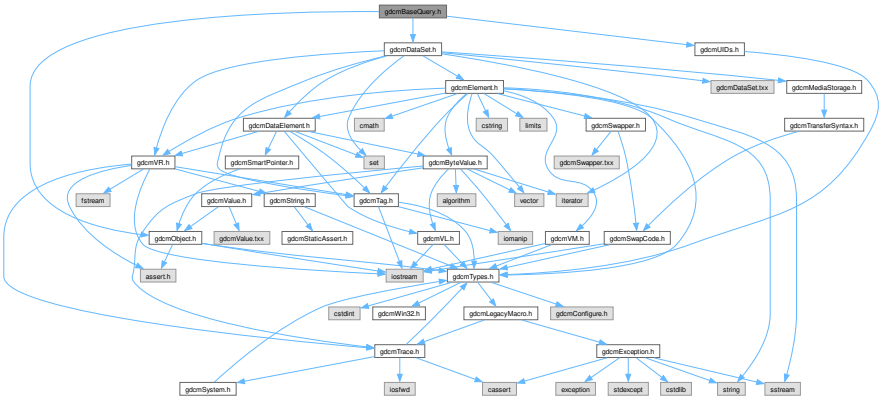
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMBASEPDU_H
#define GDCMBASEPDU_H

#include "gdcmTypes.h"

namespace gdcm
{
namespace network
{

class BasePDU
{
public:
  virtual ~BasePDU() = default;

  virtual std::istream &Read(std::istream &is) = 0;
  virtual const std::ostream &Write(std::ostream &os) const = 0;

  virtual size_t Size() const = 0;
  virtual void Print(std::ostream &os) const = 0;

```

```
  virtual bool IsLastFragment() const = 0;
};

} // end namespace network
} // end namespace gdcm

#endif // GDCMBASEPDU_H
```

13.479 gdcmBaseQuery.h File Reference

```
#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmObject.h"
```

Include dependency graph for gdcmBaseQuery.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::BaseQuery

  BaseQuery.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ENQueryType {
  gdcm::eCreateMMPS = 0 ,
  gdcm::eSetMMPS }

13.480 gdcmBaseQuery.h

Go to the documentation of this file.

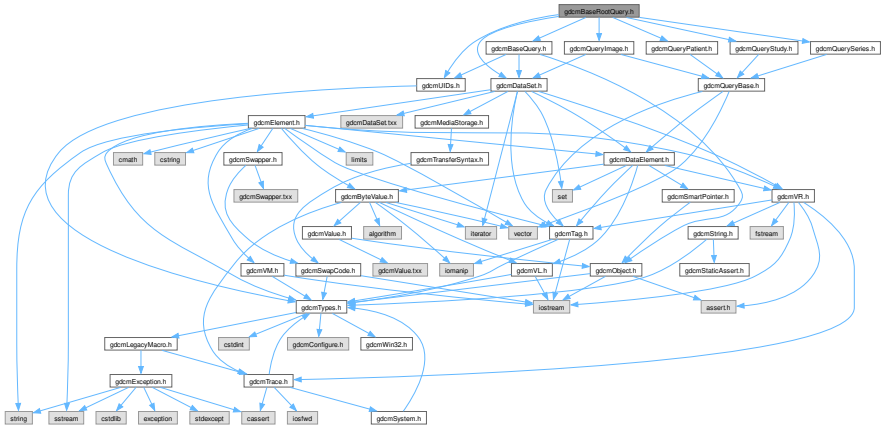
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMBASEQUERY_H
#define GDCMBASEQUERY_H

#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmObject.h"

namespace gdcm
{
  class QueryFactory;
  class DictEntry;

  enum ENQueryType
    {
    eCreateMMPS = 0,
    eSetMMPS
    };
class GDCM_EXPORT BaseQuery : public Object
{
  //these four classes contain the required, unique, and optional tags from the standard.
  //used both to list the tags as well as to validate a dataset, if ever we were to do so.
protected:

  DataSet mDataSet;
  friend class QueryFactory;
  BaseQuery();

  std::string mSopInstanceUID;

  void SetSearchParameter(const Tag& inTag, const DictEntry& inDictEntry, const std::string& inValue);

  bool ValidDataSet( const DataSet & dataSetToValid, const DataSet & dataSetReference  ) const ;
public:
  ~BaseQuery() override;

  void SetSearchParameter(const Tag& inTag, const std::string& inValue);
  void SetSearchParameter(const std::string& inKeyword, const std::string& inValue);

  const std::ostream &WriteHelpFile(std::ostream &os);

  //this function allows writing of the query to disk for storing for future use
  //virtual in case it needs to be overridden
  //returns false if the operation failed
  bool WriteQuery(const std::string& inFileName);

  DataSet const & GetQueryDataSet() const;
```

```
  DataSet & GetQueryDataSet();
  void AddQueryDataSet(const DataSet & ds);

  virtual bool ValidateQuery( bool inStrict = true ) const = 0;

  virtual UIDs::TSName GetAbstractSyntaxUID() const = 0;
  std::string GetSOPInstanceUID() const { return mSopInstanceUID ; }
  void SetSOPInstanceUID( const std::string & iSopInstanceUID ) { mSopInstanceUID = iSopInstanceUID ; }



  void Print(std::ostream &os) const override;
};

} // end namespace gdcm

#endif //GDCMBASEROOTQUERY_H
```

13.481 gdcmBaseRootQuery.h File Reference

```
#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmBaseQuery.h"
#include "gdcmQueryPatient.h"
#include "gdcmQueryStudy.h"
#include "gdcmQuerySeries.h"
#include "gdcmQueryImage.h"
```

Include dependency graph for gdcmBaseRootQuery.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::BaseRootQuery

  BaseRootQuery.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::EQueryLevel {
  gdcm::ePatient = 0 ,
  gdcm::eStudy = 1 ,
  gdcm::eSeries = 2 ,
  gdcm::eImage = 3 }
- enum gdcm::EQueryType {
  gdcm::eFind = 0 ,
  gdcm::eMove ,
  gdcm::eWLMFind }

13.482 gdcmBaseRootQuery.h

Go to the documentation of this file.

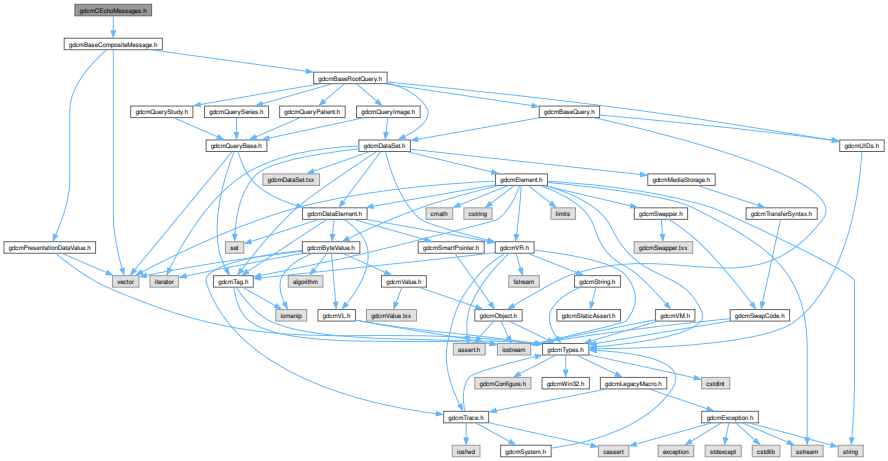
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMBASEROOTQUERY_H
#define GDCMBASEROOTQUERY_H

#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmBaseQuery.h"
#include "gdcmQueryPatient.h"
#include "gdcmQueryStudy.h"
#include "gdcmQuerySeries.h"
#include "gdcmQueryImage.h"

namespace gdcm
{
  class QueryFactory;
  class DictEntry;

  enum EQueryLevel
    {
    // -1 is reserved do not use
    ePatient = 0,
```

```
    eStudy   = 1,
    eSeries  = 2,
    eImage   = 3
    };
  enum EQueryType
    {
    eFind= 0,
    eMove,
    eWLMFind
    };

class GDCM_EXPORT BaseRootQuery : public BaseQuery
{
  //these four classes contain the required, unique, and optional tags from the standard.
  //used both to list the tags as well as to validate a dataset, if ever we were to do so.
protected:
  QueryPatient mPatient;
  QueryStudy mStudy;
  QuerySeries mSeries;
  QueryImage mImage;

  friend class QueryFactory;
  BaseRootQuery();

  ERootType mRootType; //set in construction, and it's something else in the study root type
  std::string mHelpDescription; //used when generating the help output

public:
  ~BaseRootQuery() override = default;

  virtual std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) = 0;

  virtual void InitializeDataSet(const EQueryLevel& inQueryLevel) = 0;

  bool ValidateQuery( bool inStrict = true ) const override = 0;

  static const char *GetQueryLevelString( EQueryLevel ql );
  static int GetQueryLevelFromString( const char * str );

  static QueryBase * Construct(ERootType inRootType, EQueryLevel qlevel);
  EQueryLevel GetQueryLevelFromQueryRoot( ERootType roottype );
};

} // end namespace gdcm

#endif //GDCMBASEROOTQUERY_H
```

13.483 gdcmCEchoMessages.h File Reference

`#include "gdcmBaseCompositeMessage.h"`

Include dependency graph for gdcmCEchoMessages.h:



Classes

- class gdcm::network::CEchoRQ

  CEchoRQ.
- class gdcm::network::CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.484 gdcmCEchoMessages.h

Go to the documentation of this file.

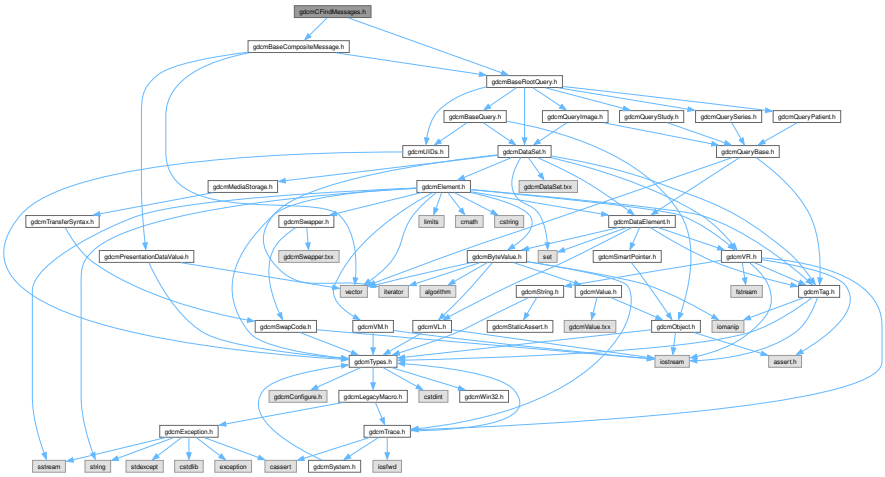
```
00001 /*=========================================================================
00002  *
00003  *  Copyright NumFOCUS
00004  *
00005  *  Licensed under the Apache License, Version 2.0 (the "License");
00006  *  you may not use this file except in compliance with the License.
00007  *  You may obtain a copy of the License at
00008  *
00009  *         http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *  Unless required by applicable law or agreed to in writing, software
00012  *  distributed under the License is distributed on an "AS IS" BASIS,
00013  *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *  See the License for the specific language governing permissions and
```

```
 *  limitations under the License.
 *
 *=============================================================================*/
#ifndef GDCMCECHOMESSAGES_H
#define GDCMCECHOMESSAGES_H

#include "gdcmBaseCompositeMessage.h"

namespace gdcm{
  namespace network{

class ULConnection;

class CEchoRQ : public BaseCompositeMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
        const BaseRootQuery* inRootQuery) override;
    };

    class CEchoRSP : public BaseCompositeMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    };
  }
}
#endif // GDCMCECHOMESSAGES_H
```

13.485 gdcmCFindMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmCFindMessages.h:



Classes

- class gdcm::network::CFindCancelRQ

  CFindCancelRQ this file defines the messages for the cfind action.
- class gdcm::network::CFindRQ

  CFindRQ.
- class gdcm::network::CFindRSP

  CFindRSP this file defines the messages for the cfind action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.486 gdcmCFindMessages.h

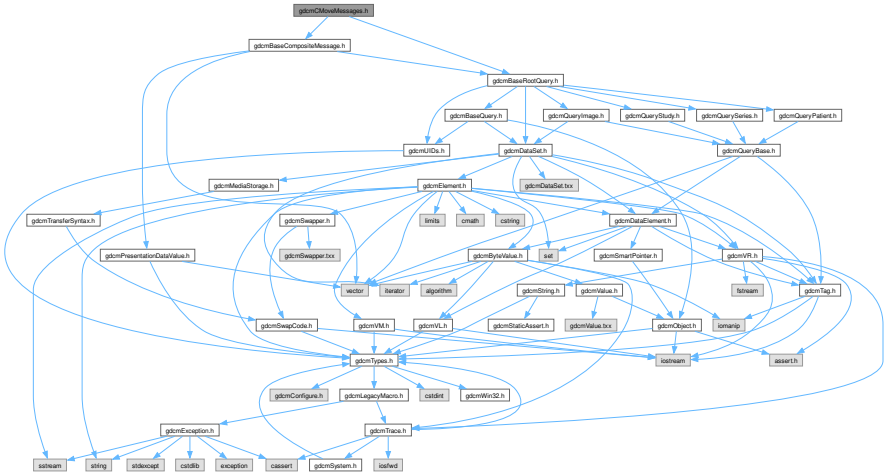
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMCFINDMESSAGES_H
#define GDCMCFINDMESSAGES_H

#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"

namespace gdcm
{
namespace network
{

class CFindRQ : public BaseCompositeMessage
{
public:
  std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
    const BaseRootQuery* inRootQuery) override;
};

class CFindRSP : public BaseCompositeMessage {
public:
  std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
};

  class CFindCancelRQ : public BaseCompositeMessage {
  public:
    std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
  };
}
}
#endif
```

13.487 gdcmCMoveMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmCMoveMessages.h:



Classes

- class gdcm::network::CMoveCancelRq
- class gdcm::network::CMoveRQ

  CMoveRQ.
- class gdcm::network::CMoveRSP

  CMoveRSP this file defines the messages for the cmove action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.488 gdcmCMoveMessages.h

Go to the documentation of this file.

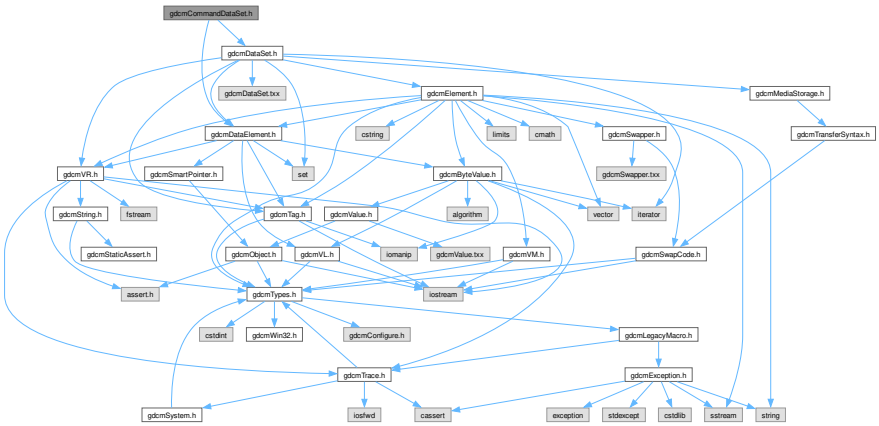
```
00001 /*=========================================================================
00002  *
00003  *  Copyright NumFOCUS
00004  *
00005  *  Licensed under the Apache License, Version 2.0 (the "License");
00006  *  you may not use this file except in compliance with the License.
00007  *  You may obtain a copy of the License at
00008  *
00009  *         http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *  Unless required by applicable law or agreed to in writing, software
00012  *  distributed under the License is distributed on an "AS IS" BASIS,
00013  *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *  See the License for the specific language governing permissions and
00015  *  limitations under the License.
00016  *
00017  *=========================================================================*/
00018 #ifndef GDCMCMOVEMESSAGES_H
```

```
#define GDCMCMOVEMESSAGES_H

#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"

namespace gdcm{
  namespace network{
  class ULConnection;
class CMoveRQ : public BaseCompositeMessage {
      //this class will fulfill the inheritance,
      //but additional information is needed by cmovd
      //namely, the root type or the calling AE-TITLE
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    public:
      std::vector<PresentationDataValue> ConstructPDV(
        const ULConnection &inConnection,
        const BaseRootQuery* inRootQuery) override;
    };

class CMoveRSP : public BaseCompositeMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    };

    class CMoveCancelRq : public BaseCompositeMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    };
  }
}
#endif
```

13.489 gdcmCommandDataSet.h File Reference

```
#include "gdcmDataSet.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmCommandDataSet.h:



Classes

- class gdcm::CommandDataSet

  Class to represent a Command DataSet.

Namespaces

- namespace gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CommandDataSet &val)

13.490 gdcmCommandDataSet.h

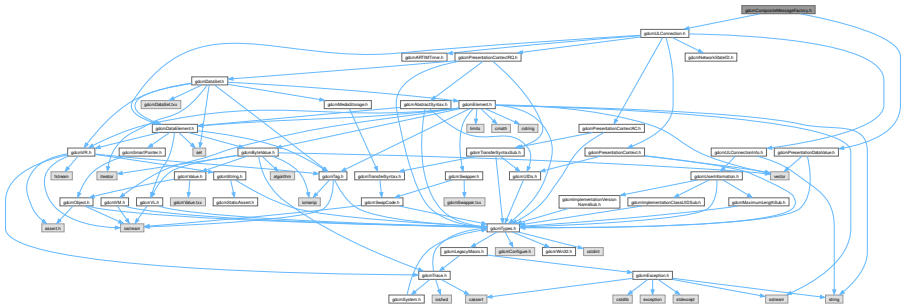
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCOMMANDDATASET_H
#define GDCMCOMMANDDATASET_H

#include "gdcmDataSet.h"
#include "gdcmDataElement.h"

namespace gdcm
{
class GDCM_EXPORT CommandDataSet : public DataSet
{
public:
  CommandDataSet() = default;
 ~CommandDataSet() = default;

  friend std::ostream &operator«(std::ostream &_os, const CommandDataSet &_val);

  // FIXME: no virtual function means: duplicate code...
  void Insert(const DataElement& de) {
    if( de.GetTag().GetGroup() == 0x0000 )
      {
      InsertDataElement( de );
      }
    else
      {
      gdcmErrorMacro( "Cannot add element with group != 0x0000 in the command dataset : " « de );
      }
  }
  void Replace(const DataElement& de) {
    Remove(de.GetTag());
    Insert(de);
  }

  std::istream &Read(std::istream &is);

  std::ostream &Write(std::ostream &os) const;

protected:
};
//-----------------------------------------------------------------------------
inline std::ostream& operator«(std::ostream &os, const CommandDataSet &val)
{
  val.Print( os );
  return os;
}

} // end namespace gdcm

#endif //GDCMFILEMETAINFORMATION_H
```

13.491 gdcmCompositeMessageFactory.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmCompositeMessageFactory.h:



Classes

- class gdcm::network::CompositeMessageFactory

  CompositeMessageFactory.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.492 gdcmCompositeMessageFactory.h

Go to the documentation of this file.

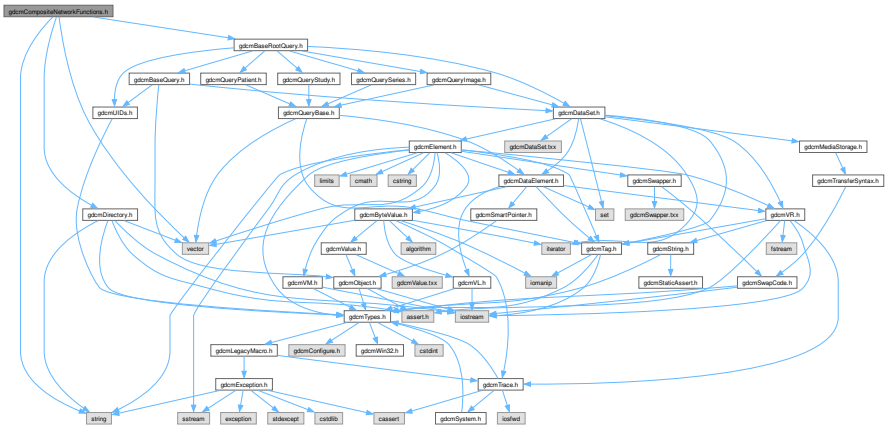
```
00001 /*=========================================================================
00002  *
00003  *  Copyright NumFOCUS
00004  *
00005  *  Licensed under the Apache License, Version 2.0 (the "License");
00006  *  you may not use this file except in compliance with the License.
00007  *  You may obtain a copy of the License at
00008  *
00009  *         http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *  Unless required by applicable law or agreed to in writing, software
00012  *  distributed under the License is distributed on an "AS IS" BASIS,
00013  *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *  See the License for the specific language governing permissions and
00015  *  limitations under the License.
00016  *
00017  *=========================================================================*/
00018 #ifndef GDCMCOMPOSITEMESSAGEFACTORY_H
00019 #define GDCMCOMPOSITEMESSAGEFACTORY_H
00020
00021 #include "gdcmPresentationDataValue.h"
00022 #include "gdcmULConnection.h"
00023
00024 namespace gdcm {
```

```
  class BaseRootQuery;
  class File;
  namespace network {
    class BasePDU;
class CompositeMessageFactory
{
    public:
      //the echo request only needs a properly constructed PDV.
      //find, move, etc, may need something more robust, but since those are
      //easily placed into the appropriate pdatapdu in the pdufactory,
      //this approach without a base class (but done internally) is useful.
      static std::vector<PresentationDataValue> ConstructCEchoRQ(const ULConnection& inConnection);

      static std::vector<PresentationDataValue> ConstructCStoreRQ(const ULConnection& inConnection,const File &file, bool writeDataSet = true );
      static std::vector<PresentationDataValue> ConstructCStoreRSP(const DataSet *inDataSet, const BasePDU* inPC);

      static  std::vector<PresentationDataValue> ConstructCFindRQ(const ULConnection& inConnection, const BaseRootQuery* inRootQuery);

      static  std::vector<PresentationDataValue> ConstructCMoveRQ(const ULConnection& inConnection, const BaseRootQuery* inRootQuery);


    };
  }
}

#endif // GDCMCOMPOSITEMESSAGEFACTORY_H
```

13.493 gdcmCompositeNetworkFunctions.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
#include <string>
```

Include dependency graph for gdcmCompositeNetworkFunctions.h:



Classes

- class gdcm::CompositeNetworkFunctions

  Composite Network Functions.

Namespaces

- namespace gdcm

13.494 gdcmCompositeNetworkFunctions.h

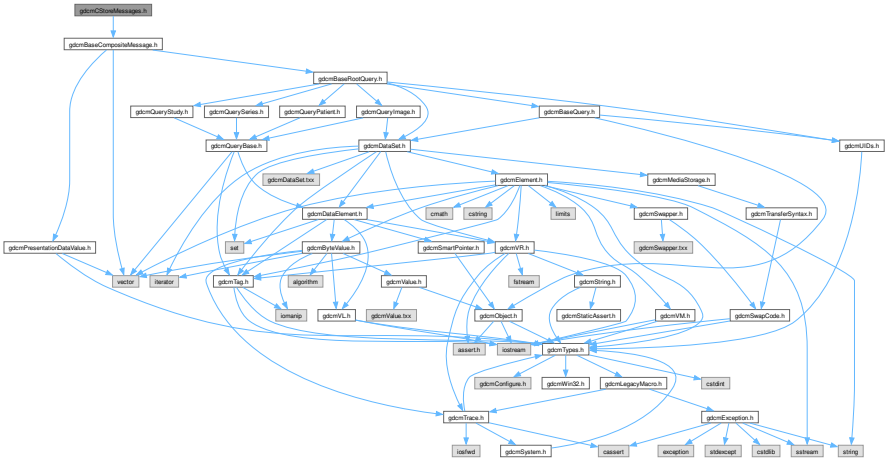
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMCOMPOSITENETWORKFUNCTIONS_H
#define GDCMCOMPOSITENETWORKFUNCTIONS_H

#include "gdcmDirectory.h"
#include "gdcmBaseRootQuery.h" // EQueryLevel / EQueryType

#include <vector>
#include <string>

namespace gdcm
{
class GDCM_EXPORT CompositeNetworkFunctions
{
public:
  static bool CEcho( const char *remote, uint16_t portno, const char *aetitle = nullptr,
    const char *call = nullptr );

  typedef std::pair<Tag, std::string> KeyValuePairType;
  typedef std::vector< KeyValuePairType  > KeyValuePairArrayType;

  static BaseRootQuery* ConstructQuery(ERootType inRootType, EQueryLevel inQueryLevel,
    const DataSet& queryds, EQueryType queryType = eFind );

  static BaseRootQuery* ConstructQuery(ERootType inRootType, EQueryLevel inQueryLevel,
    const KeyValuePairArrayType& keys, EQueryType queryType = eFind );

  static bool CMove( const char *remote, uint16_t portno, const BaseRootQuery* query,
    uint16_t portscp, const char *aetitle = nullptr,
    const char *call = nullptr, const char *outputdir = nullptr);

  static bool CFind( const char *remote, uint16_t portno,
    const BaseRootQuery* query,
    std::vector<DataSet> &retDataSets,
    const char *aetitle = nullptr,
    const char *call = nullptr );

  static bool CStore( const char *remote, uint16_t portno,
    const Directory::FilenamesType & filenames,
    const char *aetitle = nullptr, const char *call = nullptr);
};

} // end namespace gdcm

#endif // GDCMCOMPOSITENETWORKFUNCTIONS_H
```

13.495 gdcmCStoreMessages.h File Reference

`#include "gdcmBaseCompositeMessage.h"`

Include dependency graph for gdcmCStoreMessages.h:



Classes

- class gdcm::network::CStoreRQ

  CStoreRQ.
- class gdcm::network::CStoreRSP

  CStoreRSP this file defines the messages for the cecho action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.496 gdcmCStoreMessages.h

Go to the documentation of this file.

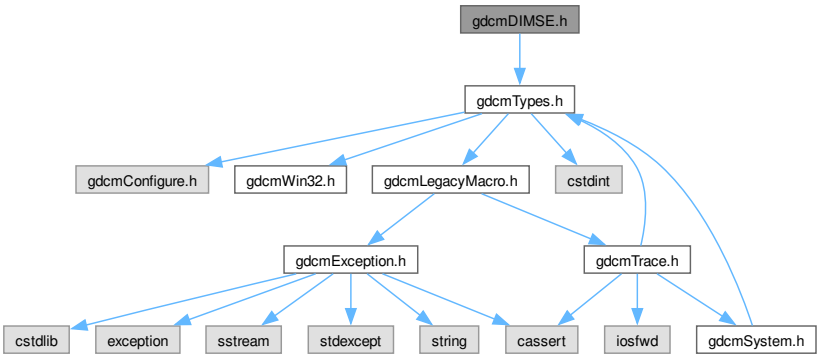
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
```

```
 *  limitations under the License.
 *
 *=============================================================================*/
#ifndef GDCMCSTOREMESSAGES_H
#define GDCMCSTOREMESSAGES_H

#include "gdcmBaseCompositeMessage.h"

namespace gdcm{
class File;
  namespace network{
    class BasePDU;
class CStoreRQ : public BaseCompositeMessage {
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection, const BaseRootQuery* inRootQuery) override;//to fulfill the virtual contract
    public:
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
        const File& file,  bool writeDataSet = true );
    };

    class CStoreRSP : public BaseCompositeMessage {
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection, const BaseRootQuery* inRootQuery) override;//to fulfill the virtual contract
    public:
      std::vector<PresentationDataValue> ConstructPDV(const DataSet* inDataSet, const BasePDU* inPC);
    };
  }
}
#endif // GDCMCSTOREMESSAGES_H
```

13.497 gdcmDIMSE.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmDIMSE.h:



Classes

- class gdcm::network::CEchoRQ

  CEchoRQ.
- class gdcm::network::CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.
- class gdcm::network::CFind
- class gdcm::network::DIMSE

  DIMSE.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.498 gdcmDIMSE.h

Go to the documentation of this file.

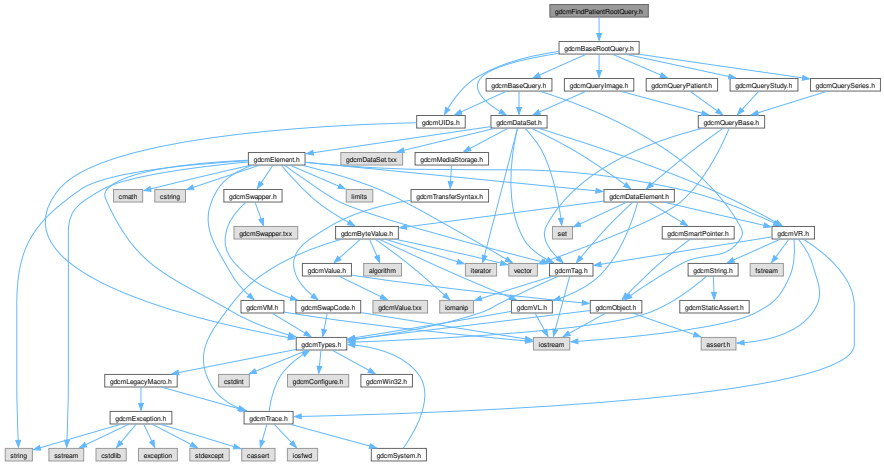
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMDIMSE_H
#define GDCMDIMSE_H

#include "gdcmTypes.h"

namespace gdcm
{

namespace network
{

class DIMSE {
public:
  typedef enum {
C_STORE_RQ         = 0x0001,
C_STORE_RSP        = 0x8001,
C_GET_RQ           = 0x0010,
C_GET_RSP          = 0x8010,
C_FIND_RQ          = 0x0020,
C_FIND_RSP         = 0x8020,
C_MOVE_RQ          = 0x0021,
C_MOVE_RSP         = 0x8021,
C_ECHO_RQ          = 0x0030,
C_ECHO_RSP         = 0x8030,
N_EVENT_REPORT_RQ  = 0x0100,
N_EVENT_REPORT_RSP = 0x8100,
N_GET_RQ           = 0x0110,
N_GET_RSP          = 0x8110,
N_SET_RQ           = 0x0120,
N_SET_RSP          = 0x8120,
N_ACTION_RQ        = 0x0130,
N_ACTION_RSP       = 0x8130,
N_CREATE_RQ        = 0x0140,
N_CREATE_RSP       = 0x8140,
N_DELETE_RQ        = 0x0150,
N_DELETE_RSP       = 0x8150,
C_CANCEL_RQ        = 0x0FFF
  } CommandTypes;
};

/*
9.1.5.1 C-ECHO parameters
Table 9.1-5
C-ECHO PARAMETERS
*/
class CEchoRQ
{
public:
  uint16_t          MessageID;                              /* M */
  UIComp            AffectedSOPClassUID;                    /* M */
};

```

```
class CEchoRSP
{
public:
/*
Message ID M U
Message ID Being Responded To M
Affected SOP Class UID M U(=)
Status M
*/
};

class CFind
{
/*
Failure Refused: Out of Resources A700 (0000,0902)
Identifier does not match SOP Class A900 (0000,0901)
(0000,0902)
Unable to process Cxxx (0000,0901)
(0000,0902)
Cancel Matching terminated due to Cancel
request
FE00 None
Success Matching is complete - No final Identifier
is supplied.
0000 None
Pending Matches are continuing - Current Match
is supplied and any Optional Keys were
supported in the same manner as
Required Keys.
FF00 Identifier
Matches are continuing - Warning that
one or more Optional Keys were not
supported for existence and/or matching
for this Identifier.
FF01 Identifier
*/
};


} // end namespace network

} // end namespace gdcm

#endif //GDCMDIMSE_H
```

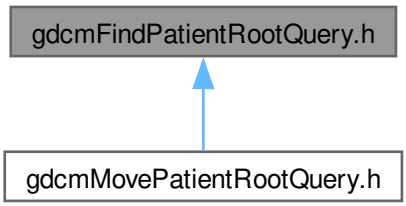
13.499 gdcmFindPatientRootQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`

Include dependency graph for gdcmFindPatientRootQuery.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::FindPatientRootQuery

  PatientRootQuery.

Namespaces

- namespace gdcm

13.500 gdcmFindPatientRootQuery.h

Go to the documentation of this file.

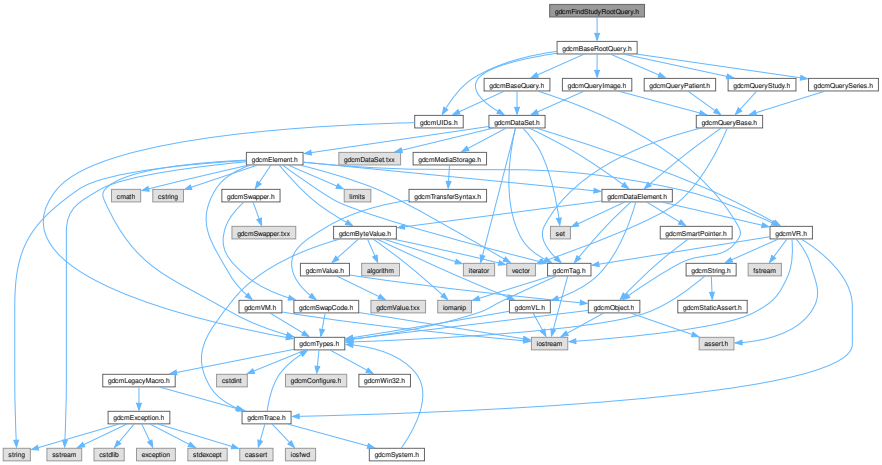
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMFINDPATIENTROOTQUERY_H
#define GDCMFINDPATIENTROOTQUERY_H

#include "gdcmBaseRootQuery.h"

namespace gdcm
{
class GDCM_EXPORT FindPatientRootQuery : public BaseRootQuery
{
  friend class QueryFactory;
public:
  FindPatientRootQuery();

  void InitializeDataSet(const EQueryLevel& inQueryLevel) override;

  std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
  bool ValidateQuery(bool inStrict = true) const override;
```

```
00035
00036   UIDs::TSName GetAbstractSyntaxUID() const override;
00037 };
00038
00039 } // end namespace gdcm
00040
00041 #endif // GDCMFINDPATIENTROOTQUERY_H
```

13.501 gdcmFindStudyRootQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`

Include dependency graph for gdcmFindStudyRootQuery.h:



Classes

- class gdcm::FindStudyRootQuery

  FindStudyRootQuery.

Namespaces

- namespace gdcm

13.502 gdcmFindStudyRootQuery.h

Go to the documentation of this file.

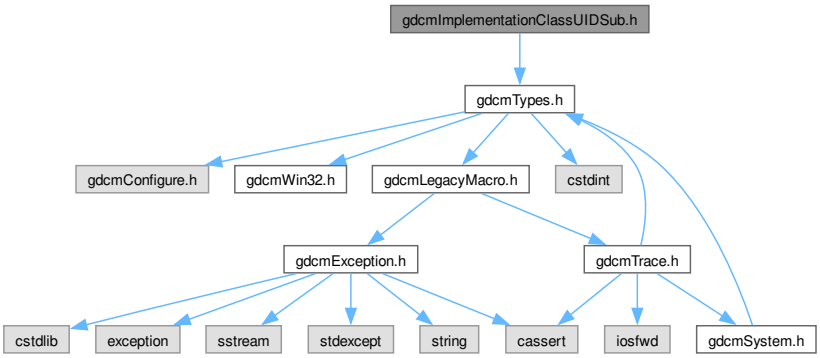
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMFINDSTUDYROOTQUERY_H
00015 #define GDCMFINDSTUDYROOTQUERY_H
00016
00017 #include "gdcmBaseRootQuery.h"
00018
00019 namespace gdcm
00020 {
00025 class GDCM_EXPORT FindStudyRootQuery : public BaseRootQuery
00026 {
00027   friend class QueryFactory;
00028 public:
00029   FindStudyRootQuery();
00030
00031   void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
00032
00033   std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
00034
00038   bool ValidateQuery(bool inStrict = true) const override;
00039
00040   UIDs::TSName GetAbstractSyntaxUID() const override;
00041 };
00042
00043 } // end namespace gdcm
00044
00045 #endif // GDCMFINDSTUDYROOTQUERY_H
```

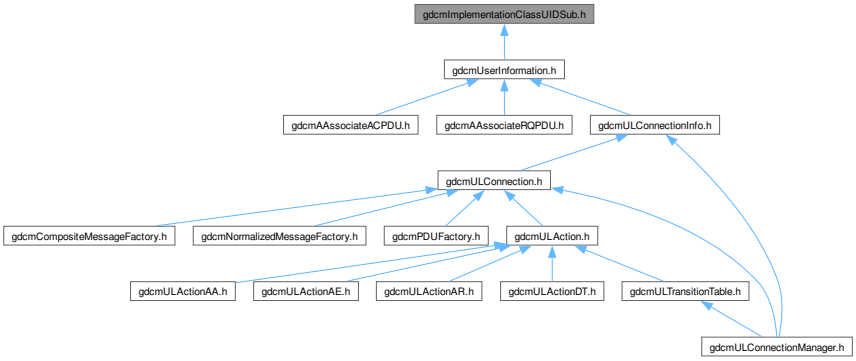
13.503 gdcmImplementationClassUIDSub.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmImplementationClassUIDSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ImplementationClassUIDSub

  ImplementationClassUIDSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.504 gdcmImplementationClassUIDSub.h

Go to the documentation of this file.

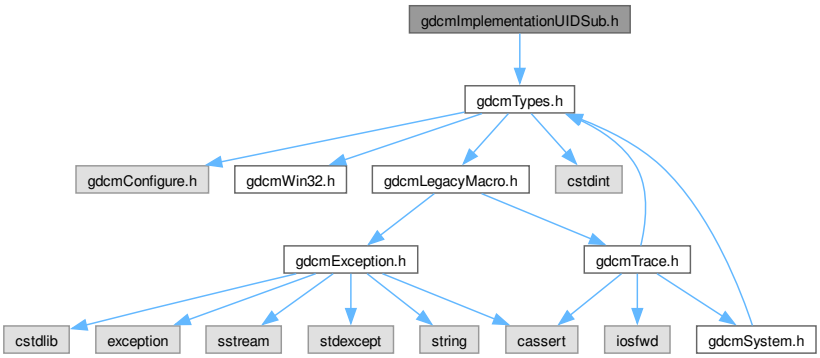
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMPLEMENTATIONCLASSUIDSUB_H
#define GDCMIMPLEMENTATIONCLASSUIDSUB_H

#include "gdcmTypes.h"

namespace gdcm
{

namespace network
{

class ImplementationClassUIDSub
{
public:
```

```
  ImplementationClassUIDSub();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;

  void Print(std::ostream &os) const;

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength;
  std::string ImplementationClassUID;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMMAXIMUMLENGTHSUB_H
```

13.505 gdcmImplementationUIDSub.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmImplementationUIDSub.h:



Classes

- class gdcm::network::ImplementationUIDSub

  ImplementationUIDSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.506 gdcmImplementationUIDSub.h

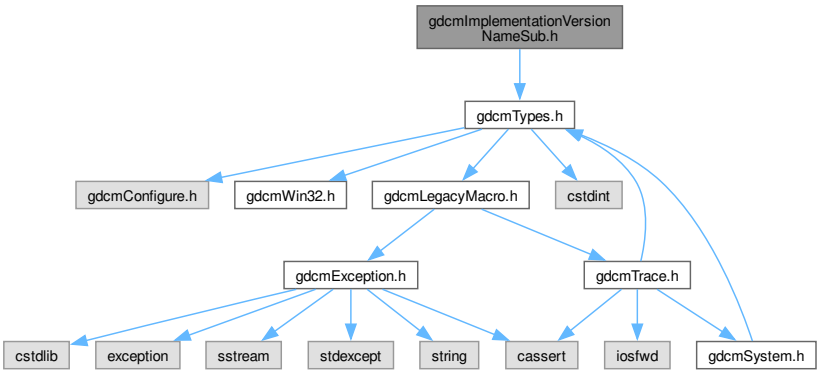
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMPLEMENTATIONUIDSUB_H
#define GDCMIMPLEMENTATIONUIDSUB_H

#include "gdcmTypes.h"

namespace gdcm
{

namespace network
{

class GDCM_EXPORT ImplementationUIDSub
{
public:
  ImplementationUIDSub();
  const std::ostream &Write(std::ostream &os) const;
private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength;
  std::string ImplementationClassUID;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMMAXIMUMLENGTHSUB_H
```

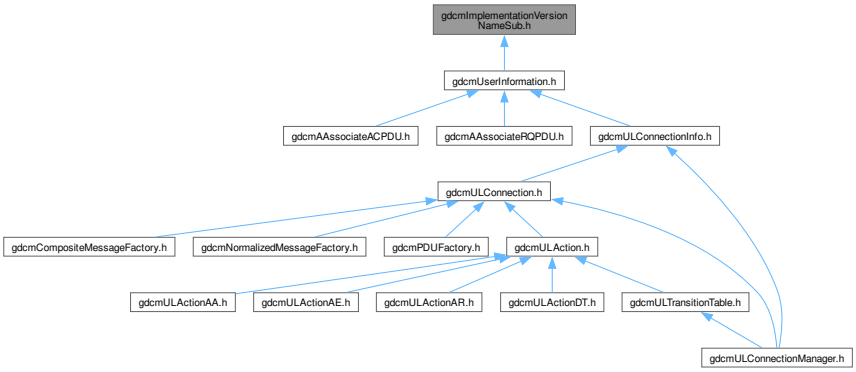
13.507 gdcmImplementationVersionNameSub.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmImplementationVersionNameSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ImplementationVersionNameSub

  ImplementationVersionNameSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.508 gdcmImplementationVersionNameSub.h

Go to the documentation of this file.

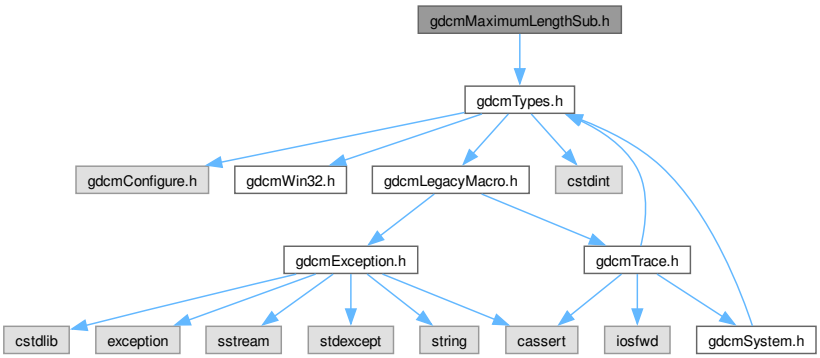
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMIMPLEMENTATIONVERSIONNAMESUB_H
#define GDCMIMPLEMENTATIONVERSIONNAMESUB_H

#include "gdcmTypes.h"

namespace gdcm
{

namespace network
{

class ImplementationVersionNameSub
{
```

```
public:
  ImplementationVersionNameSub();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;
  void Print(std::ostream &os) const;

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength;
  std::string ImplementationVersionName;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMMAXIMUMLENGTHSUB_H
```

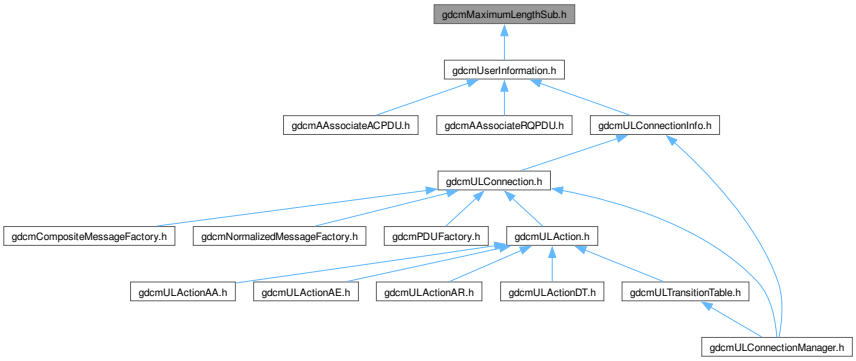
13.509 gdcmMaximumLengthSub.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmMaximumLengthSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::MaximumLengthSub

  MaximumLengthSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.510 gdcmMaximumLengthSub.h

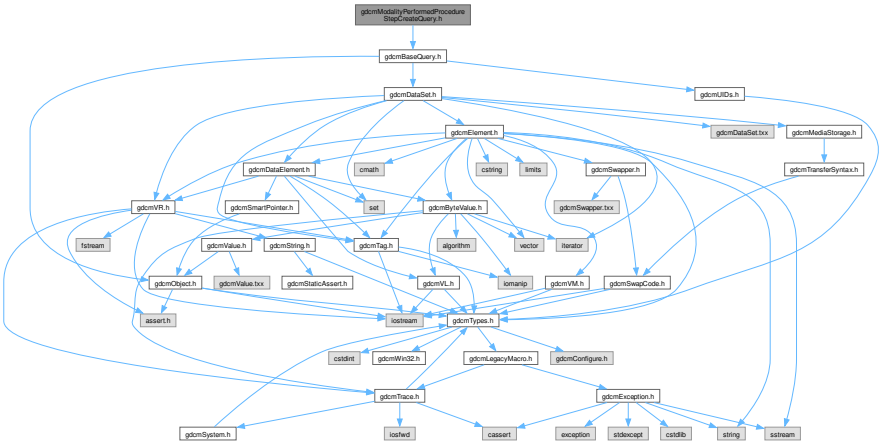
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMAXIMUMLENGTHSUB_H
#define GDCMMAXIMUMLENGTHSUB_H

#include "gdcmTypes.h"

namespace gdcm
{

namespace network
{

class MaximumLengthSub
{
public:
  MaximumLengthSub();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;

  uint32_t GetMaximumLength() const { return MaximumLength; }
  void SetMaximumLength(uint32_t maximumlength);

  void Print(std::ostream &os) const;

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength;
  uint32_t MaximumLength;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMMAXIMUMLENGTHSUB_H
```

13.511 gdcmModalityPerformedProcedureStepCreateQuery.h File Reference

`#include "gdcmBaseQuery.h"`

Include dependency graph for gdcmModalityPerformedProcedureStepCreateQuery.h:



Classes

- class gdcm::ModalityPerformedProcedureStepCreateQuery

  ModalityPerformedProcedureStepCreateQuery.

Namespaces

- namespace gdcm

13.512 gdcmModalityPerformedProcedureStepCreateQuery.h

Go to the documentation of this file.

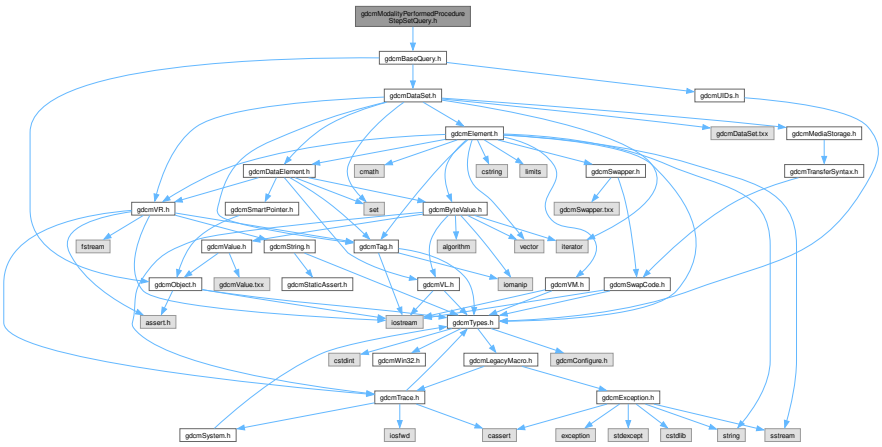
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2014 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMMODALITYPERFORMEDPROCEDURESTEPCREATEQUERY_H
00015 #define GDCMMODALITYPERFORMEDPROCEDURESTEPCREATEQUERY_H
00016
00017 #include "gdcmBaseQuery.h"
00018
```

```
namespace gdcm
{
class GDCM_EXPORT ModalityPerformedProcedureStepCreateQuery : public BaseQuery{
  friend class QueryFactory;
public:
  ModalityPerformedProcedureStepCreateQuery( const std::string & iSopInstanceUID );

  gdcm::DataSet GetRequiredDataSet() const;
  bool ValidateQuery(bool inStrict = true) const override;
  UIDs::TSName GetAbstractSyntaxUID() const override;
};

} // end namespace gdcm

#endif // GDCMMODALITYPERFORMEDPROCEDURESTEPCREATEQUERY_H
```

13.513 gdcmModalityPerformedProcedureStepSetQuery.h File Reference

`#include "gdcmBaseQuery.h"`

Include dependency graph for gdcmModalityPerformedProcedureStepSetQuery.h:



Classes

- class gdcm::ModalityPerformedProcedureStepSetQuery

  ModalityPerformedProcedureStepSetQuery.

Namespaces

- namespace gdcm

13.514 gdcmModalityPerformedProcedureStepSetQuery.h

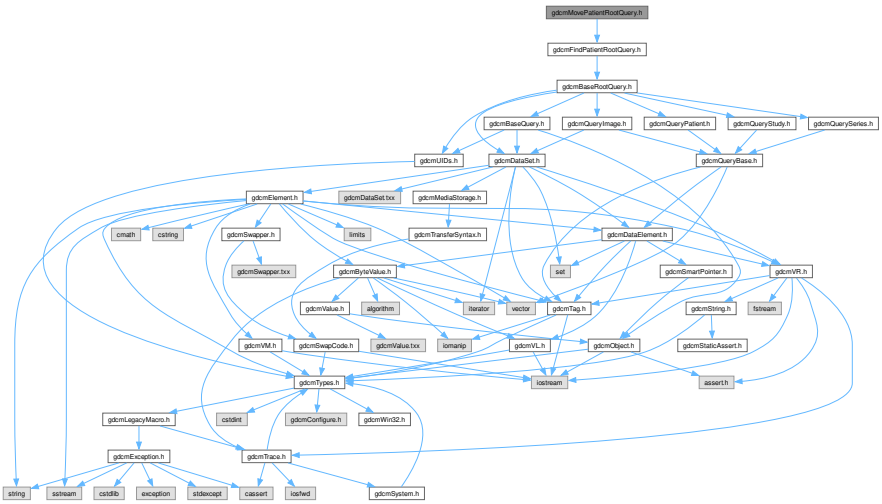
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
#define GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H

#include "gdcmBaseQuery.h"

namespace gdcm
{
class GDCM_EXPORT ModalityPerformedProcedureStepSetQuery : public BaseQuery{
  friend class QueryFactory;
public:
  ModalityPerformedProcedureStepSetQuery( const std::string & iSopInstanceUID );

  gdcm::DataSet GetRequiredDataSet() const;
  bool ValidateQuery(bool inStrict = true) const override;
  UIDs::TSName GetAbstractSyntaxUID() const override;
};

} // end namespace gdcm

#endif // GDCMMODALITYPERFORMEDPROCEDURESTEPSETQUERY_H
```

13.515 gdcmMovePatientRootQuery.h File Reference

`#include "gdcmFindPatientRootQuery.h"`

Include dependency graph for gdcmMovePatientRootQuery.h:



Classes

- class gdcm::MovePatientRootQuery

  MovePatientRootQuery.

Namespaces

- namespace gdcm

13.516 gdcmMovePatientRootQuery.h

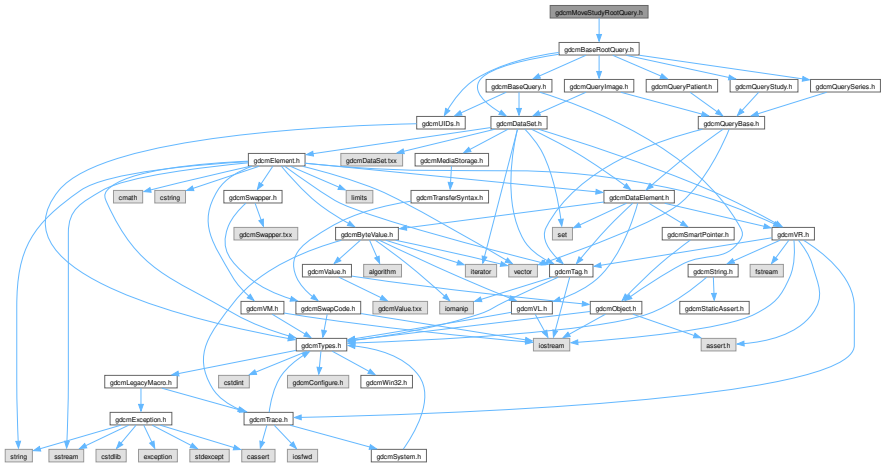
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMOVEPATIENTROOTQUERY_H
#define GDCMMOVEPATIENTROOTQUERY_H

#include "gdcmFindPatientRootQuery.h"

namespace gdcm
{
class GDCM_EXPORT MovePatientRootQuery : public BaseRootQuery
{
  friend class QueryFactory;
public:
  MovePatientRootQuery();

  void InitializeDataSet(const EQueryLevel& inQueryLevel) override;

  std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;

  bool ValidateQuery(bool inStrict = true) const override;

  UIDs::TSName GetAbstractSyntaxUID() const override;
};

} // end namespace gdcm

#endif // GDCMMOVEPATIENTROOTQUERY_H
```

13.517 gdcmMoveStudyRootQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`

Include dependency graph for gdcmMoveStudyRootQuery.h:



Classes

- class gdcm::MoveStudyRootQuery

  MoveStudyRootQuery.

Namespaces

- namespace gdcm

13.518 gdcmMoveStudyRootQuery.h

Go to the documentation of this file.

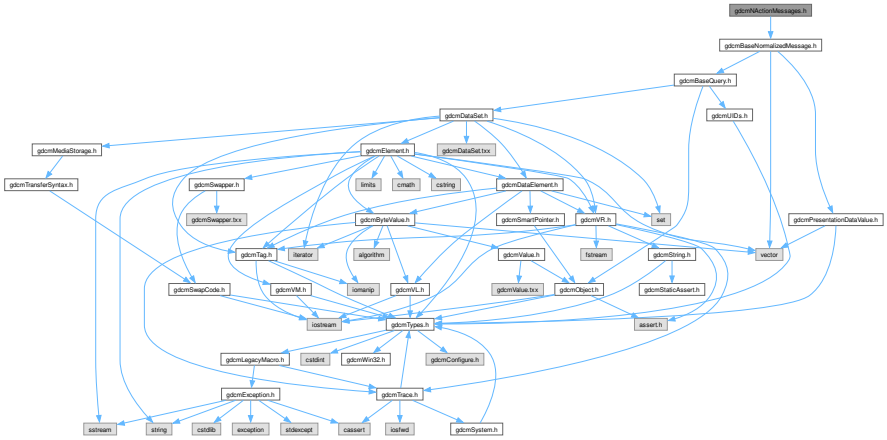
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMMOVESTUDYROOTQUERY_H
#define GDCMMOVESTUDYROOTQUERY_H

#include "gdcmBaseRootQuery.h"

namespace gdcm
```

```
{
class GDCM_EXPORT MoveStudyRootQuery : public BaseRootQuery
{
  friend class QueryFactory;
public:
  MoveStudyRootQuery();

  void InitializeDataSet(const EQueryLevel& inQueryLevel) override;

  std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;

  bool ValidateQuery(bool inStrict = true) const override;

  UIDs::TSName GetAbstractSyntaxUID() const override;
};

} // end namespace gdcm

#endif // GDCMMOVESTUDYROOTQUERY_H
```

13.519 gdcmNActionMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`

Include dependency graph for gdcmNActionMessages.h:



Classes

- class gdcm::network::NActionRQ

  NActionRQ.
- class gdcm::network::NActionRSP

  NActionRSP this file defines the messages for the NAction action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.520 gdcmNActionMessages.h

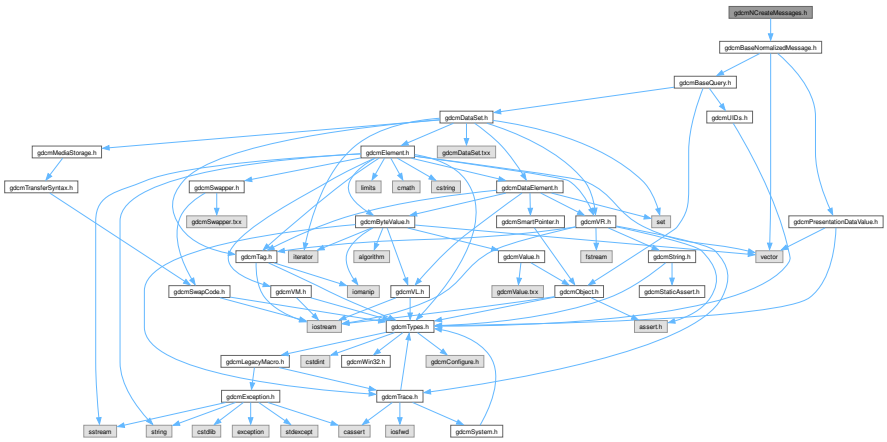
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2014 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCNACTIONMESSAGES_H
#define GDCMCNACTIONMESSAGES_H

#include "gdcmBaseNormalizedMessage.h"

namespace gdcm{
  namespace network{

class ULConnection;

class NActionRQ : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
        const BaseQuery* inQuery) override;
    };

    class NActionRSP : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    };
  }
}
#endif // GDCMCNACTIONMESSAGES_H
```

13.521 gdcmNCreateMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`

Include dependency graph for gdcmNCreateMessages.h:



Classes

- class gdcm::network::NCreateRQ

  NCreateRQ.
- class gdcm::network::NCreateRSP

  NCreateRSP this file defines the messages for the ncreate action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.522 gdcmNCreateMessages.h

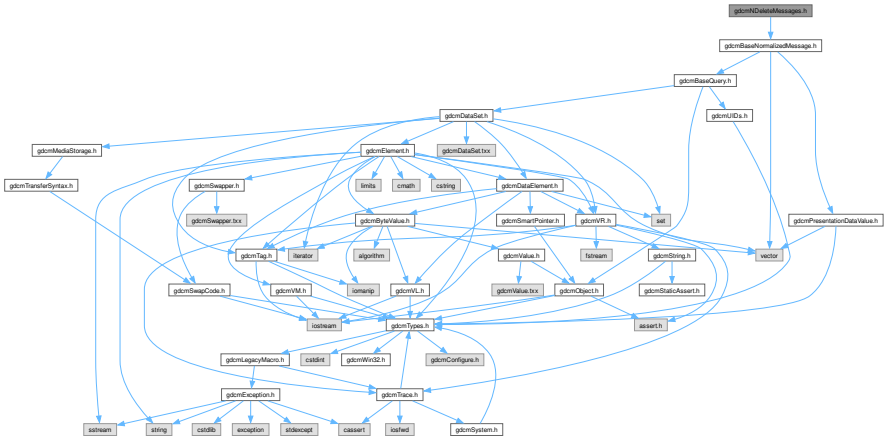
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2014 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCNCREATEMESSAGES_H
#define GDCMCNCREATEMESSAGES_H

#include "gdcmBaseNormalizedMessage.h"

namespace gdcm{
  namespace network{

class ULConnection;

class NCreateRQ : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
        const BaseQuery* inQuery) override;
    };

    class NCreateRSP : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    };
  }
}
#endif // GDCMCNCREATEMESSAGES_H
```

13.523 gdcmNDeleteMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`

Include dependency graph for gdcmNDeleteMessages.h:



Classes

- class gdcm::network::NDeleteRQ

  NDeleteRQ.
- class gdcm::network::NDeleteRSP

  NDeleteRSP this file defines the messages for the ndelete action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.524 gdcmNDeleteMessages.h

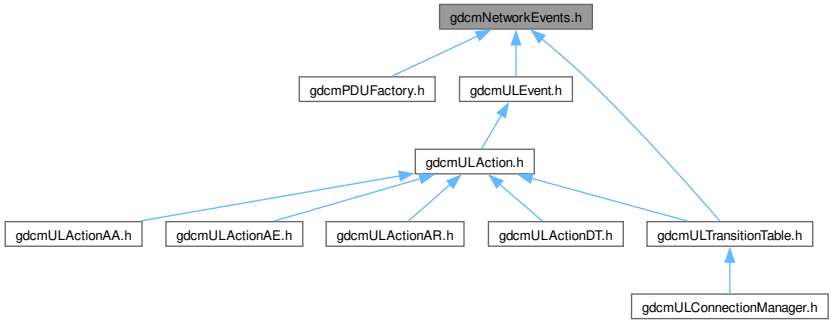
Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2014 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMCNDELETEMESSAGES_H
00015 #define GDCMCNDELETEMESSAGES_H
```

```
#include "gdcmBaseNormalizedMessage.h"

namespace gdcm{
  namespace network{

class ULConnection;

class NDeleteRQ : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
        const BaseQuery* inQuery) override;
    };

    class NDeleteRSP : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    };
  }
}
#endif // GDCMCNDELETEMESSAGES_H
```

13.525 gdcmNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace gdcm
- namespace gdcm::network

Enumerations

- enum gdcm::network::EEventID {
  gdcm::network::eAASSOCIATERequestLocalUser = 0 ,
  gdcm::network::eTransportConnConfirmLocal ,
  gdcm::network::eASSOCIATE_ACPDUreceived ,
  gdcm::network::eASSOCIATE_RJPDUreceived ,
  gdcm::network::eTransportConnIndicLocal ,
  gdcm::network::eAASSOCIATE_RQPDUreceived ,

gdcm::network::eAASSOCIATEresponseAccept ,
gdcm::network::eAASSOCIATEresponseReject ,
gdcm::network::ePDATArequest ,
gdcm::network::ePDATATFPDU ,
gdcm::network::eARELEASERequest ,
gdcm::network::eARELEASE_RQPDUReceivedOpen ,
gdcm::network::eARELEASE_RPPDUReceived ,
gdcm::network::eARELEASEResponse ,
gdcm::network::eAABORTRequest ,
gdcm::network::eAABORTPDUReceivedOpen ,
gdcm::network::eTransportConnectionClosed ,
gdcm::network::eARTIMTimerExpired ,
gdcm::network::eUnrecognizedPDUReceived ,
gdcm::network::eEventDoesNotExist }

Variables

- const int gdcm::network::cMaxEventID = eEventDoesNotExist

13.526 gdcmNetworkEvents.h

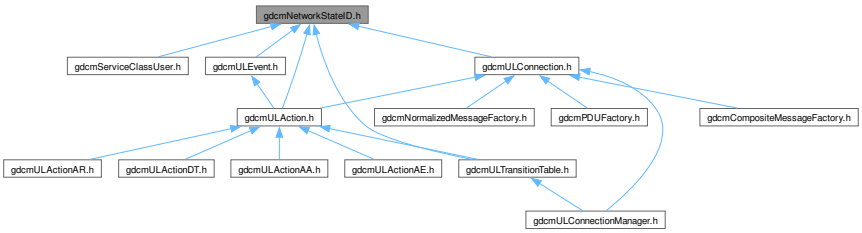
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
/*
The NetworkEvents enumeration defines the inputs into the state of the network connection.

These inputs can come either from user input or input from other things on the socket,
ie, responses from the peer or ARTIM timeouts.

Note that this enumeration is not 'power of two', like the states, because you can't have
multiple simultaneous events.  Multiple state outputs in transition tables, however, is possible.

*/
#ifndef GDCMNETWORKEVENTS_H
#define GDCMNETWORKEVENTS_H

namespace gdcm {
  namespace network{
    typedef enum {
      eAASSOCIATERequestLocalUser = 0,
      eTransportConnConfirmLocal,
      eASSOCIATE_ACPDUreceived,
      eASSOCIATE_RJPDUreceived,
      eTransportConnIndicLocal,
      eAASSOCIATE_RQPDUreceived,
      eAASSOCIATEresponseAccept,
      eAASSOCIATEresponseReject,
      ePDATArequest,
```

```
      ePDATATFPDU,
      eARELEASERequest,
      eARELEASE_RQPDUReceivedOpen,
      eARELEASE_RPPDUReceived,
      eARELEASEResponse,
      eAABORTRequest,
      eAABORTPDUReceivedOpen,
      eTransportConnectionClosed,
      eARTIMTimerExpired,
      eUnrecognizedPDUReceived,
      eEventDoesNotExist
    } EEventID;

    const int cMaxEventID = eEventDoesNotExist;
  }
}

#endif //NETWORKEVENTS_H
```

13.527 gdcmNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace gdcm
- namespace gdcm::network

Enumerations

- enum gdcm::network::EStateID {
  gdcm::network::eStaDoesNotExist = 0 ,
  gdcm::network::eSta1Idle = 1 ,
  gdcm::network::eSta2Open = 2 ,
  gdcm::network::eSta3WaitLocalAssoc = 4 ,
  gdcm::network::eSta4LocalAssocDone = 8 ,
  gdcm::network::eSta5WaitRemoteAssoc = 16 ,
  gdcm::network::eSta6TransferReady = 32 ,
  gdcm::network::eSta7WaitRelease = 64 ,
  gdcm::network::eSta8WaitLocalRelease = 128 ,
  gdcm::network::eSta9ReleaseCollisionRqLocal = 256 ,
  gdcm::network::eSta10ReleaseCollisionAc = 512 ,
  gdcm::network::eSta11ReleaseCollisionRq = 1024 ,
  gdcm::network::eSta12ReleaseCollisionAcLocal = 2048 ,
  gdcm::network::eSta13AwaitingClose = 4096 }

Functions

- int gdcm::network::GetStateIndex (EStateID inState)

Variables

- const int gdcm::network::cMaxStateID = 13

13.528 gdcmNetworkStateID.h

Go to the documentation of this file.

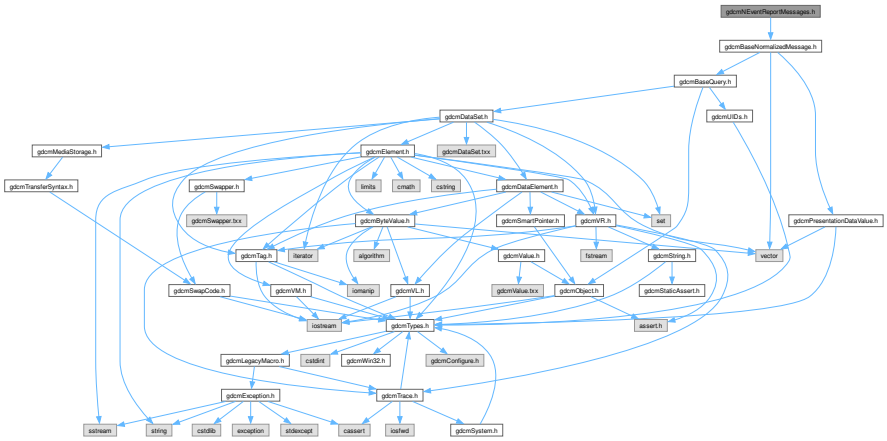
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMNETWORKSTATEID_H
#define GDCMNETWORKSTATEID_H

namespace gdcm {
  namespace network {

    enum EStateID {
      eStaDoesNotExist = 0,
      eSta1Idle = 1,
      eSta2Open = 2,
      eSta3WaitLocalAssoc = 4,
      eSta4LocalAssocDone = 8,
      eSta5WaitRemoteAssoc = 16,
      eSta6TransferReady = 32,
      eSta7WaitRelease = 64,
      eSta8WaitLocalRelease = 128,
      eSta9ReleaseCollisionRqLocal = 256,
      eSta10ReleaseCollisionAc = 512,
      eSta11ReleaseCollisionRq = 1024,
      eSta12ReleaseCollisionAcLocal = 2048,
      eSta13AwaitingClose = 4096
    };

    const int cMaxStateID = 13;

    //the transition table is built on state indices
    //this function will produce the index from the power-of-two EStateID
    inline int GetStateIndex(EStateID inState){
      switch (inState){
        case eStaDoesNotExist:
        default:
          return -1;
        case eSta1Idle:
          return 0;
        case eSta2Open:
          return 1;
        case eSta3WaitLocalAssoc:
          return 2;
        case eSta4LocalAssocDone:
          return 3;
        case eSta5WaitRemoteAssoc:
          return 4;
```

```
        case eSta6TransferReady:
          return 5;
        case eSta7WaitRelease:
          return 6;
        case eSta8WaitLocalRelease:
          return 7;
        case eSta9ReleaseCollisionRqLocal:
          return 8;
        case eSta10ReleaseCollisionAc:
          return 9;
        case eSta11ReleaseCollisionRq:
          return 10;
        case eSta12ReleaseCollisionAcLocal:
          return 11;
        case eSta13AwaitingClose:
          return 12;
      }
    }
  }
}

#endif //GDCMNETWORKSTATEID_H
```

13.529 gdcmNEventReportMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`

Include dependency graph for gdcmNEventReportMessages.h:



Classes

- class gdcm::network::NEventReportRQ

  NEventReportRQ.
- class gdcm::network::NEventReportRSP

  NEventReportRSP this file defines the messages for the neventreport action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.530 gdcmNEventReportMessages.h

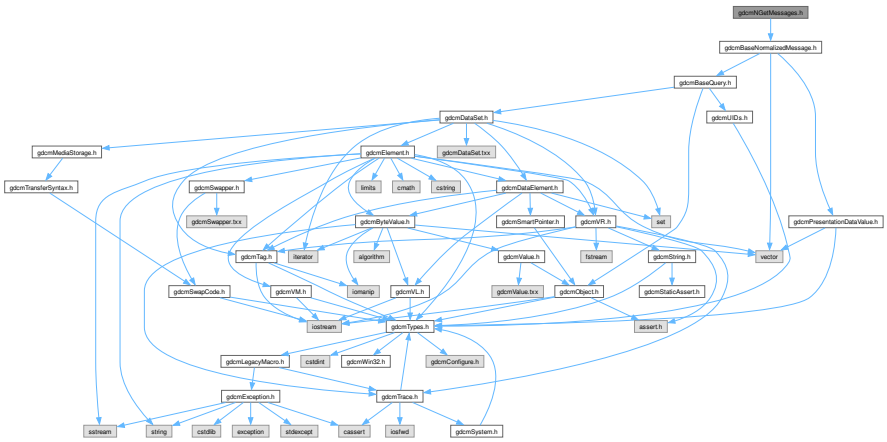
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2014 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCNEVENTREPORTMESSAGES_H
#define GDCMCNEVENTREPORTMESSAGES_H

#include "gdcmBaseNormalizedMessage.h"

namespace gdcm{
  namespace network{

class ULConnection;

class NEventReportRQ : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
        const BaseQuery* inQuery) override;
    };

    class NEventReportRSP : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    };
  }
}
#endif // GDCMCNEVENTREPORTMESSAGES_H
```

13.531 gdcmNGetMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`

Include dependency graph for gdcmNGetMessages.h:



Classes

- class gdcm::network::NGetRQ

  NGetRQ.
- class gdcm::network::NGetRSP

  NGetRSP this file defines the messages for the nget action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.532 gdcmNGetMessages.h

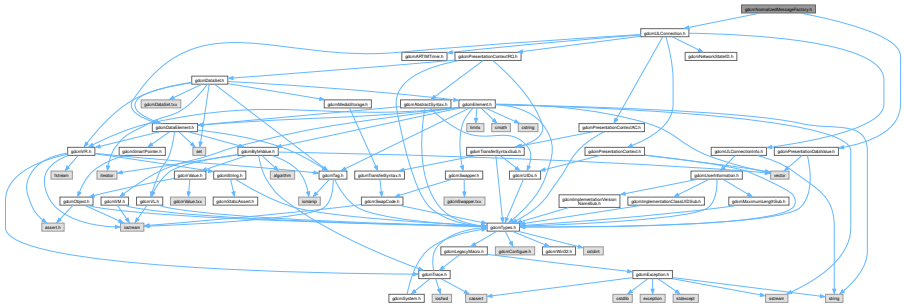
Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2014 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMCNGETMESSAGES_H
00015 #define GDCMCNGETMESSAGES_H
00016
00017 #include "gdcmBaseNormalizedMessage.h"
00018
00019 namespace gdcm{
00020   namespace network{
00021
00022 class ULConnection;
00023
00028 class NGetRQ : public BaseNormalizedMessage {
00029     public:
00030       std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
00031         const BaseQuery* inQuery) override;
00032     };
00033
00038     class NGetRSP : public BaseNormalizedMessage {
00039     public:
00040       std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
00041     };
00042   }
00043 }
00044 #endif // GDCMCNGETMESSAGES_H
```

13.533 gdcmNormalizedMessageFactory.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmNormalizedMessageFactory.h:



Classes

- class gdcm::network::NormalizedMessageFactory

Namespaces

- namespace gdcm
- namespace gdcm::network

13.534 gdcmNormalizedMessageFactory.h

Go to the documentation of this file.

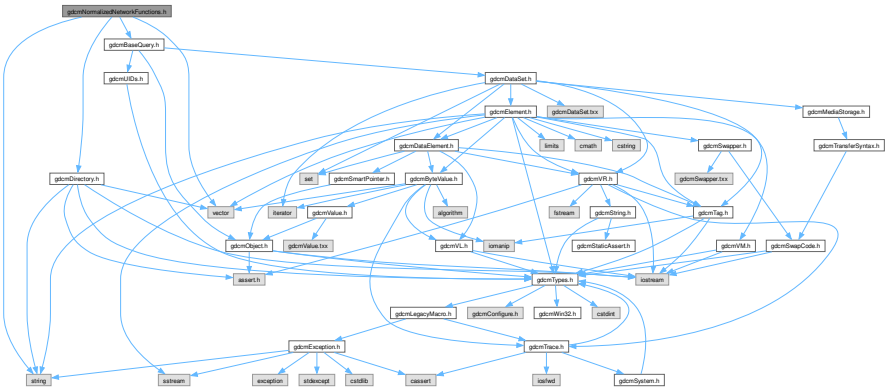
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2014 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMNORMALIZEDMESSAGEFACTORY_H
#define GDCMNORMALIZEDMESSAGEFACTORY_H

#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"

namespace gdcm {
  class BaseQuery;
  class File;
  namespace network {
    class BasePDU;

class NormalizedMessageFactory
{
    public:
      static  std::vector<PresentationDataValue> ConstructNEventReport  (const ULConnection& inConnection, const BaseQuery* inQuery);
      static  std::vector<PresentationDataValue> ConstructNGet      (const ULConnection& inConnection, const BaseQuery* inQuery);
      static  std::vector<PresentationDataValue> ConstructNSet      (const ULConnection& inConnection, const BaseQuery* inQuery);
```

```
      static  std::vector<PresentationDataValue> ConstructNAction   (const ULConnection& inConnection,
    const BaseQuery* inQuery);
      static  std::vector<PresentationDataValue> ConstructNCreate   (const ULConnection& inConnection,
    const BaseQuery* inQuery);
      static  std::vector<PresentationDataValue> ConstructNDelete   (const ULConnection& inConnection,
    const BaseQuery* inQuery);


    };
  }
}

#endif // GDCMNORMALIZEDMESSAGEFACTORY_H
```

13.535 gdcmNormalizedNetworkFunctions.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmBaseQuery.h"
#include <vector>
#include <string>
```

Include dependency graph for gdcmNormalizedNetworkFunctions.h:



Classes

- class gdcm::NormalizedNetworkFunctions

  Normalized Network Functions.

Namespaces

- namespace gdcm

13.536 gdcmNormalizedNetworkFunctions.h

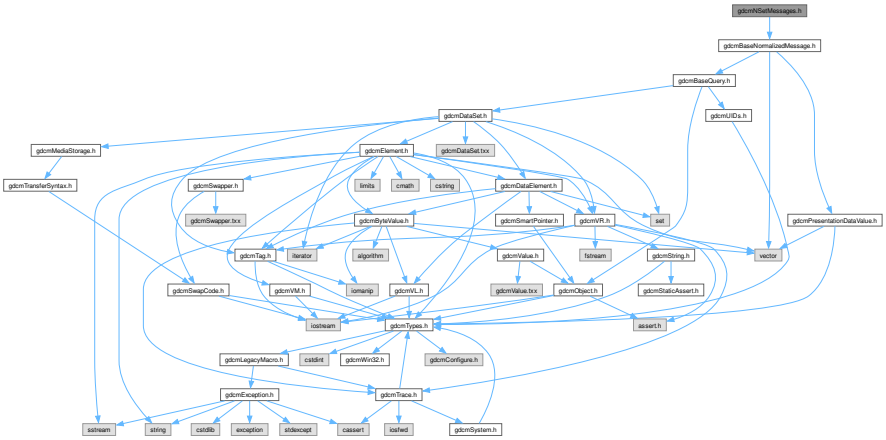
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2014 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMNORMALIZEDNETWORKFUNCTIONS_H
#define GDCMNORMALIZEDNETWORKFUNCTIONS_H

#include "gdcmDirectory.h"
#include "gdcmBaseQuery.h" // EQueryLevel / EQueryType

#include <vector>
#include <string>

namespace gdcm
{
class GDCM_EXPORT NormalizedNetworkFunctions
{
public:
  static BaseQuery* ConstructQuery( const std::string & sopInstanceUID,
                    const DataSet& queryds, ENQueryType queryType = eCreateMMPS );
  static bool NEventReport( const char *remote, uint16_t portno,
                const BaseQuery* query, std::vector<DataSet> &retDataSets,
                const char *aetitle, const char *call );
  static bool NGet( const char *remote, uint16_t portno,
                      const BaseQuery* query, std::vector<DataSet> &retDataSets,
                      const char *aetitle, const char *call );
  static bool NSet( const char *remote, uint16_t portno,
                      const BaseQuery* query, std::vector<DataSet> &retDataSets,
                      const char *aetitle, const char *call );
  static bool NAction( const char *remote, uint16_t portno,
                      const BaseQuery* query, std::vector<DataSet> &retDataSets,
                      const char *aetitle, const char *call );
  static bool NCreate( const char *remote, uint16_t portno,
                      BaseQuery* query, std::vector<DataSet> &retDataSets,
                      const char *aetitle, const char *call );
  static bool NDelete( const char *remote, uint16_t portno,
                      const BaseQuery* query, std::vector<DataSet> &retDataSets,
                      const char *aetitle, const char *call );
};

} // end namespace gdcm

#endif // GDCMCOMPOSITENETWORKFUNCTIONS_H
```

13.537 gdcmNSetMessages.h File Reference

`#include "gdcmBaseNormalizedMessage.h"`

Include dependency graph for gdcmNSetMessages.h:



Classes

- class gdcm::network::NSetRQ

  NSetRQ.
- class gdcm::network::NSetRSP

  NSetRSP this file defines the messages for the nset action.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.538 gdcmNSetMessages.h

Go to the documentation of this file.

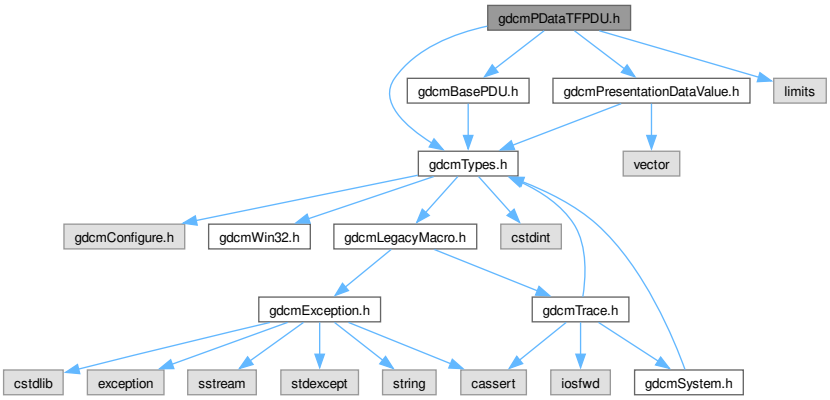
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2014 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMCNSETMESSAGES_H
#define GDCMCNSETMESSAGES_H
```

```

#include "gdcmBaseNormalizedMessage.h"

namespace gdcm{
  namespace network{

class ULConnection;

class NSetRQ : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDV(const ULConnection &inConnection,
        const BaseQuery* inQuery) override;
    };

    class NSetRSP : public BaseNormalizedMessage {
    public:
      std::vector<PresentationDataValue> ConstructPDVByDataSet(const DataSet* inDataSet);
    };
  }
}
#endif // GDCMCNSETMESSAGES_H
```

13.539 gdcmPDataTFPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmBasePDU.h"
#include <limits>
```

Include dependency graph for gdcmPDataTFPDU.h:



Classes

- class gdcm::network::PDataTFPDU

  PDataTFPDU.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.540 gdcmPDataTFPDU.h

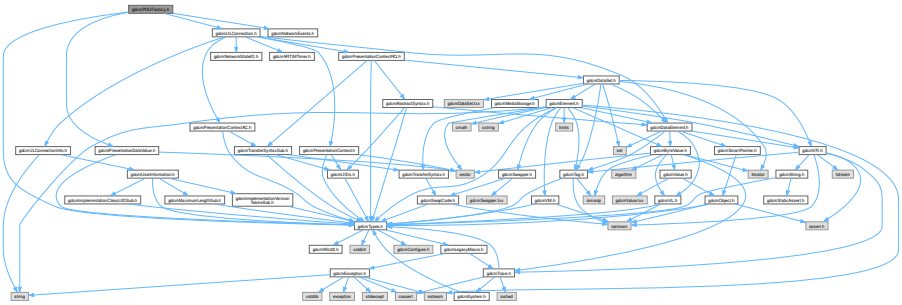
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPDATATFPDU_H
#define GDCMPDATATFPDU_H

#include "gdcmTypes.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmBasePDU.h"
#include <limits>

namespace gdcm
{

namespace network
{

class GDCM_EXPORT PDataTFPDU : public BasePDU
{
public:
  PDataTFPDU();
  std::istream &Read(std::istream &is) override;
  const std::ostream &Write(std::ostream &os) const override;

  size_t Size() const override;

  void AddPresentationDataValue( PresentationDataValue const &pdv ) {
    V.push_back( pdv );
    assert(Size() < std::numeric_limits<uint32_t>::max());
    ItemLength = (uint32_t)Size() - 6;
    }

  typedef std::vector<PresentationDataValue>::size_type SizeType;
  PresentationDataValue const &GetPresentationDataValue(SizeType i) const {
    assert( !V.empty() && i < V.size() );
    return V[i];
  }
  SizeType GetNumberOfPresentationDataValues() const {
    return V.size();
    }

  void Print(std::ostream &os) const override;
  bool IsLastFragment() const override;

protected:
  std::istream &ReadInto(std::istream &is, std::ostream &os);
private:
  static const uint8_t ItemType; // PDUType ?
  static const uint8_t Reserved2;
  uint32_t ItemLength; // PDU Length ?
  std::vector<PresentationDataValue> V;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMPDATATFPDU_H
```

13.541 gdcmPDUFactory.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULConnection.h"
#include "gdcmPresentationDataValue.h"
```

Include dependency graph for gdcmPDUFactory.h:



Classes

- class gdcm::network::PDUFactory

  PDUFactory basically, given an initial byte, construct the.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.542 gdcmPDUFactory.h

Go to the documentation of this file.

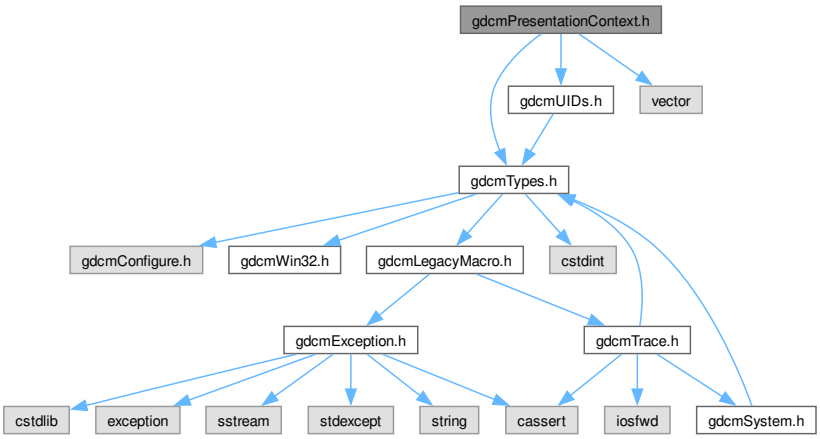
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMPDUFACTORY_H
#define GDCMPDUFACTORY_H

#include "gdcmTypes.h"
#include "gdcmNetworkEvents.h"
```

```
#include "gdcmULConnection.h"
#include "gdcmPresentationDataValue.h"

namespace gdcm{
  class BaseRootQuery;
  class BaseQuery;
  class File;
  namespace network{
    class BasePDU;

  class PDUFactory {
      public:
      static BasePDU* ConstructPDU(uint8_t itemtype);//eventually needs to be smartpointer'd
      static EEventID DetermineEventByPDU(const BasePDU* inPDU);
      static BasePDU* ConstructReleasePDU();
      static BasePDU* ConstructAbortPDU();

      //these are the composite PDU construction methods for the PDataPDUs.
      //basically, builds a pdatapdu, and then puts the appropriate information in
      //for the appropriate composite service (c-echo, c-find, c-store, c-get, c-move)
      //the connection is necessary to construct the stream of PDVs that will
      //be then placed into the vector of PDUs
      static std::vector<BasePDU*> CreateCEchoPDU(const ULConnection& inConnection);
      static std::vector<BasePDU*> CreateCStoreRQPDU(const ULConnection& inConnection, const File &file, bool writeDataSet = true );
      static std::vector<BasePDU*> CreateCStoreRSPPDU(const DataSet *inDataSet, const BasePDU* inPC);
      static std::vector<BasePDU*> CreateCFindPDU(const ULConnection& inConnection, const BaseRootQuery* inRootQuery);
      static std::vector<BasePDU*> CreateCMovePDU(const ULConnection& inConnection, const BaseRootQuery* inRootQuery);

    static std::vector<BasePDU*> CreateNEventReportPDU  (const ULConnection& inConnection, const BaseQuery *inQuery);
    static std::vector<BasePDU*> CreateNGetPDU      (const ULConnection& inConnection, const BaseQuery *inQuery);
    static std::vector<BasePDU*> CreateNSetPDU      (const ULConnection& inConnection, const BaseQuery *inQuery);
    static std::vector<BasePDU*> CreateNActionPDU     (const ULConnection& inConnection, const BaseQuery *inQuery);
    static std::vector<BasePDU*> CreateNCreatePDU     (const ULConnection& inConnection, const BaseQuery *inQuery);
    static std::vector<BasePDU*> CreateNDeletePDU     (const ULConnection& inConnection, const BaseQuery *inQuery);


      //given data pdus, produce the presentation data values stored within.
      //all operations have these as the payload of the data sending operation
      //however, echo does not have a dataset in the pdv.
      static std::vector<PresentationDataValue> GetPDVs(const std::vector<BasePDU*> & inDataPDUs);
    };
  }
}
#endif //GDCMPDUFACTORY_H
```

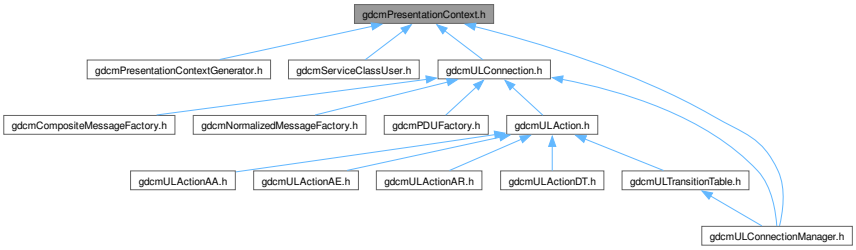
13.543 gdcmPresentationContext.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include <vector>
```

Include dependency graph for gdcmPresentationContext.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::PresentationContext

  PresentationContext.

Namespaces

- namespace gdcm

13.544 gdcmPresentationContext.h

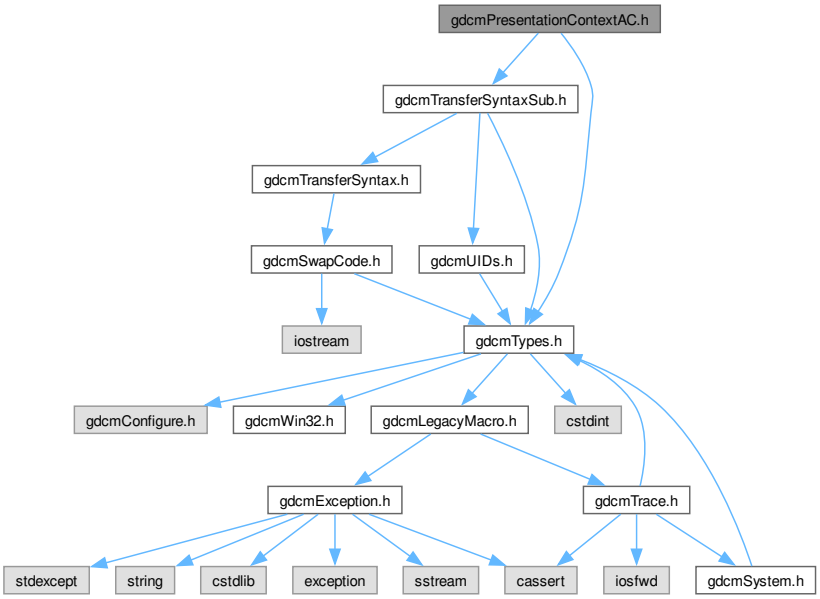
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPRESENTATIONCONTEXT_H
#define GDCMPRESENTATIONCONTEXT_H

#include "gdcmTypes.h"
#include "gdcmUIDs.h"

#include <vector>

namespace gdcm
{

class GDCM_EXPORT PresentationContext
{
public:
  PresentationContext();

  PresentationContext( UIDs::TSName asname,
    UIDs::TSName tsname = UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM );

  void SetAbstractSyntax( const char *absyn ) { AbstractSyntax = absyn; }
  const char *GetAbstractSyntax() const { return AbstractSyntax.c_str(); }

  void AddTransferSyntax( const char *tsstr );
  typedef std::vector<std::string> TransferSyntaxArrayType;
  typedef TransferSyntaxArrayType::size_type SizeType;
  const char *GetTransferSyntax(SizeType i) const { return TransferSyntaxes[i].c_str(); }
  SizeType GetNumberOfTransferSyntaxes() const { return TransferSyntaxes.size(); }

  void SetPresentationContextID( uint8_t id );
  uint8_t GetPresentationContextID() const;

  void Print(std::ostream &os) const;

  bool operator==(const PresentationContext & pc) const
    {
    assert( TransferSyntaxes.size() == 1 ); // TODO
    assert( pc.TransferSyntaxes.size() == 1 );
    return AbstractSyntax == pc.AbstractSyntax && TransferSyntaxes == pc.TransferSyntaxes;
    }

protected :
  std::string AbstractSyntax;
  std::vector<std::string> TransferSyntaxes;
  uint8_t /*PresentationContext*/ID;
};

} // end namespace gdcm

#endif //GDCMPRESENTATIONCONTEXT_H
```

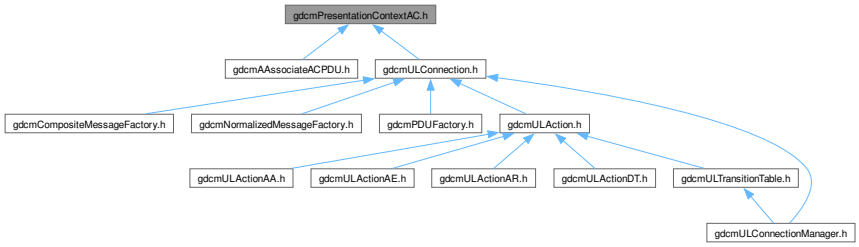
13.545 gdcmPresentationContextAC.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTransferSyntaxSub.h"
```

Include dependency graph for gdcmPresentationContextAC.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::PresentationContextAC

  PresentationContextAC.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.546 gdcmPresentationContextAC.h

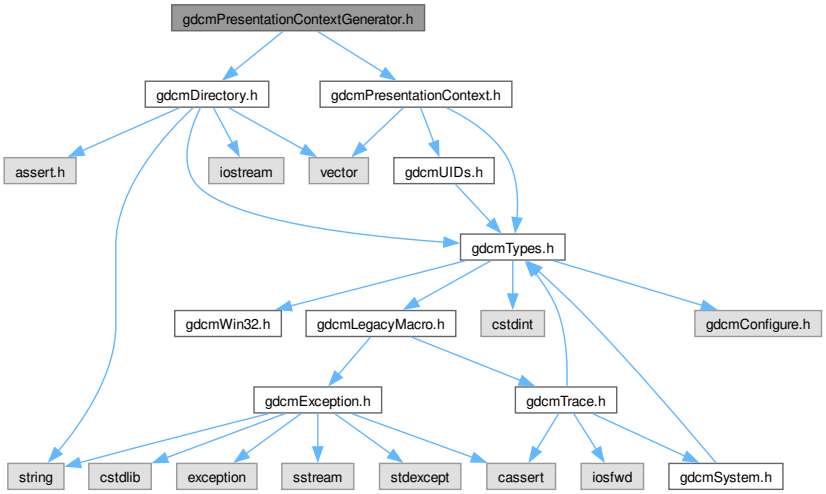
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPRESENTATIONCONTEXTAC_H
#define GDCMPRESENTATIONCONTEXTAC_H

#include "gdcmTypes.h"
#include "gdcmTransferSyntaxSub.h"

namespace gdcm
{

namespace network
{

class PresentationContextAC
{
public:
  PresentationContextAC();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;

  void SetTransferSyntax( TransferSyntaxSub const &ts );
  void SetPresentationContextID( uint8_t id );

  void Print(std::ostream &os) const;

  uint8_t GetPresentationContextID() const
    {
    return ID;
    }
  TransferSyntaxSub const & GetTransferSyntax() const { return SubItems; }

  void SetReason( uint8_t r ) { Result = r; }
  uint8_t GetReason() const { return Result; }

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength; // len of last transfer syntax
  uint8_t /*PresentationContext*/ID;
  static const uint8_t Reserved6;
  uint8_t /*Reason*/Result;
  static const uint8_t Reserved8;
  TransferSyntaxSub SubItems;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMPRESENTATIONCONTEXTAC_H
```

13.547 gdcmPresentationContextGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmPresentationContextGenerator.h:



Classes

- class gdcm::PresentationContextGenerator

  PresentationContextGenerator.

Namespaces

- namespace gdcm

13.548 gdcmPresentationContextGenerator.h

Go to the documentation of this file.

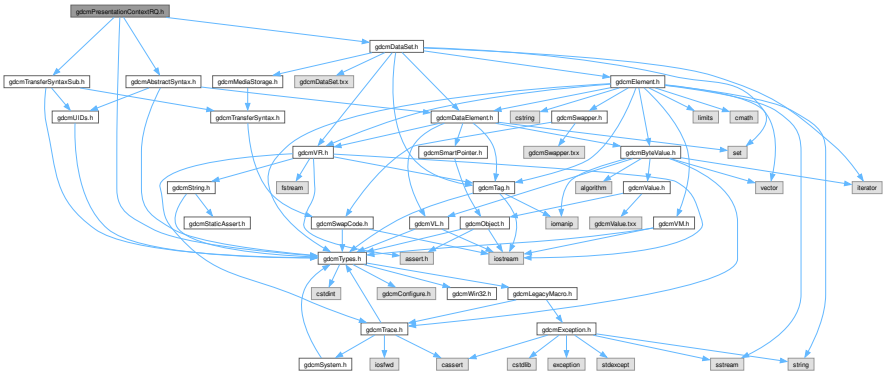
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPRESENTATIONCONTEXTGENERATOR_H
#define GDCMPRESENTATIONCONTEXTGENERATOR_H

#include "gdcmDirectory.h"
#include "gdcmPresentationContext.h"

namespace gdcm
{
```

```
class TransferSyntax;
class File;

class GDCM_EXPORT PresentationContextGenerator
{
public:
  PresentationContextGenerator();

  // Set MergeMode
  // Default mode, each pair AbstractSyntax/TransferSyntax are only merged when
  // exactly identical
  void SetMergeModeToAbstractSyntax();

  // Set MergeMode
  // Merge is done on a per AbstractSyntax basis. Any new TransferSyntax for a
  // given AbstractSyntax is merge to the existing PresentationContext referring
  // to that AbstractSyntax
  void SetMergeModeToTransferSyntax();

  bool GenerateFromUID(UIDs::TSName asname);

  bool GenerateFromFilenames(const Directory::FilenamesType &files);

  bool AddFromFile(const File &file);

  typedef std::vector<PresentationContext> PresentationContextArrayType;
  typedef PresentationContextArrayType::size_type SizeType;
  PresentationContextArrayType const &GetPresentationContexts() { return PresContext; }

  void SetDefaultTransferSyntax( const TransferSyntax &ts );
protected:
  bool AddPresentationContext( const char *absyn, const char *ts );
  const char *GetDefaultTransferSyntax() const;

private:
  std::vector<PresentationContext> PresContext;
  static std::string DefaultTransferSyntax;
};

} // end namespace gdcm

#endif //GDCMPRESENTATIONCONTEXTGENERATOR_H
```

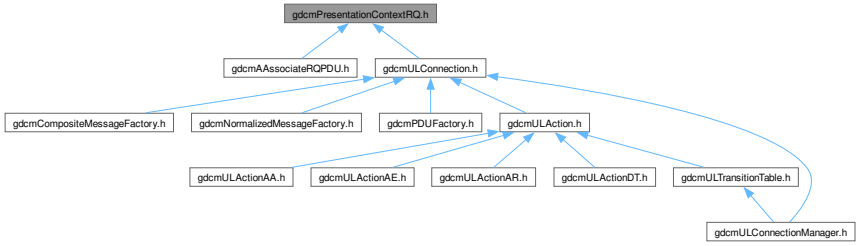
13.549 gdcmPresentationContextRQ.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmAbstractSyntax.h"
#include "gdcmTransferSyntaxSub.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmPresentationContextRQ.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::PresentationContextRQ

  PresentationContextRQ.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.550 gdcmPresentationContextRQ.h

Go to the documentation of this file.

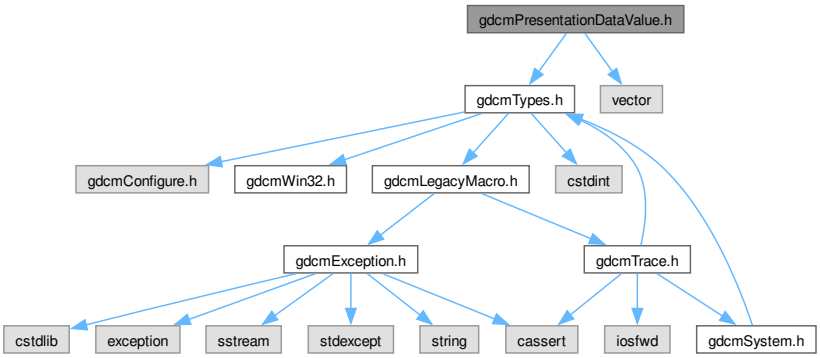
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPRESENTATIONCONTEXTRQ_H
#define GDCMPRESENTATIONCONTEXTRQ_H

#include "gdcmTypes.h"
#include "gdcmAbstractSyntax.h"
#include "gdcmTransferSyntaxSub.h"
#include "gdcmDataSet.h"

namespace gdcm
{
class PresentationContext;
namespace network
{

class GDCM_EXPORT PresentationContextRQ
{
public:
  PresentationContextRQ();

```

```
  PresentationContextRQ( UIDs::TSName asname, UIDs::TSName tsname =
    UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM  );

  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;
  size_t Size() const;

  void SetAbstractSyntax( AbstractSyntax const & absyn );
  AbstractSyntax const &GetAbstractSyntax() const { return SubItems; }
  AbstractSyntax &GetAbstractSyntax() { return SubItems; }

  void AddTransferSyntax( TransferSyntaxSub const &ts );
  typedef std::vector<TransferSyntaxSub>::size_type SizeType;
  TransferSyntaxSub const & GetTransferSyntax(SizeType i) const { return TransferSyntaxes[i]; }
  TransferSyntaxSub & GetTransferSyntax(SizeType i) { return TransferSyntaxes[i]; }
  std::vector<TransferSyntaxSub> const & GetTransferSyntaxes() const {return TransferSyntaxes; }
  SizeType GetNumberOfTransferSyntaxes() const { return TransferSyntaxes.size(); }

  void SetPresentationContextID( uint8_t id );
  uint8_t GetPresentationContextID() const;

  void Print(std::ostream &os) const;

  bool operator==(const PresentationContextRQ & pc) const
    {
    assert( TransferSyntaxes.size() == 1 ); // TODO
    assert( pc.TransferSyntaxes.size() == 1 );
    return SubItems == pc.SubItems && TransferSyntaxes == pc.TransferSyntaxes;
    }

  PresentationContextRQ(const PresentationContext & pc);

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength; // len of last transfer syntax
  uint8_t /*PresentationContext*/ID;
  static const uint8_t Reserved6;
  static const uint8_t Reserved7;
  static const uint8_t Reserved8;
  /*
  This variable field shall contain the following sub-items: one Abstract
  Syntax and one or more Transfer Syntax(es). For a complete
  description of the use and encoding of these sub-items see Sections
  9.3.2.2.1 and 9.3.2.2.2.
   */
  AbstractSyntax SubItems;
  std::vector<TransferSyntaxSub> TransferSyntaxes;
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMPRESENTATIONCONTEXTRQ_H
```

13.551 gdcmPresentationDataValue.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
```

Include dependency graph for gdcmPresentationDataValue.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::network::PresentationDataValue

  PresentationDataValue.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.552 gdcmPresentationDataValue.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
```

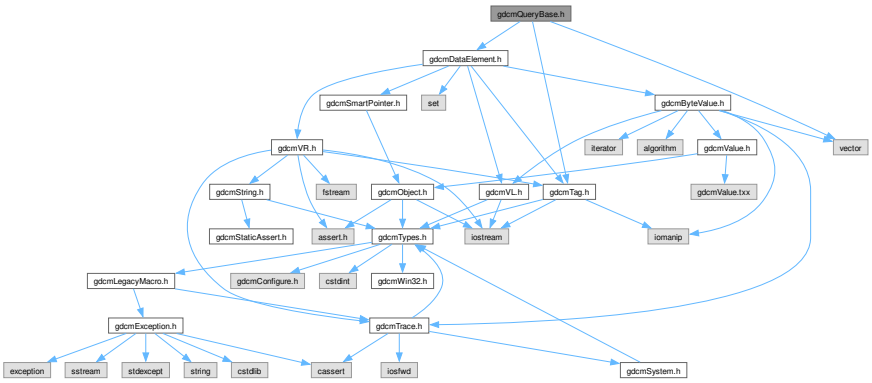
```
#ifndef GDCMPRESENTATIONDATAVALUE_H
#define GDCMPRESENTATIONDATAVALUE_H

#include "gdcmTypes.h"

#include <vector>

namespace gdcm
{
class DataSet;
namespace network
{

class GDCM_EXPORT PresentationDataValue
{
public:
  PresentationDataValue();
  std::istream &Read(std::istream &is);
  std::istream &ReadInto(std::istream &is, std::ostream &os);

  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;

  void SetDataSet(const DataSet & ds);
  void SetBlob(const std::string & partialblob);
  const std::string &GetBlob() const;

  uint8_t GetPresentationContextID() const { return PresentationContextID; }
  void SetPresentationContextID(uint8_t id) {
    assert( id );
    PresentationContextID = id;
  }
  uint8_t GetMessageHeader() const {
    assert( MessageHeader <= 0x3 );
    return MessageHeader;
  }
  // E.2 MESSAGE CONTROL HEADER ENCODING
  // Only the first two bits are considered
  void SetMessageHeader(uint8_t messageheader) {
    MessageHeader = messageheader;
    assert( MessageHeader <= 0x3 );
  }
  //flip the least significant bit of the message header to 1
  //if this is a command, else set it to 0.
  void SetCommand(bool inCommand);
  void SetLastFragment(bool inLast);//set to true if this is the last PDV of a set

  bool GetIsCommand() const;
  bool GetIsLastFragment() const;

  void Print(std::ostream &os) const;

  //NOTE that the PDVs have to be given in the order in which they were received!
  //also note that a dataset may be across multiple PDVs
  static DataSet ConcatenatePDVBlobs(const std::vector<PresentationDataValue>& inPDVs);

  static DataSet ConcatenatePDVBlobsAsExplicit(const std::vector<PresentationDataValue>& inPDVs);

private:
  uint32_t ItemLength;
  uint8_t PresentationContextID;
  uint8_t MessageHeader;
  std::string Blob;
};
} // end namespace network

} // end namespace gdcm

#endif //GDCMPRESENTATIONDATAVALUE_H
```

13.553 gdcmQueryBase.h File Reference

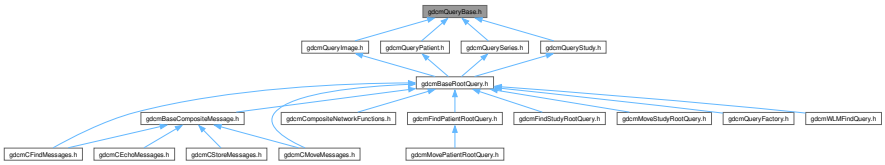
```
#include "gdcmTag.h"
#include "gdcmDataElement.h"
```

```
#include <vector>
```

Include dependency graph for gdcmQueryBase.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QueryBase

  QueryBase.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ERootType {
  gdcm::ePatientRootType ,
  gdcm::eStudyRootType }

13.554 gdcmQueryBase.h

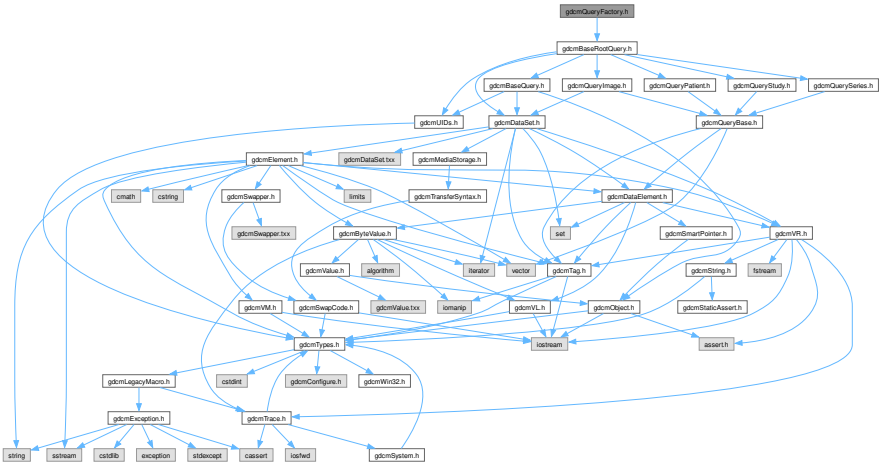
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMQUERYBASE_H
#define GDCMQUERYBASE_H

#include "gdcmTag.h"
#include "gdcmDataElement.h"

#include <vector>

namespace gdcm
{
  enum ERootType
    {
    ePatientRootType,
    eStudyRootType
    };

class GDCM_EXPORT QueryBase
{
  public:
    virtual ~QueryBase() = default;

    virtual std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const = 0;
    virtual std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const = 0;
    virtual std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const = 0;
    // C.4.1.2.1 Baseline Behavior of SCU
    // All C-FIND SCUs shall be capable of generating query requests which
    // meet the requirements of the Hierarchical Search.
    // The Identifier contained in a C-FIND request shall contain a single
    // value in the Unique Key Attribute for each level above the
    // Query/Retrieve level. No Required or Optional Keys shall be
    // specified which are associated with levels above the Query/Retrieve
    // level.
    virtual std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const = 0;

    std::vector<Tag> GetAllTags(const ERootType& inRootType) const;

    std::vector<Tag> GetAllRequiredTags(const ERootType& inRootType) const;

    virtual const char * GetName() const = 0;
    virtual DataElement GetQueryLevel() const = 0;
  };
}

#endif //GDCMQUERYBASE_H
```

13.555 gdcmQueryFactory.h File Reference

`#include "gdcmBaseRootQuery.h"`

Include dependency graph for gdcmQueryFactory.h:



Classes

- class gdcm::QueryFactory

  QueryFactory.h.

Namespaces

- namespace gdcm

Enumerations

- enum gdcm::ECharSet {
  gdcm::eLatin1 = 0 ,
  gdcm::eLatin2 ,
  gdcm::eLatin3 ,
  gdcm::eLatin4 ,
  gdcm::eCyrillic ,
  gdcm::eArabic ,
  gdcm::eGreek ,
  gdcm::eHebrew ,
  gdcm::eLatin5 ,
  gdcm::eJapanese ,
  gdcm::eThai ,
  gdcm::eJapaneseKanjiMultibyte ,
  gdcm::eJapaneseSupplementaryKanjiMultibyte ,
  gdcm::eKoreanHangulHanjaMultibyte ,
  gdcm::eUTF8 ,
  gdcm::eGB18030 }

13.556 gdcmQueryFactory.h

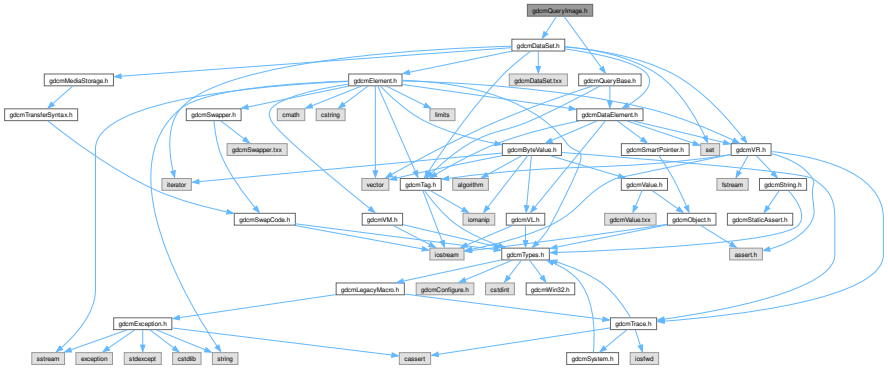
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMQUERYFACTORY_H
#define GDCMQUERYFACTORY_H

#include "gdcmBaseRootQuery.h"

namespace gdcm{
  enum ECharSet {
    eLatin1 = 0,
    eLatin2,
    eLatin3,
    eLatin4,
    eCyrillic,
    eArabic,
    eGreek,
    eHebrew,
    eLatin5, // Latin Alphabet No. 5 (Turkish) Extended
    eJapanese, // JIS X 0201 (Shift JIS) Extended
    eThai, // TIS 620-2533 (Thai) Extended
    eJapaneseKanjiMultibyte, // JIS X 0208 (Kanji) Extended
    eJapaneseSupplementaryKanjiMultibyte, // JIS X 0212 (Kanji) Extended
    eKoreanHangulHanjaMultibyte, // KS X 1001 (Hangul and Hanja) Extended
    eUTF8,
    eGB18030 // Chinese (Simplified) Extended
  };

class GDCM_EXPORT QueryFactory
{
public:

  static BaseQuery* ProduceQuery( const std::string & sopInstanceUID, ENQueryType inQueryType );
  static BaseRootQuery* ProduceQuery(ERootType inRootType, EQueryType inQueryType,
    EQueryLevel inQueryLevel);

  static DataElement ProduceCharacterSetDataElement(
    const std::vector<ECharSet>& inCharSetType);

  static ECharSet GetCharacterFromCurrentLocale();

  static void ListCharSets(std::ostream& os);
};

} // end namespace gdcm

#endif // GDCMQUERYFACTORY_H
```

13.557 gdcmQueryImage.h File Reference

```
#include "gdcmQueryBase.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmQueryImage.h:



This graph shows which files directly or indirectly include this file:

Classes

- class gdcm::QueryImage

  QueryImage.

Namespaces

- namespace gdcm

13.558 gdcmQueryImage.h

Go to the documentation of this file.

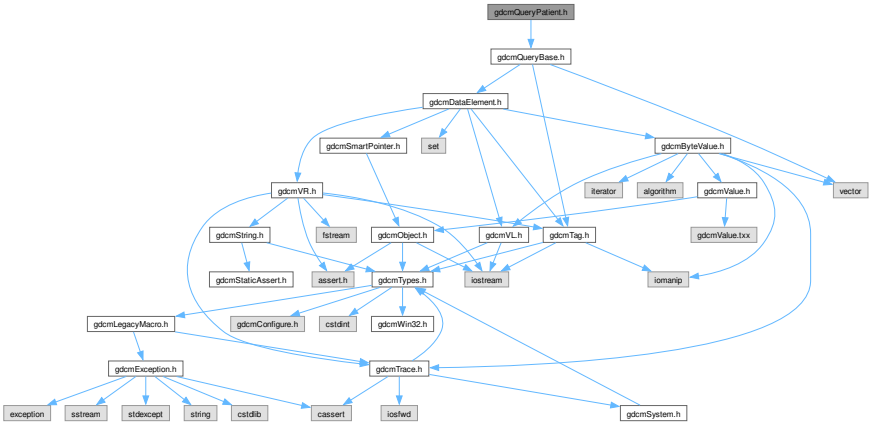
```
00001 /*=========================================================================
00002  *
00003  *  Copyright NumFOCUS
00004  *
00005  *  Licensed under the Apache License, Version 2.0 (the "License");
00006  *  you may not use this file except in compliance with the License.
00007  *  You may obtain a copy of the License at
00008  *
00009  *         http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
00011  *  Unless required by applicable law or agreed to in writing, software
00012  *  distributed under the License is distributed on an "AS IS" BASIS,
00013  *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
00014  *  See the License for the specific language governing permissions and
```

```
 *  limitations under the License.
 *
 *=============================================================================*/
#ifndef GDCMQUERYIMAGE_H
#define GDCMQUERYIMAGE_H

#include "gdcmQueryBase.h"
#include "gdcmDataSet.h"

namespace gdcm
{
class GDCM_EXPORT QueryImage : public QueryBase
{
public:
  std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;

  const char * GetName() const override;

  DataElement GetQueryLevel() const override;
};

} // end namespace gdcm

#endif // GDCMQUERYIMAGE_H
```

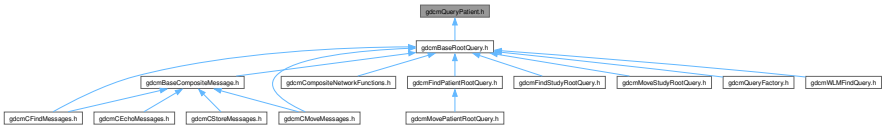
13.559 gdcmQueryPatient.h File Reference

`#include "gdcmQueryBase.h"`

Include dependency graph for gdcmQueryPatient.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QueryPatient

  QueryPatient.

Namespaces

- namespace gdcm

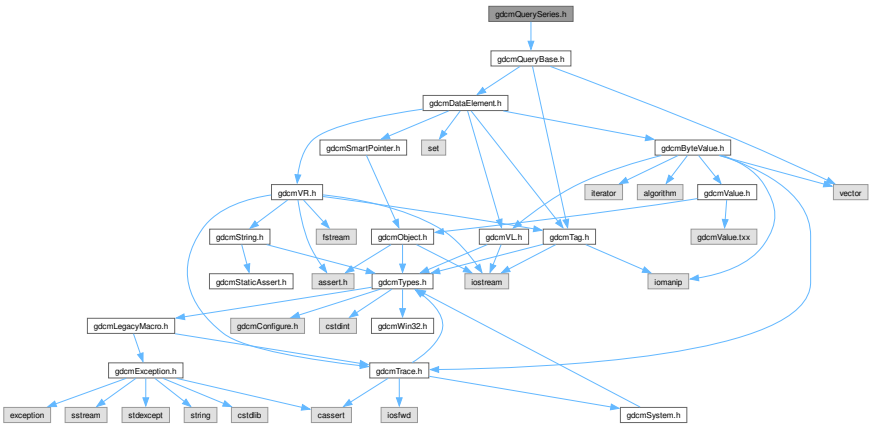
13.560 gdcmQueryPatient.h

Go to the documentation of this file.

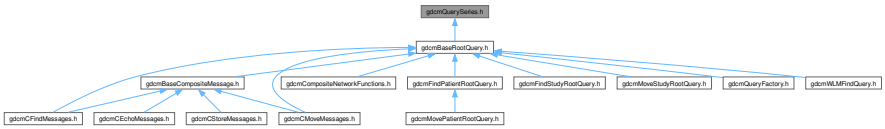
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMQUERYPATIENT_H
#define GDCMQUERYPATIENT_H

#include "gdcmQueryBase.h"

namespace gdcm
{
class GDCM_EXPORT QueryPatient : public QueryBase
{
public:
  std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;

  const char * GetName() const override;
  DataElement GetQueryLevel() const override;
};

} // end namespace gdcm

#endif //GDCMQUERYPATIENT_H
```

13.561 gdcmQuerySeries.h File Reference

`#include "gdcmQueryBase.h"`
Include dependency graph for gdcmQuerySeries.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QuerySeries

  QuerySeries.

Namespaces

- namespace gdcm

13.562 gdcmQuerySeries.h

Go to the documentation of this file.

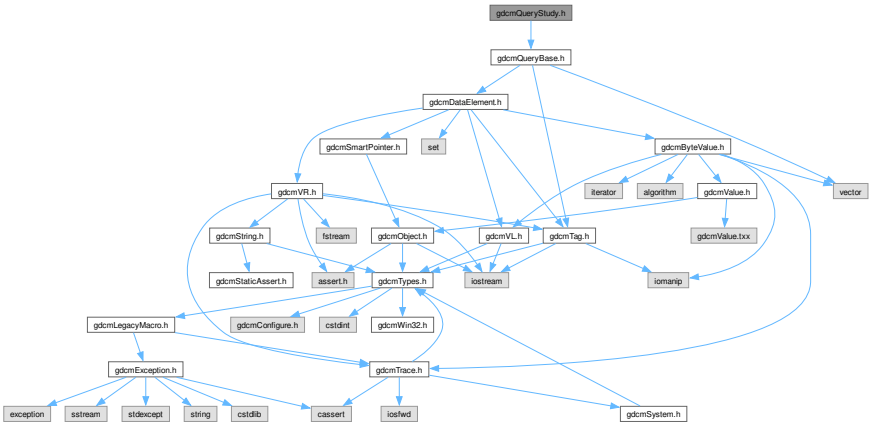
```
00001 /*=========================================================================
00002  *
00003  *  Copyright NumFOCUS
00004  *
00005  *  Licensed under the Apache License, Version 2.0 (the "License");
```

```
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMQUERYSERIES_H
#define GDCMQUERYSERIES_H

#include "gdcmQueryBase.h"

namespace gdcm
{
class GDCM_EXPORT QuerySeries : public QueryBase
{
public:
  std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;

  const char * GetName() const override;
  DataElement GetQueryLevel() const override;
};

} // end namespace gdcm

#endif //GDCMQUERYSERIES_H
```

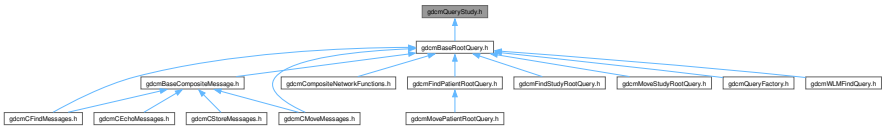
13.563 gdcmQueryStudy.h File Reference

`#include "gdcmQueryBase.h"`

Include dependency graph for gdcmQueryStudy.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QueryStudy

  QueryStudy.h.

Namespaces

- namespace gdcm

13.564 gdcmQueryStudy.h

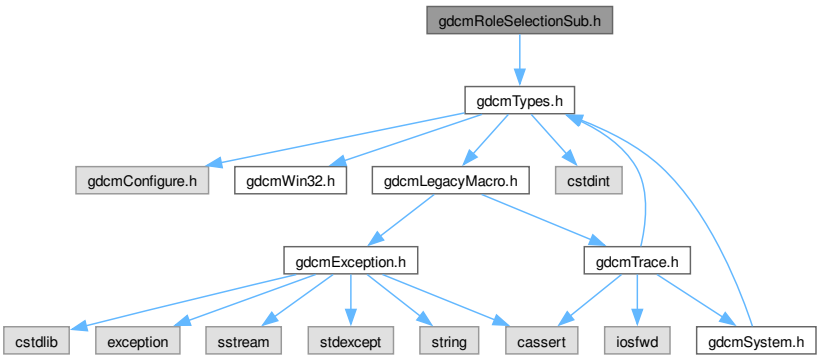
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMQUERYSTUDY_H
#define GDCMQUERYSTUDY_H

#include "gdcmQueryBase.h"

namespace gdcm
{
class GDCM_EXPORT QueryStudy : public QueryBase
{
public:
  std::vector<Tag> GetRequiredTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetUniqueTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetOptionalTags(const ERootType& inRootType) const override;
  std::vector<Tag> GetHierachicalSearchTags(const ERootType& inRootType) const override;

  const char *GetName() const override;
  DataElement GetQueryLevel() const override;
};

} // end namespace gdcm

#endif //GDCMQUERYSTUDY_H
```

13.565 gdcmRoleSelectionSub.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmRoleSelectionSub.h:



Classes

- class gdcm::network::RoleSelectionSub

  RoleSelectionSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.566 gdcmRoleSelectionSub.h

Go to the documentation of this file.

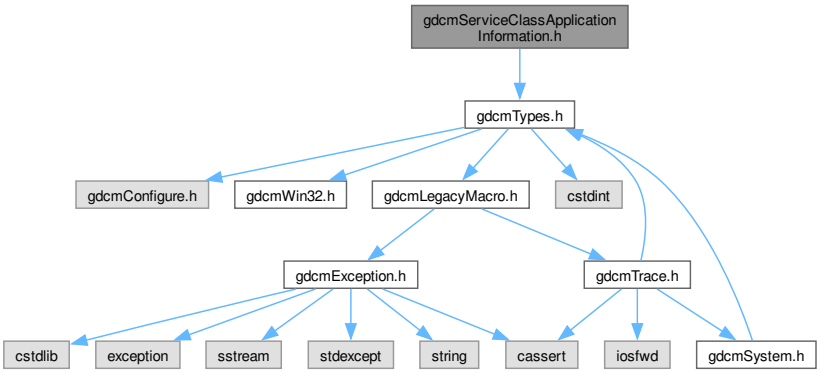
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMROLESELECTIONSUB_H
#define GDCMROLESELECTIONSUB_H

#include "gdcmTypes.h"

namespace gdcm
{
```

```
namespace network
{

class RoleSelectionSub
{
public:
  RoleSelectionSub();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;
  void Print(std::ostream &os) const;

  void SetTuple(const char *uid, uint8_t scurole, uint8_t scprole);

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength;
  uint16_t UIDLength;
  std::string /*SOP-class-uid*/ Name; // UID
  uint8_t SCURole;
  uint8_t SCPRole;
};

} // end namespace network

} // end namespace gdcm

#endif // GDCMROLESELECTIONSUB_H
```

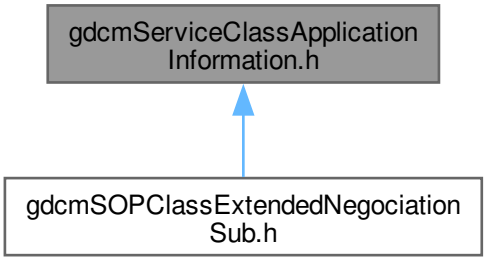
13.567 gdcmServiceClassApplicationInformation.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmServiceClassApplicationInformation.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ServiceClassApplicationInformation

Namespaces

- namespace gdcm
- namespace gdcm::network

13.568 gdcmServiceClassApplicationInformation.h

Go to the documentation of this file.

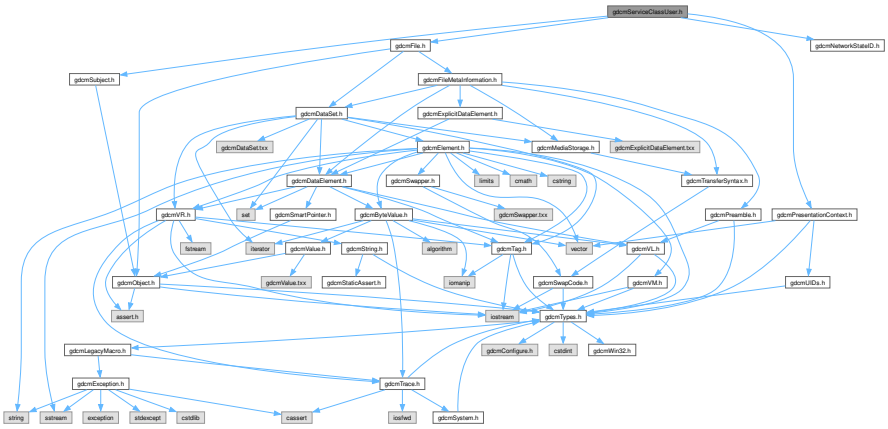
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 #ifndef GDCMSERVICECLASSAPPLICATIONINFORMATION_H
00015 #define GDCMSERVICECLASSAPPLICATIONINFORMATION_H
00016
00017 #include "gdcmTypes.h"
00018
00019 namespace gdcm
00020 {
00021
00022 namespace network
00023 {
00024
00030 class ServiceClassApplicationInformation
00031 {
00032 public:
```

```
  ServiceClassApplicationInformation();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;
  void SetTuple(uint8_t levelofsupport, uint8_t levelofdigitalsig,
    uint8_t elementcoercion);

  void Print(std::ostream &os) const;
private:
  uint8_t InternalArray[6];
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMSERVICECLASSAPPLICATIONINFORMATION_H
```

13.569 gdcmServiceClassUser.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
#include "gdcmFile.h"
#include "gdcmNetworkStateID.h"
```

Include dependency graph for gdcmServiceClassUser.h:



Classes

- class gdcm::ServiceClassUser

  ServiceClassUser.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.570 gdcmServiceClassUser.h

Go to the documentation of this file.

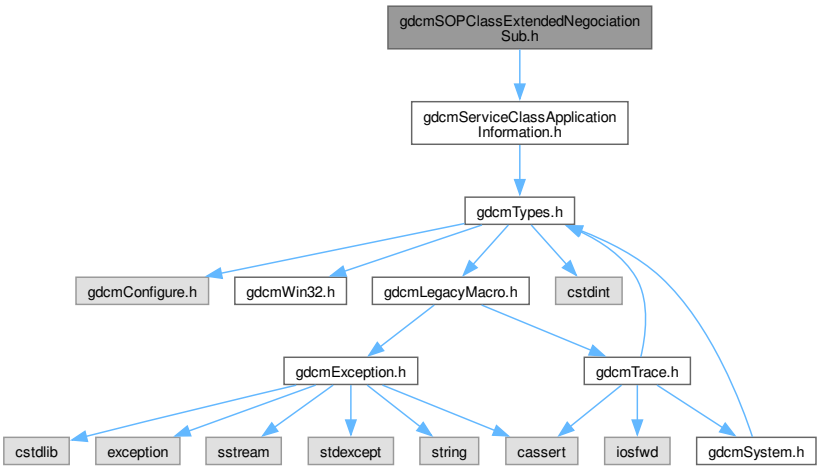
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSERVICECLASSUSER_H
#define GDCMSERVICECLASSUSER_H

#include "gdcmSubject.h"

#include "gdcmPresentationContext.h"
#include "gdcmFile.h"

#include "gdcmNetworkStateID.h" // EStateID

namespace gdcm
{
class ServiceClassUserInternals;
class BaseRootQuery;
namespace network{
class ULEvent;
class ULConnection;
class ULConnectionCallback;
}
class GDCM_EXPORT ServiceClassUser : public Subject
{
public:
  ServiceClassUser();
  ~ServiceClassUser() override;
  ServiceClassUser(const ServiceClassUser&) = delete;
  void operator=(const ServiceClassUser &) = delete;

  void SetHostname( const char *hostname );

  void SetPort( uint16_t port );

  void SetPortSCP( uint16_t portscp );

  void SetAETitle(const char *aetitle);
  const char *GetAETitle() const;

  void SetCalledAETitle(const char *aetitle);
  const char *GetCalledAETitle() const;

  void SetTimeout(double t);
  double GetTimeout() const;

  bool InitializeConnection();

  void SetPresentationContexts(std::vector<PresentationContext> const & pcs);

  bool IsPresentationContextAccepted(const PresentationContext& pc) const;

  bool StartAssociation();

  bool StopAssociation();

  bool SendEcho();

  bool SendStore(const char *filename);
  bool SendStore(File const &file);
  bool SendStore(DataSet const &ds);

  bool SendFind(const BaseRootQuery* query, std::vector<DataSet> &retDatasets);

  bool SendMove(const BaseRootQuery* query, const char *outputdir);
  bool SendMove(const BaseRootQuery* query, std::vector<DataSet> &retDatasets);
```

```
  bool SendMove(const BaseRootQuery* query, std::vector<File> &retFile);

  static SmartPointer<ServiceClassUser> New() { return new ServiceClassUser; }

private:
  network::EStateID RunEventLoop(network::ULEvent& inEvent,
    network::ULConnection* inWhichConnection,
    network::ULConnectionCallback* inCallback, const bool& startWaiting);
  network::EStateID RunMoveEventLoop(network::ULEvent& inEvent,
    network::ULConnectionCallback* inCallback);

private:
  ServiceClassUserInternals *Internals;
};

} // end namespace gdcm

#endif // GDCMSERVICECLASSUSER_H
```

13.571 gdcmSOPClassExtendedNegociationSub.h File Reference

`#include "gdcmServiceClassApplicationInformation.h"`

Include dependency graph for gdcmSOPClassExtendedNegociationSub.h:



Classes

- class gdcm::network::SOPClassExtendedNegociationSub

  SOPClassExtendedNegociationSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.572 gdcmSOPClassExtendedNegociationSub.h

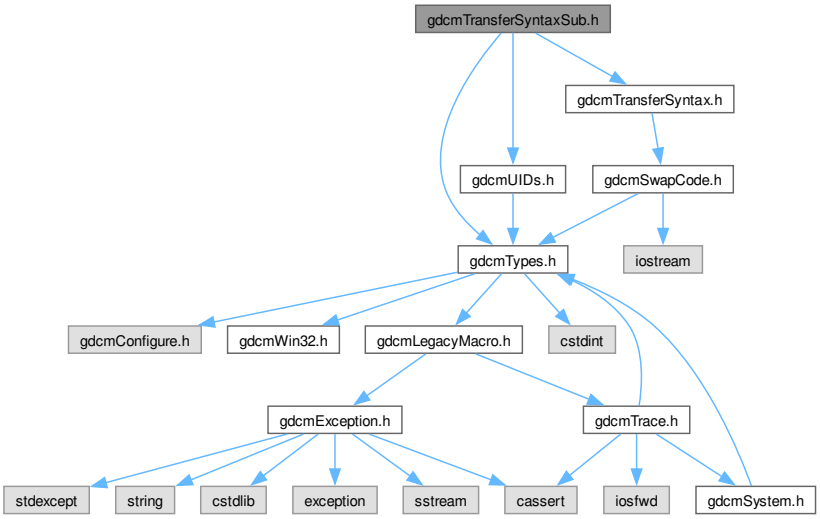
Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
#define GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H

#include "gdcmServiceClassApplicationInformation.h"

namespace gdcm
{
namespace network
{

class SOPClassExtendedNegociationSub
{
public:
  SOPClassExtendedNegociationSub();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;

  size_t Size() const;
  void Print(std::ostream &os) const;

  void SetTuple(const char *uid, uint8_t levelofsupport = 3,
    uint8_t levelofdigitalsig = 0,
    uint8_t elementcoercion = 2);

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength;
  uint16_t UIDLength;
  std::string /*SOP-class-uid*/ Name; // UID
  ServiceClassApplicationInformation SCAI;
};

} // end namespace network

} // end namespace gdcm

#endif // GDCMSOPCLASSEXTENDEDNEGOCIATIONSUB_H
```

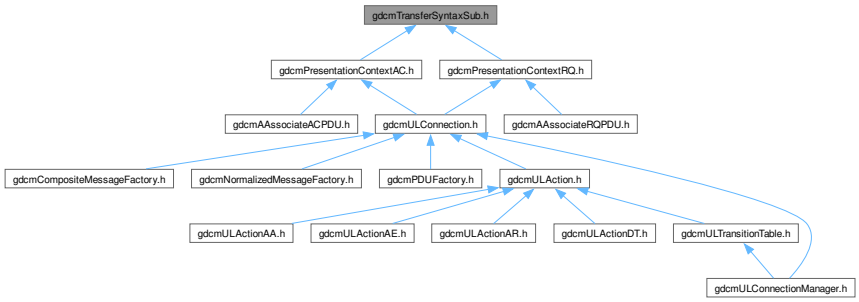
13.573 gdcmTransferSyntaxSub.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDs.h"
```

Include dependency graph for gdcmTransferSyntaxSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::TransferSyntaxSub

  TransferSyntaxSub.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.574 gdcmTransferSyntaxSub.h

Go to the documentation of this file.

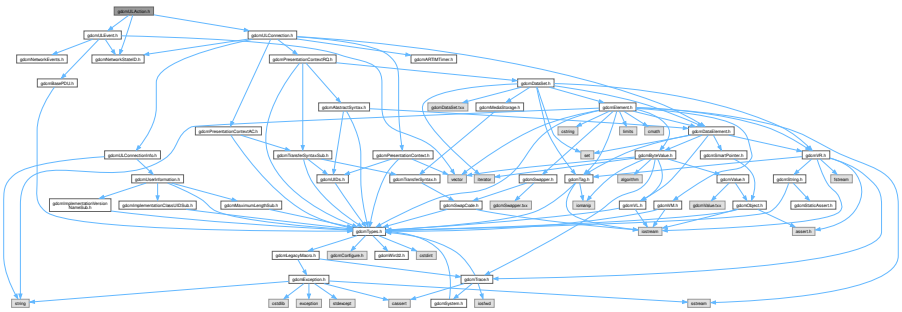
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMTRANSFERSYNTAXSUB_H
#define GDCMTRANSFERSYNTAXSUB_H

#include "gdcmTypes.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDs.h"

namespace gdcm
{

namespace network
{

class TransferSyntaxSub
{
public:
  TransferSyntaxSub();
  void SetName( const char *name );
  const char *GetName() const { return Name.c_str(); }

  // accept a UIDs::TSType also...
  void SetNameFromUID( UIDs::TSName tsname );

  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;
  size_t Size() const;
  void Print(std::ostream &os) const;

  bool operator==(const TransferSyntaxSub & ts) const
    {
    return Name == ts.Name;
    }

private:
  void UpdateName( const char *name );
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength; // len of
  std::string /*TransferSyntaxSub*/ Name; // UID
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMTRANSFERSYNTAXSUB_H
```

13.575 gdcmULAction.h File Reference

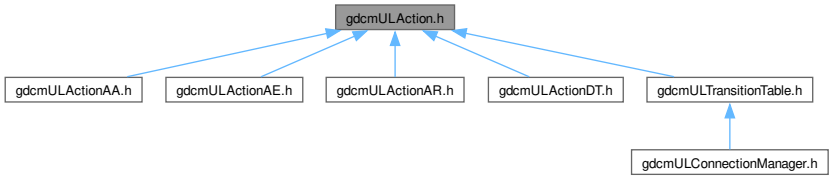
```
#include "gdcmNetworkStateID.h"
#include "gdcmULEvent.h"
```

```
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmULAction.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULAction

  ULAction.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.576 gdcmULAction.h

Go to the documentation of this file.

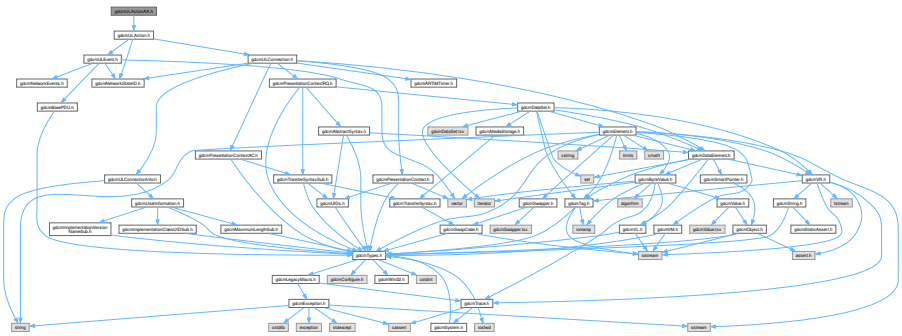
```
00001 /*=========================================================================
00002  *
00003  *  Copyright NumFOCUS
00004  *
00005  *  Licensed under the Apache License, Version 2.0 (the "License");
00006  *  you may not use this file except in compliance with the License.
00007  *  You may obtain a copy of the License at
00008  *
00009  *         http://www.apache.org/licenses/LICENSE-2.0.txt
00010  *
```

```
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=============================================================================*/
#ifndef GDCMULACTION_H
#define GDCMULACTION_H

#include "gdcmNetworkStateID.h"
#include "gdcmULEvent.h"
#include "gdcmULConnection.h"

namespace gdcm {
class Subject;
  namespace network {

class ULAction {

    protected:


    public:
      ULAction() = default;
      //make sure destructors are virtual to avoid memory leaks
      virtual ~ULAction() = default;
      //cannot copy a ULAction
      ULAction(const ULAction& inAction) = delete;
      void operator=(const ULAction&) = delete;

      virtual EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) = 0;
    };
  }
}

#endif // GDCMULACTION_H
```

13.577 gdcmULActionAA.h File Reference

`#include "gdcmULAction.h"`

Include dependency graph for gdcmULActionAA.h:



Classes

- class gdcm::network::ULActionAA1
- class gdcm::network::ULActionAA2
- class gdcm::network::ULActionAA3

- class gdcm::network::ULActionAA4
- class gdcm::network::ULActionAA5
- class gdcm::network::ULActionAA6
- class gdcm::network::ULActionAA7
- class gdcm::network::ULActionAA8

Namespaces

- namespace gdcm
- namespace gdcm::network

13.578 gdcmULActionAA.h

Go to the documentation of this file.

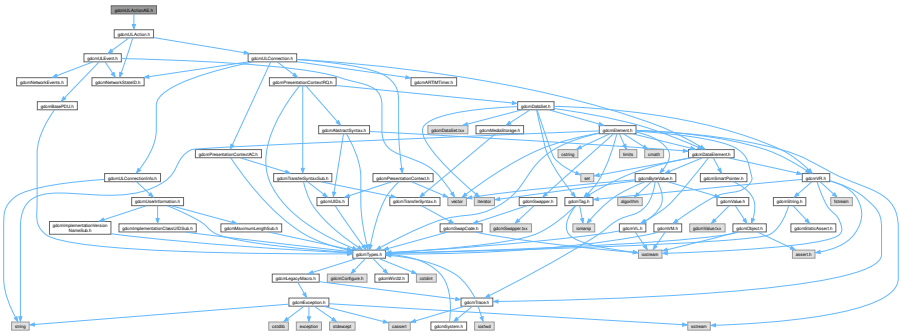
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULACTIONAA_H
#define GDCMULACTIONAA_H

#include "gdcmULAction.h"


namespace gdcm {
  namespace network {

    //Send A-ABORT PDU (service-user source) and start (or restart if already started) ARTIM timer
    //Next State: eSta13AwaitingClose
    class ULActionAA1 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Stop ARTIM timer if running.  Close transport connection.
    //Next State: eSta1Idle
    class ULActionAA2 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //If (service-user initiated abort)
    //- issue A-ABORT indication and close transport connection
    //otherwise (service-provider initiated abort):
    //- issue A-P-ABORT indication and close transport connection
    //Next State: eSta1Idle
    class ULActionAA3 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue A-P-ABORT indication primitive
```

```
    //Next State: eSta1Idle
    class ULActionAA4 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Stop ARTIM timer
    //Next State: eSta1Idle
    class ULActionAA5 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Ignore PDU
    //Next State: eSta13AwaitingClose
    class ULActionAA6 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Send A-ABORT PDU
    //Next State: eSta13AwaitingClose
    class ULActionAA7 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Send A-ABORT PDU (service-provider source), issue an A-P-ABORT indication, and start ARTIM timer
    //Next State: eSta13AwaitingClose
    class ULActionAA8 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };
  }
}

#endif // GDCMULACTIONAA_H
```

13.579 gdcmULActionAE.h File Reference

`#include "gdcmULAction.h"`

Include dependency graph for gdcmULActionAE.h:



Classes

- class gdcm::network::ULActionAE1

- class gdcm::network::ULActionAE2
- class gdcm::network::ULActionAE3
- class gdcm::network::ULActionAE4
- class gdcm::network::ULActionAE5
- class gdcm::network::ULActionAE6
- class gdcm::network::ULActionAE7
- class gdcm::network::ULActionAE8

Namespaces

- namespace gdcm
- namespace gdcm::network

13.580 gdcmULActionAE.h

Go to the documentation of this file.

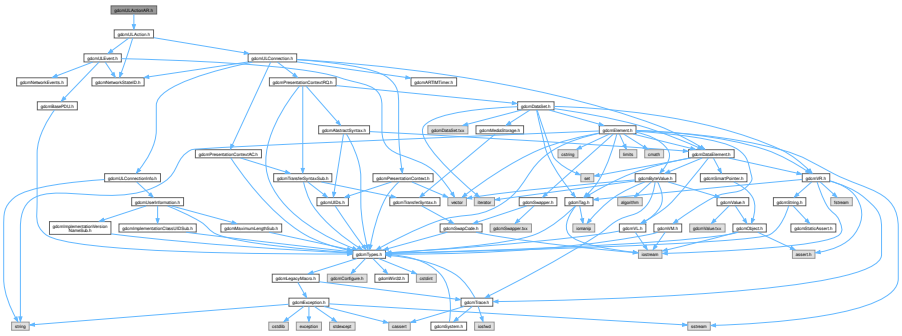
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULACTIONAE_H
#define GDCMULACTIONAE_H

#include "gdcmULAction.h"


namespace gdcm {
  namespace network {

    //Issue TRANSPORT CONNECT request primitive to local transport service.
    class ULActionAE1 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Send A-ASSOCIATE-RQ-PDU
    //Next State: eSta5WaitRemoteAssoc
    class ULActionAE2 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue A-ASSOCIATE confirmation (accept) primitive
    //Next State: eSta6TransferReady
    class ULActionAE3 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue A-ASSOCIATE confirmation (reject) primitive and close transport connection
```

```
    //Next State: eSta1Idle
    class ULActionAE4 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue Transport connection response primitive, start ARTIM timer
    //Next State: eSta2Open
    class ULActionAE5 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Stop ARTIM timer and if A-ASSOCIATE-RQ acceptable by service-provider:
    //- issue A-ASSOCIATE indication primitive
    //Next state: eSta3WaitLocalAssoc
    //otherwise:
    //- issue A-ASSOCIATE-RJ-PDU and start ARTIM timer
    //Next state: eSta13AwaitingClose
    class ULActionAE6 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Send A-ASSOCIATE-AC PDU
    //Next State: eSta6TransferReady
    class ULActionAE7 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Send A-ASSOCIATE-RJ PDU and start ARTIM timer
    //Next State: eSta13AwaitingClose
    class ULActionAE8 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };
  }
}
#endif // GDCMULACTIONAE_H
```

13.581 gdcmULActionAR.h File Reference

`#include "gdcmULAction.h"`

Include dependency graph for gdcmULActionAR.h:



Classes

- class gdcm::network::ULActionAR1
- class gdcm::network::ULActionAR10
- class gdcm::network::ULActionAR2
- class gdcm::network::ULActionAR3
- class gdcm::network::ULActionAR4
- class gdcm::network::ULActionAR5
- class gdcm::network::ULActionAR6
- class gdcm::network::ULActionAR7
- class gdcm::network::ULActionAR8
- class gdcm::network::ULActionAR9

Namespaces

- namespace gdcm
- namespace gdcm::network

13.582 gdcmULActionAR.h

Go to the documentation of this file.

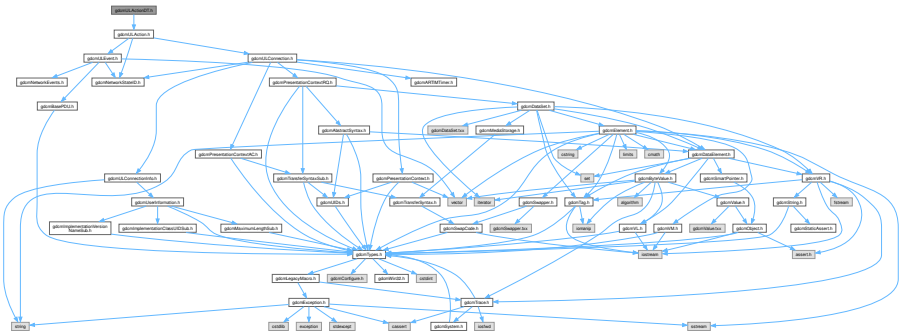
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULACTIONAR_H
#define GDCMULACTIONAR_H

#include "gdcmULAction.h"


namespace gdcm {
  namespace network {

    //Send A-RELEASE-RQ-PDU
    //Next State: eSta7WaitRelease
    class ULActionAR1 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue A-RELEASE indication primitive
    //Next State: eSta8WaitLocalRelease
    class ULActionAR2 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue A-RELEASE confirmation primitive, and close transport connection
```

```
    //Next State: eSta1Idle
    class ULActionAR3 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue A-RELEASE-RP PDU and start ARTIM timer
    //Next State: eSta13AwaitingClose
    class ULActionAR4 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Stop ARTIM timer
    //Next State: eSta1Idle
    class ULActionAR5 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue P-Data indication
    //Next State: eSta7WaitRelease
    class ULActionAR6 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue P-DATA-TF PDU
    //Next State: eSta8WaitLocalRelease
    class ULActionAR7 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue A-RELEASE indication (release collision):
    //- If association-requestor, next state is eSta9ReleaseCollisionRqLocal
    //- if not, next state is eSta10ReleaseCollisionAc
    class ULActionAR8 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Send A-RELEASE-RP PDU
    //Next State: eSta11ReleaseCollisionRq
    class ULActionAR9 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Issue A-RELEASE confirmation primitive
    //Next State: eSta12ReleaseCollisionAcLocal
    class ULActionAR10 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };
  }
}
#endif // GDCMULACTIONAR_H
```

13.583 gdcmULActionDT.h File Reference

`#include "gdcmULAction.h"`

Include dependency graph for gdcmULActionDT.h:



Classes

- class gdcm::network::ULActionDT1
- class gdcm::network::ULActionDT2

Namespaces

- namespace gdcm
- namespace gdcm::network

13.584 gdcmULActionDT.h

Go to the documentation of this file.

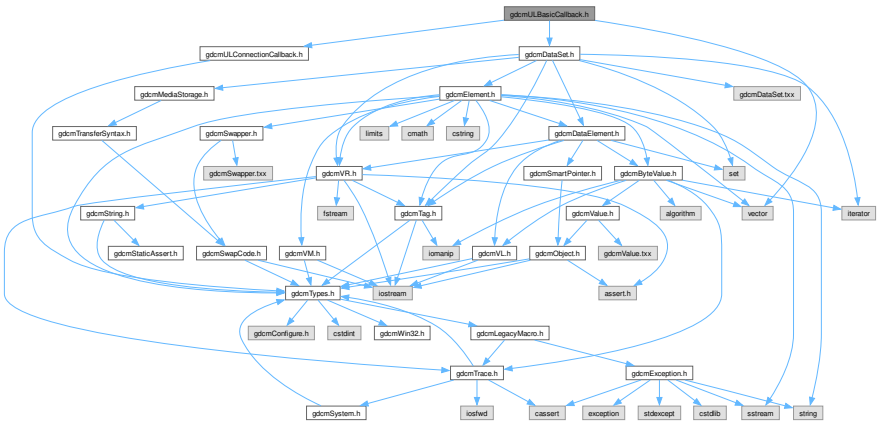
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULACTIONDT_H
#define GDCMULACTIONDT_H

#include "gdcmULAction.h"


namespace gdcm {
  namespace network {
```

```

    //Send P-DATA-TF PDU
    //Next state: eSta6TransferReady
    class ULActionDT1 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };

    //Send P-DATA indication primitive
    //Next state: eSta6TransferReady
    class ULActionDT2 : public ULAction {
    public:
      EStateID PerformAction(Subject *s, ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) override;
    };
  }
}
#endif // GDCMULACTIONDT_H
```

13.585 gdcmULBasicCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
#include "gdcmDataSet.h"
#include <vector>
```

Include dependency graph for gdcmULBasicCallback.h:



Classes

- class gdcm::network::ULBasicCallback

  ULBasicCallback.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.586 gdcmULBasicCallback.h

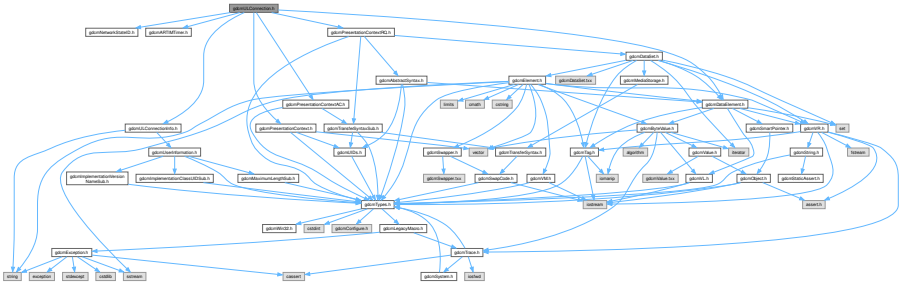
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULCONNECTIONBASICCALLBACK_H
#define GDCMULCONNECTIONBASICCALLBACK_H

#include "gdcmULConnectionCallback.h"
#include "gdcmDataSet.h"
#include <vector>

namespace gdcm
{
  namespace network
    {
    class GDCM_EXPORT ULBasicCallback : public ULConnectionCallback
    {
    std::vector<DataSet> mDataSets;
    std::vector<DataSet> mResponses;
  public:
    ULBasicCallback() = default;
    ~ULBasicCallback() override = default; //empty, for later inheritance

    void HandleDataSet(const DataSet& inDataSet) override;
    void HandleResponse(const DataSet& inDataSet) override;

    std::vector<DataSet> const & GetDataSets() const;
    std::vector<DataSet> const & GetResponses() const;
    };
    } // end namespace network
} // end namespace gdcm

#endif // GDCMULCONNECTIONBASICCALLBACK_H
```

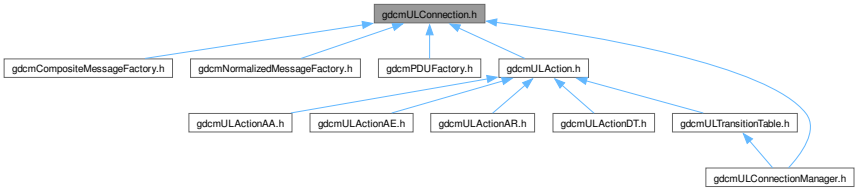
13.587 gdcmULConnection.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmARTIMTimer.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmDataElement.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmULConnection.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULConnection

  ULConnection.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.588 gdcmULConnection.h

Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

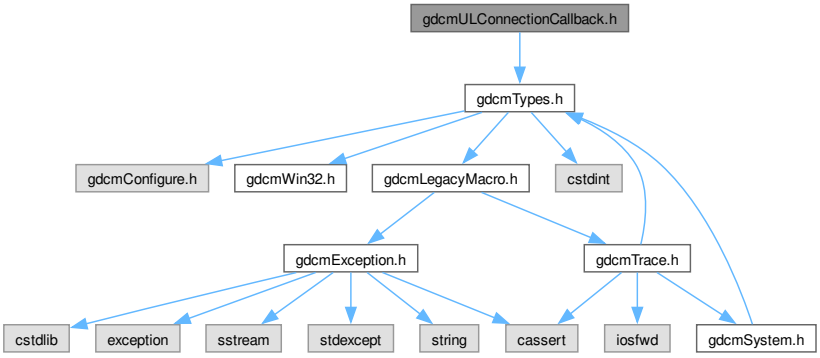
```
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULCONNECTION_H
#define GDCMULCONNECTION_H

#include "gdcmNetworkStateID.h"
#include "gdcmARTIMTimer.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmDataElement.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmPresentationContext.h"

class iosockinet;
class echo;
namespace gdcm{
  namespace network{

class GDCM_EXPORT ULConnection
{
      ULConnectionInfo mInfo;
      //this is a dirty dirty hack
      //but to establish an outgoing connection (scu), we need the echo service
      //to establish incoming, we just need a port and localhost, so an iosockinet works while an
      //echo would fail (probably because one already exists)
      echo* mEcho;
      iosockinet* mSocket;//of the three protocols offered by socket++-- echo, smtp, and ftp--
      //echo most closely matches what the DICOM standard describes as a network connection
      ARTIMTimer mTimer;

      EStateID mCurrentState;

      std::vector<PresentationContextRQ> mPresentationContexts;
      //this is our list of presentation contexts of what we can send
      uint32_t mMaxPDUSize;

      std::vector<PresentationContextAC> mAcceptedPresentationContexts;//these come back from the server
      //and tell us what can be sent over this connection

      TransferSyntaxSub cstorets;

      friend class ULActionAE6;
      void SetCStoreTransferSyntax( TransferSyntaxSub const & ts );
      friend class ULConnectionManager;
      TransferSyntaxSub const & GetCStoreTransferSyntax( ) const;
    public:

      ULConnection(const ULConnectionInfo& inUserInformation);
      //destructors are virtual to prevent memory leaks by inherited classes
      virtual ~ULConnection();

      EStateID GetState() const;
      void SetState(const EStateID& inState);//must be able to update state...

      //echo* GetProtocol();
      std::iostream* GetProtocol();
      void StopProtocol();

      ARTIMTimer& GetTimer();

      const ULConnectionInfo &GetConnectionInfo() const;

      //when the connection is first associated, the connection is told
      //the max packet/PDU size and the way in which to present data
      //(presentation contexts, etc). Store that here.
      void SetMaxPDUSize(uint32_t inSize);
      uint32_t GetMaxPDUSize() const;

      const PresentationContextAC *GetPresentationContextACByID(uint8_t id) const;
      const PresentationContextRQ *GetPresentationContextRQByID(uint8_t id) const;

      uint8_t GetPresentationContextIDFromPresentationContext(PresentationContextRQ const & pc) const;

      std::vector<PresentationContextRQ> const & GetPresentationContexts() const;
      void SetPresentationContexts(const std::vector<PresentationContextRQ>& inContexts);

      void SetPresentationContexts(const std::vector<PresentationContext>& inContexts);

      //given a particular data element, presumably the SOP class,
```

```
      //find the presentation context for that SOP
      //NOT YET IMPLEMENTED
      PresentationContextRQ FindContext(const DataElement& de) const;

      std::vector<PresentationContextAC> const & GetAcceptedPresentationContexts() const;
      std::vector<PresentationContextAC> & GetAcceptedPresentationContexts();
      void AddAcceptedPresentationContext(const PresentationContextAC& inPC);

      bool InitializeConnection();

      bool InitializeIncomingConnection();

      ULConnection(const ULConnection&) = delete;
      void operator=(const ULConnection&) = delete;
    };
  }
}

#endif // ULCONNECTION_H
```

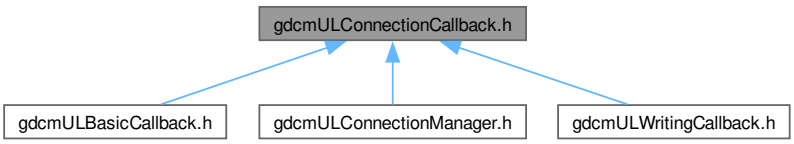
13.589 gdcmULConnectionCallback.h File Reference

`#include "gdcmTypes.h"`

Include dependency graph for gdcmULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULConnectionCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

13.590 gdcmULConnectionCallback.h

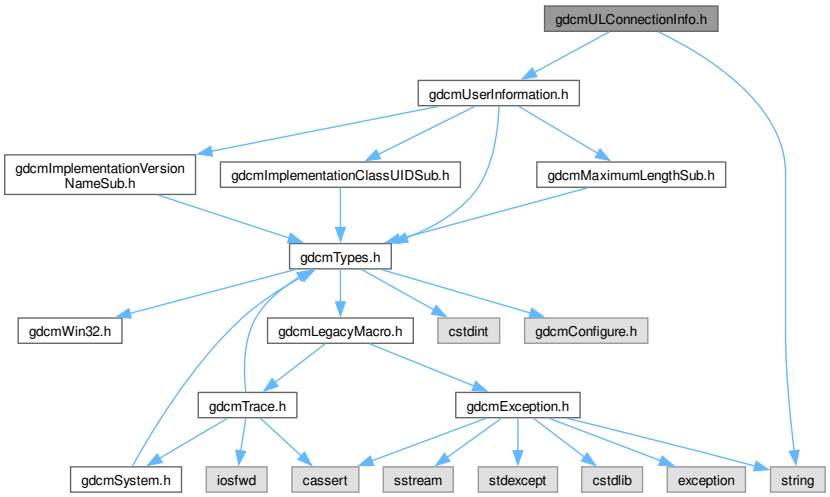
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULCONNECTIONCALLBACK_H
#define GDCMULCONNECTIONCALLBACK_H

#include "gdcmTypes.h" //to be able to export the class

namespace gdcm
{
  class DataSet;
  namespace network
  {
    class GDCM_EXPORT ULConnectionCallback {
      bool mHandledDataSet;
    protected:
      bool mImplicit;
      //inherited callbacks MUST call this function for the cmove loop to work properly
      void DataSetHandled() { mHandledDataSet = true; }
    public:
      ULConnectionCallback():mHandledDataSet(false),mImplicit(true){}
      virtual ~ULConnectionCallback() = default; //placeholder for inherited objects
      virtual void HandleDataSet(const DataSet& inDataSet) = 0;
      virtual void HandleResponse(const DataSet& inDataSet) = 0;

      bool DataSetHandles() const { return mHandledDataSet; }
      void ResetHandledDataSet() { mHandledDataSet = false; }

      void SetImplicitFlag( const bool imp ) { mImplicit = imp; }
    };
  }
}
#endif //GDCMULCONNECTIONCALLBACK_H
```

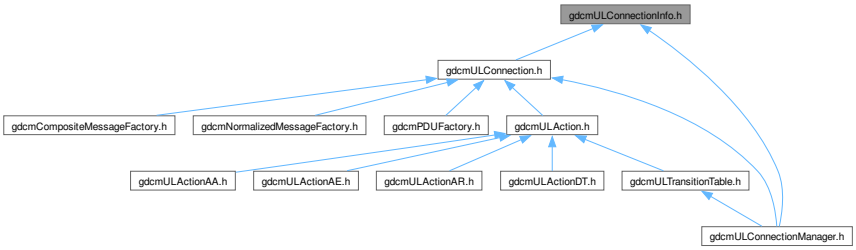
13.591 gdcmULConnectionInfo.h File Reference

```
#include "gdcmUserInformation.h"
#include <string>
```

Include dependency graph for gdcmULConnectionInfo.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULConnectionInfo

  ULConnectionInfo.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.592 gdcmULConnectionInfo.h

Go to the documentation of this file.

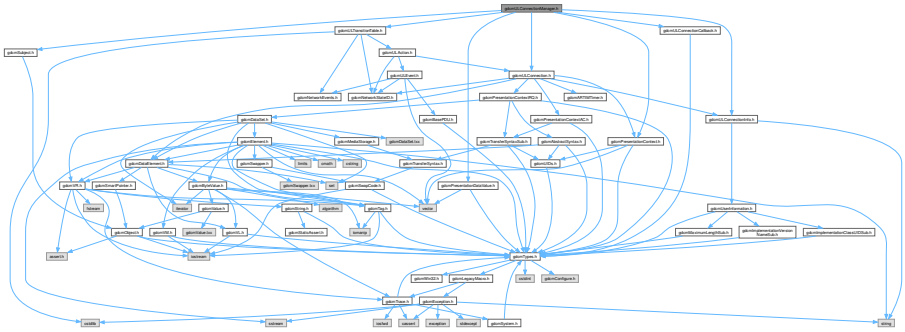
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULCONNECTIONINFO_H
#define GDCMULCONNECTIONINFO_H

#include "gdcmUserInformation.h"
#include <string>

namespace gdcm{
  namespace network {
class ULConnectionInfo {
      UserInformation mUserInformation;

      std::string mCalledAETitle;
      std::string mCallingAETitle;

      unsigned long mCalledIPAddress;
      int mCalledIPPort;
      std::string mCalledComputerName; //either the IP or the name has to be filled in

      unsigned long mMaxPDULength;
    public:
      ULConnectionInfo();

      //it is possible to misinitialize this object, so
      //have it return false if something breaks (ie, given AEs are bigger than 16 characters,
      //no name or IP address).
      bool Initialize(UserInformation const &inUserInformation,
        const char *inCalledAETitle, const char *inCallingAETitle,
        unsigned long inCalledIPAddress, int inCalledIPPort,
        std::string inCalledComputerName);

      //UserInformation GetUserInformation() const;
      const char* GetCalledAETitle() const;
      const char* GetCallingAETitle() const;

      unsigned long GetCalledIPAddress() const;
      int GetCalledIPPort() const;
      std::string GetCalledComputerName() const;

      //CStore needs to know the max pdu length, so the value gets initialized
      //when a cstore connection is established (but not for the others).
      void SetMaxPDULength(unsigned long inMaxPDULength);
      unsigned long GetMaxPDULength() const;
    };
  }
}

#endif //GDCMULCONNECTIONINFO_H
```

13.593 gdcmULConnectionManager.h File Reference

```
#include "gdcmULTransitionTable.h"
#include "gdcmULConnection.h"
```

```
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnectionCallback.h"
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmULConnectionManager.h:



Classes

- class gdcm::network::ULConnectionManager

  ULConnectionManager.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.594 gdcmULConnectionManager.h

Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULCONNECTIONMANAGER_H
#define GDCMULCONNECTIONMANAGER_H

#include "gdcmULTransitionTable.h"
#include "gdcmULConnection.h"
#include "gdcmULConnectionInfo.h"
```

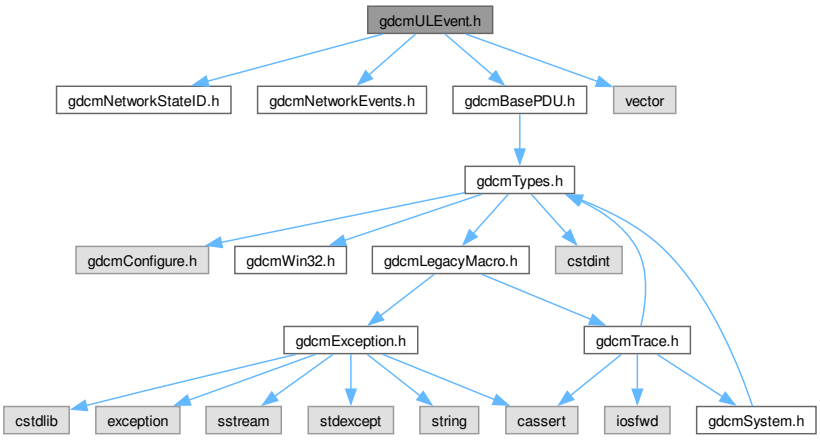
```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnectionCallback.h"
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"

namespace gdcm {
  class File;
  class BaseRootQuery;
  class BaseQuery;

  namespace network {

  class GDCM_EXPORT ULConnectionManager : public Subject
  {
    protected:
      ULConnection* mConnection;
      ULConnection* mSecondaryConnection;
      ULTransitionTable mTransitions;

      //no copying
      ULConnectionManager(const ULConnectionManager& inCM);

      //event handler loop.
      //will just keep running until the current event is nonexistent.
      //at which point, it will return the current state of the connection
      //this starts by initiating an action, but can be put into a passive mode
      //for a cmove/cstore combination by setting startWaiting to true
      EStateID RunEventLoop(ULEvent& inEvent, ULConnection* inWhichConnection,
        ULConnectionCallback* inCallback, const bool& startWaiting);

      //like the above, but will manage the event loop for a move event (which
      //is basically two simultaneous connections interwoven, one inbound and
      //the other outbound.  Note, for instance, that cmoversp's can be sent back
      //during the other connection's operation.
      EStateID RunMoveEventLoop(ULEvent& inEvent, ULConnectionCallback* inCallback);

    public:
      ULConnectionManager();
      ~ULConnectionManager() override;

      // NOTE: (MM) The following two functions are difficult to use, therefore marking
      // them as internal for now.

      // \internal
      bool EstablishConnection(const std::string& inAETitle,
        const std::string& inConnectAETitle,
        const std::string& inComputerName, long inIPAddress,
        uint16_t inConnectPort, double inTimeout,
        std::vector<PresentationContext> const & pcVector );

      bool EstablishConnectionMove(const std::string& inAETitle,
        const std::string& inConnectAETitle,
        const std::string& inComputerName, long inIPAddress,
        uint16_t inConnectPort, double inTimeout,
        uint16_t inReturnPort,
        std::vector<PresentationContext> const & pcVector);
      // \endinternal


      //bool ReestablishConnection(const EConnectionType& inConnectionType,
      //  const DataSet& inDS);

      //allows for a connection to be broken, but waits for an acknowledgement
      //of the breaking for a certain amount of time.  Returns true of the
      //other side acknowledges the break
      bool BreakConnection(const double& inTimeout);

      //severs the connection, if it's open, without waiting for any kind of response.
      //typically done if the program is going down.
      void BreakConnectionNow();

      //This function will send a given piece of data
      //across the network connection.  It will return true if the
      //sending worked, false otherwise.
      //note that sending is asynchronous; as such, there's
      //also a 'receive' option, but that requires a callback function.
      //bool SendData();

      //send the Data PDU associated with Echo (ie, a default DataPDU)
      //this lets the user confirm that the connection is alive.
      //the user should look to cout to see the response of the echo command
```

```
      //returns the PresentationDataValue that was returned by the remote
      //host.  Note that the PDV can be uninitialized, which would indicate failure.
      //Echo does not use a callback for results.
      std::vector<PresentationDataValue> SendEcho();

      // \internal
      // API will change...
      std::vector<DataSet> SendStore(const File &file, std::istream * pStream = nullptr, std::streampos dataSetOffset = 0  );
      std::vector<DataSet> SendFind(const BaseRootQuery* inRootQuery);
      std::vector<DataSet> SendMove(const BaseRootQuery* inRootQuery);

    std::vector<DataSet> SendNEventReport (const BaseQuery* inQuery);
      std::vector<DataSet> SendNGet     (const BaseQuery* inQuery);
      std::vector<DataSet> SendNSet     (const BaseQuery* inQuery);
      std::vector<DataSet> SendNAction    (const BaseQuery* inQuery);
      std::vector<DataSet> SendNCreate    (const BaseQuery* inQuery);
      std::vector<DataSet> SendNDelete    (const BaseQuery* inQuery);
      // \endinternal

      void SendStore(const File & file, ULConnectionCallback* inCallback, std::istream * pStream = nullptr, std::streampos dataSetOffset = 0 );
      void SendFind(const BaseRootQuery* inRootQuery, ULConnectionCallback* inCallback);
      bool SendMove(const BaseRootQuery* inRootQuery, ULConnectionCallback* inCallback);

      void SendNEventReport (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
      void SendNGet     (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
      void SendNSet     (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
      void SendNAction    (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
      void SendNCreate    (const BaseQuery* inQuery, ULConnectionCallback* inCallback);
      void SendNDelete    (const BaseQuery* inQuery, ULConnectionCallback* inCallback);

    };
  }
}

#endif // GDCMULCONNECTIONMANAGER_H
```

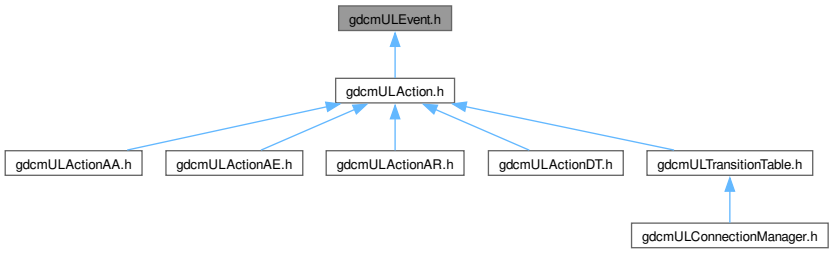
13.595 gdcmULEvent.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmBasePDU.h"
#include <vector>
```

Include dependency graph for gdcmULEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ULEvent

  ULEvent.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.596 gdcmULEvent.h

Go to the documentation of this file.

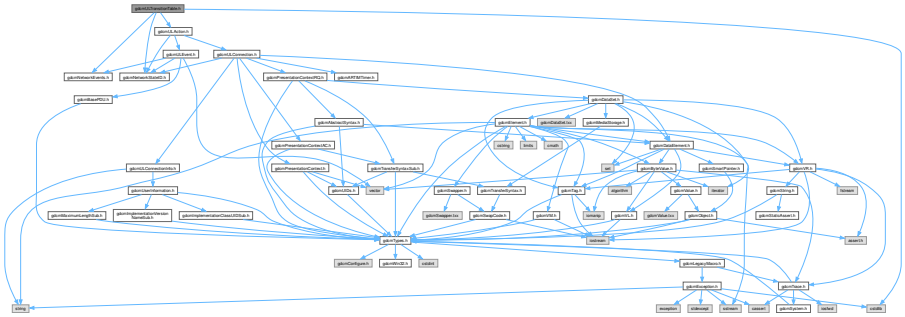
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULEVENT_H
#define GDCMULEVENT_H

#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmBasePDU.h"
#include <vector>

namespace gdcm {
  namespace network {

class ULEvent {
      EEventID mEvent;
      std::vector<BasePDU*> mBasePDU;
```

```
    std::istream * m_pStream ;
    std::streampos m_posDataSet ;
      void DeletePDUVector(){
        std::vector<BasePDU*>::iterator baseItor;
        for (baseItor = mBasePDU.begin(); baseItor < mBasePDU.end(); baseItor++){
          if (*baseItor != nullptr){
            delete *baseItor;
            *baseItor = nullptr;
          }
        }
      }

    public:
      ULEvent(const EEventID& inEventID, std::vector<BasePDU*> inBasePDU, std::istream * iStream = nullptr, std::streampos posDataSet = 0 ){
        mEvent = inEventID;
        mBasePDU = inBasePDU;
    m_pStream = iStream ;
    m_posDataSet = posDataSet ;
      }
      ULEvent(const EEventID& inEventID, BasePDU* inBasePDU, std::istream * iStream = nullptr, std::streampos posDataSet = 0 ){
        mEvent = inEventID;
        mBasePDU.push_back(inBasePDU);
    m_pStream = iStream ;
    m_posDataSet = posDataSet ;
      }
      ~ULEvent(){
        DeletePDUVector();
      }

      EEventID GetEvent() const { return mEvent; }
      std::vector<BasePDU*> const & GetPDUs() const { return mBasePDU; }
    std::istream * GetIStream() const { return m_pStream; }
    std::streampos GetDataSetPos() const { return m_posDataSet; }

      void SetEvent(const EEventID& inEvent) { mEvent = inEvent; }
      void SetPDU(std::vector<BasePDU*> const & inPDU) {
        DeletePDUVector();
        mBasePDU = inPDU;
      }
    };
  }
}

#endif //GDCMULEVENT_H
```

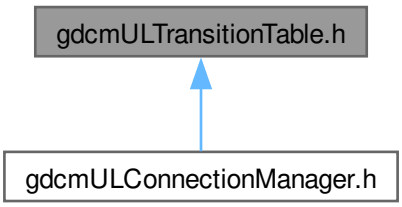
13.597 gdcmULTransitionTable.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULAction.h"
#include <cstdlib>
```

Include dependency graph for gdcmULTransitionTable.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::TableRow
- struct gdcm::network::Transition
- class gdcm::network::ULTransitionTable

  ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.598 gdcmULTransitionTable.h

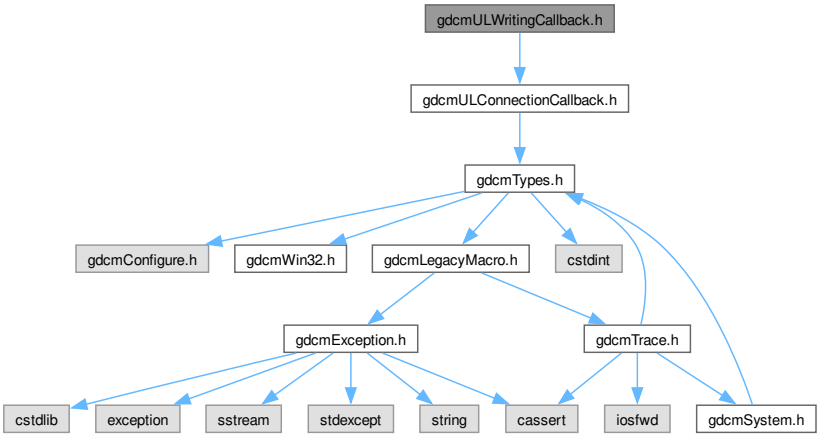
Go to the documentation of this file.

```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULTRANSITIONTABLE_H
#define GDCMULTRANSITIONTABLE_H

#include "gdcmNetworkStateID.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULAction.h"

#include <cstdlib>  // NULL
```

```

namespace gdcm {
class Subject;
  namespace network{
class ULConnection;
class ULAction;
class ULEvent;

    //The transition dictates the action that should be taken from the start state to the end state
    struct Transition {
      int mEnd;
      ULAction* mAction;
      Transition(){
        mEnd = eStaDoesNotExist;
        mAction = nullptr;
      }
      ~Transition(){
        if (mAction != nullptr){
          delete mAction;
          mAction = nullptr;
        }
      }
      Transition(int inEndState, ULAction* inAction){
        mEnd = inEndState;
        mAction = inAction;
      }
      static Transition* MakeNew(int inEndState, ULAction* inAction){
        return new Transition(inEndState, inAction);
      }
    };

    //used to define a row in table 9-10 of 3.8 2009
    //the transition table is events, then state,
    //then the transition itself (which has the event
    //and start state implied by their starting locations)
    //don't need to store the event; that's implicitly defined in the Table itself by location
    class TableRow{
    public:
      TableRow() {
        for(int stateIndex = 0; stateIndex < cMaxStateID; ++stateIndex)
          {
          transitions[stateIndex] = nullptr;
          }
      }
      ~TableRow() {
        for(int stateIndex = 0; stateIndex < cMaxStateID; ++stateIndex)
          {
          Transition *t = transitions[stateIndex];
          delete t;
          }
       }
      Transition *transitions[cMaxStateID];

      //copy constructor for stl additions into the transition table below.
    };

class ULTransitionTable
{
    private:
      TableRow mTable[cMaxEventID];
    public:
      ULTransitionTable();

      void HandleEvent(Subject*s,ULEvent& inEvent, ULConnection& inConnection,
        bool& outWaitingForEvent, EEventID& outRaisedEvent) const;

      void PrintTable() const; //so that the table can be printed and verified against the DICOM standard
    };
  }
}
#endif // GDCMULTRANSITIONTABLE_H
```

13.599 gdcmULWritingCallback.h File Reference

`#include "gdcmULConnectionCallback.h"`
Include dependency graph for gdcmULWritingCallback.h:



Classes

- class gdcm::network::ULWritingCallback

Namespaces

- namespace gdcm
- namespace gdcm::network

13.600 gdcmULWritingCallback.h

Go to the documentation of this file.

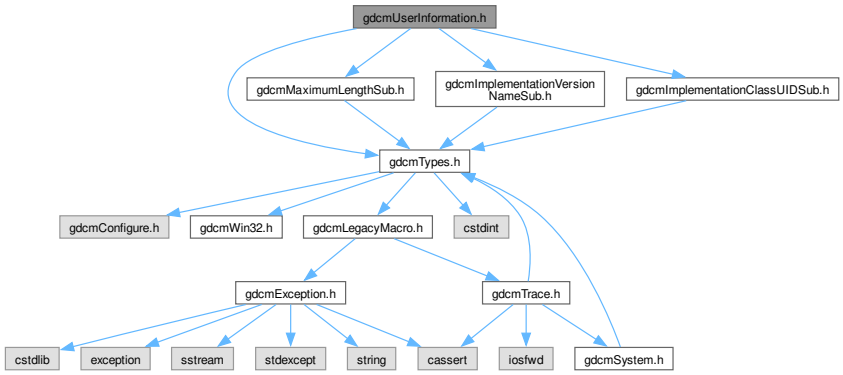
```
/*=========================================================================
 *
 *  Copyright NumFOCUS
 *
 *  Licensed under the Apache License, Version 2.0 (the "License");
 *  you may not use this file except in compliance with the License.
 *  You may obtain a copy of the License at
 *
 *         http://www.apache.org/licenses/LICENSE-2.0.txt
 *
 *  Unless required by applicable law or agreed to in writing, software
 *  distributed under the License is distributed on an "AS IS" BASIS,
 *  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *  See the License for the specific language governing permissions and
 *  limitations under the License.
 *
 *=========================================================================*/
#ifndef GDCMULCONNECTIONWRITINGCALLBACK_H
```

```
#define GDCMULCONNECTIONWRITINGCALLBACK_H

#include "gdcmULConnectionCallback.h"

namespace gdcm
{
class DataSet;
namespace network
{
/* \brief ULWritingCallback
 * \details This is the most basic of callbacks for how the ULConnectionManager handles
 * incoming datasets.  DataSets are immediately written to disk as soon as they
 * are received.  NOTE that if the incoming connection is faster than the disk
 * writing speed, this callback could cause some pileups!
 */
class GDCM_EXPORT ULWritingCallback : public ULConnectionCallback
{
  std::string mDirectoryName;
public:
  ULWritingCallback() = default;
  ~ULWritingCallback() override = default; //empty, for later inheritance

  void SetDirectory(const std::string& inDirectoryName) { mDirectoryName = inDirectoryName; }

  void HandleDataSet(const DataSet& inDataSet) override;
  void HandleResponse(const DataSet& inDataSet) override;
};
} // end namespace network
} // end namespace gdcm

#endif //GDCMULCONNECTIONWRITINGCALLBACK_H
```

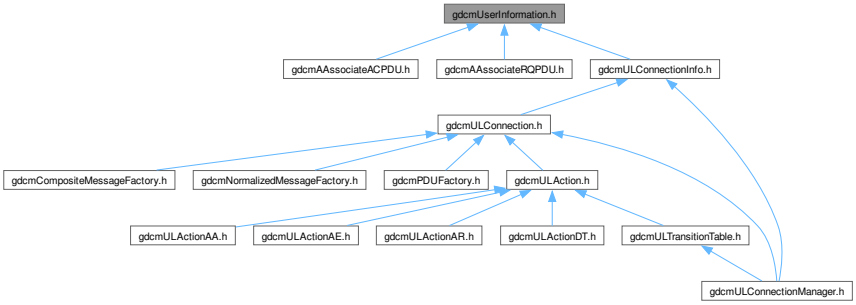
13.601 gdcmUserInformation.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmMaximumLengthSub.h"
#include "gdcmImplementationVersionNameSub.h"
#include "gdcmImplementationClassUIDSub.h"
```

Include dependency graph for gdcmUserInformation.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::UserInformation

  UserInformation.

Namespaces

- namespace gdcm
- namespace gdcm::network

13.602 gdcmUserInformation.h

Go to the documentation of this file.

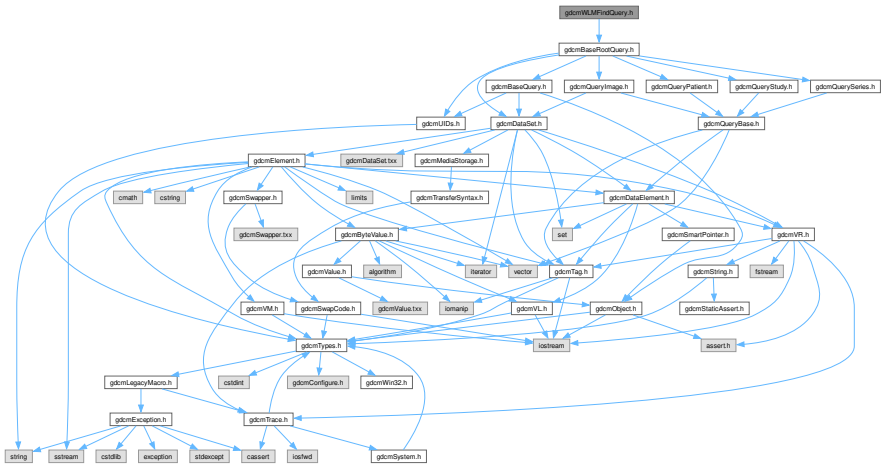
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMUSERINFORMATION_H
#define GDCMUSERINFORMATION_H

#include "gdcmTypes.h"
#include "gdcmMaximumLengthSub.h"
#include "gdcmImplementationVersionNameSub.h"
#include "gdcmImplementationClassUIDSub.h"

namespace gdcm
{

namespace network
{

class AsynchronousOperationsWindowSub;
class RoleSelectionSub;
struct RoleSelectionSubItems;
```

```
class SOPClassExtendedNegociationSub;
struct SOPClassExtendedNegociationSubItems;
class UserInformation
{
public:
  UserInformation();
  ~UserInformation();
  std::istream &Read(std::istream &is);
  const std::ostream &Write(std::ostream &os) const;
  size_t Size() const;

  void Print(std::ostream &os) const;

  const MaximumLengthSub &GetMaximumLengthSub() const { return MLS; }
  MaximumLengthSub &GetMaximumLengthSub() { return MLS; }

  void AddRoleSelectionSub( RoleSelectionSub const & r );
  void AddSOPClassExtendedNegociationSub( SOPClassExtendedNegociationSub const & s );

private:
  static const uint8_t ItemType;
  static const uint8_t Reserved2;
  uint16_t ItemLength; // len of
  MaximumLengthSub MLS;
  ImplementationClassUIDSub ICUID;
  AsynchronousOperationsWindowSub *AOWS;
  RoleSelectionSubItems *RSSI;
  SOPClassExtendedNegociationSubItems *SOPCENSI;
  ImplementationVersionNameSub IVNS;

public:
  UserInformation(const UserInformation&) = delete;
  UserInformation &operator=(const UserInformation&);
};

} // end namespace network

} // end namespace gdcm

#endif //GDCMUSERINFORMATION_H
```

13.603 gdcmWLMFindQuery.h File Reference

`#include "gdcmBaseRootQuery.h"`

Include dependency graph for gdcmWLMFindQuery.h:



Classes

- class gdcm::WLMFindQuery

  PatientRootQuery.

Namespaces

- namespace gdcm

13.604 gdcmWLMFindQuery.h

Go to the documentation of this file.

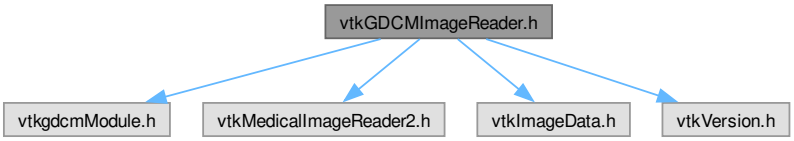
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMWLMFindQuery_H
#define GDCMWLMFindQuery_H

#include "gdcmBaseRootQuery.h"

namespace gdcm
{
class GDCM_EXPORT WLMFindQuery : public BaseRootQuery
{
  friend class QueryFactory;
public:
  WLMFindQuery();

  // no sense here
  void InitializeDataSet(const EQueryLevel& inQueryLevel) override;
  std::vector<Tag> GetTagListByLevel(const EQueryLevel& inQueryLevel) override;
  // validate query has required tag
  bool ValidateQuery(bool inStrict = true) const override;

  UIDs::TSName GetAbstractSyntaxUID() const override;
protected :
  DataSet GetValidDataSet() const;
};

} // end namespace gdcm

#endif // GDCMWLMFindQuery_H
```

13.605 vtkGDCMImageReader.h File Reference

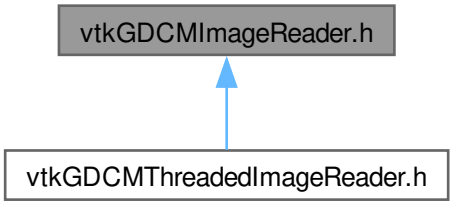
```
#include "vtkgdcmModule.h"
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
```

```
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vtkGDCMImageReader

Namespaces

- namespace gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

13.605.1 Macro Definition Documentation

13.605.1.1 VTK_CMYK

```
#define VTK_CMYK 8
```

13.605.1.2 VTK_INVERSE_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

13.605.1.3 VTK_LOOKUP_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

13.605.1.4 VTK_YBR

```
#define VTK_YBR 7
```

13.606 vtkGDCMImageReader.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkGDCMImageReader - read DICOM Image files (Pixel Data)
// .SECTION Description
// vtkGDCMImageReader is a source object that reads some DICOM files
// this reader is single threaded.
// .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
// upside down as VTK would expect, use this option only if you know what you are doing.
// .SECTION Implementation note: when reading a series of 2D slices, user is
// expected to provide an ordered list of filenames. No sorting will be applied afterward.
// .SECTION Implementation note: Although 99% of the time the Zspacing as read
// from a tag in a 2D DICOM file should be correct, there has been reports that this
// value can be missing, or incorrect, in which case users are advised to override this
// value using the return value from gdcm::IPPSorter::GetZSpacing() and set it via
// vtkImageChangeInformation on the reader itself.
// .SECTION TODO
// This reader does not handle a series of 3D images, only a single 3D (multi frame) or a
// list of 2D files are supported for now.
// .SECTION TODO
// Did not implement SetFilePattern / SetFilePrefix API, move it to protected section for now.
// .SECTION BUG
// Overlay are assumed to have the same extent as image. Right now if overlay origin is not
// 0,0 the overlay will have an offset...
// Only the very first overlay is loaded at the VTK level, for now (even if there are more than one in the
file)
```

```
// .SECTION DataOrigin
// When the reader is instantiated with FileLowerLeftOn the DataOrigin and Image Position (Patient) are
// identical. But when FileLowerLeft is Off, we have to reorder the Y-line of the image, and thus the DataOrigin
// is then translated to the other side of the image.
// .SECTION Spacing
// When reading a 3D volume, the spacing along the Z dimension might be negative (so as to respect up-side-down)
// as specified in the Image Orientation (Patient) tag. When Z-spacing is 0, this means the multi-frame object
// contains image which do not represent uniform volume.
// .SECTION Warning
// When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader
// it is *required* that FileLowerLeft is set to ON as coordinate system
// would be inconsistent in between the two data structures.
// .SECTION Color Space mapping:
// * VTK_LUMINANCE         <-> MONOCHROME2
// * VTK_LUMINANCE_ALPHA   <-> Not supported
// * VTK_RGB               <-> RGB
// * VTK_RGBA              <-> ARGB (deprecated, DICOM 2008)
// * VTK_INVERSE_LUMINANCE <-> MONOCHROME1
// * VTK_LOOKUP_TABLE      <-> PALETTE COLOR
// * VTK_YBR               <-> YBR_FULL
//
// For detailed information on color space transformation and true lossless transformation see:
// http://gdcm.sourceforge.net/wiki/index.php/Color_Space_Transformations

// .SECTION See Also
// vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMPolyDataReader vtkGDCMImageWriter
// vtkDICOMImageReader

#ifndef VTKGDCMIMAGEREADER_H
#define VTKGDCMIMAGEREADER_H

#include "vtkgdcmModule.h"
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
#include "vtkVersion.h"

#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
#else
class vtkMedicalImageProperties;
#endif
#if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
#else
class vtkStringArray;
#endif
class vtkPolyData;

// vtkSystemIncludes.h defines:
// #define VTK_LUMINANCE       1
// #define VTK_LUMINANCE_ALPHA 2
// #define VTK_RGB             3
// #define VTK_RGBA            4
#ifndef VTK_INVERSE_LUMINANCE
#define VTK_INVERSE_LUMINANCE 5
#endif
#ifndef VTK_LOOKUP_TABLE
#define VTK_LOOKUP_TABLE 6
#endif
#ifndef VTK_YBR
#define VTK_YBR 7
#endif
#ifndef VTK_CMYK
#define VTK_CMYK 8
#endif

//BTX
namespace gdcm { class ImageReader; }
//ETX
class vtkMatrix4x4;
class VTKGDCM_EXPORT vtkGDCMImageReader : public vtkMedicalImageReader2
{
public:
  static vtkGDCMImageReader *New();
  vtkTypeMacro(vtkGDCMImageReader,vtkMedicalImageReader2);
  virtual void PrintSelf(ostream& os, vtkIndent indent);

  // Description: is the given file name a DICOM file containing an image ?
  virtual int CanReadFile(const char* fname);

```

```
  // Description:
  // Valid extensions
  virtual const char* GetFileExtensions()
    {
    // I would like to get rid of ACR/NEMA/IMA so only allow dcm extension for now
    return ".dcm .DCM";
    }

  // Description:
  // A descriptive name for this format
  virtual const char* GetDescriptiveName()
    {
    return "DICOM";
    }

  // Description:
  // Get the Image Position (Patient) as stored in the DICOM file
  // This is a read-only data member
  vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);

#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
#else
  // Description:
  // Get the medical image properties object
  vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
#endif
  virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);

#if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
#else
  virtual void SetFileNames(vtkStringArray*);
  vtkGetObjectMacro(FileNames, vtkStringArray);
#endif

  // Description:
  // Specifically request to load the overlay into the gdcm-VTK layer (gdcm always loads them when found).
  // If no overlay is found in the image, then the vtkImageData for the overlay will be empty.
  vtkGetMacro(LoadOverlays,int);
  vtkSetMacro(LoadOverlays,int);
  vtkBooleanMacro(LoadOverlays,int);

  // Description:
  // Set/Get whether or not to load the Icon as vtkImageData (if found in the DICOM file)
  vtkGetMacro(LoadIconImage,int);
  vtkSetMacro(LoadIconImage,int);
  vtkBooleanMacro(LoadIconImage,int);

  // Description:
  // Set/Get whether or not the image was compressed using a lossy compression algorithm
  vtkGetMacro(LossyFlag,int);
  vtkSetMacro(LossyFlag,int);
  vtkBooleanMacro(LossyFlag,int);

  // Description:
  // Read only: number of overlays as found in this image (multiple overlays per slice is allowed)
  // Only valid when LoadOverlays is true
  vtkGetMacro(NumberOfOverlays,int);

  // Description:
  // Read only: number of icon image (there can only be zero or one icon per file)
  // Only valid when LoadIconImage is true
  vtkGetMacro(NumberOfIconImages,int);

  // Description:
  // Get Overlay/IconImage
  // Remember to ALWAYS use those methods in your code, as the internal number for the output port
  // is not guaranteed to remain the same, as features are added to the reader
#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
//FIXME: Need to get rid of BTX/ETX if only the Python Wrapper of VTK 4.2 would let me
//BTX
  vtkAlgorithmOutput* GetOverlayPort(int index);
  vtkAlgorithmOutput* GetIconImagePort();
//ETX
#endif
  vtkImageData* GetOverlay(int i);
  vtkImageData* GetIconImage();

  // Description:
  // Load image with its associated Lookup Table
  vtkGetMacro(ApplyLookupTable,int);
  vtkSetMacro(ApplyLookupTable,int);
```

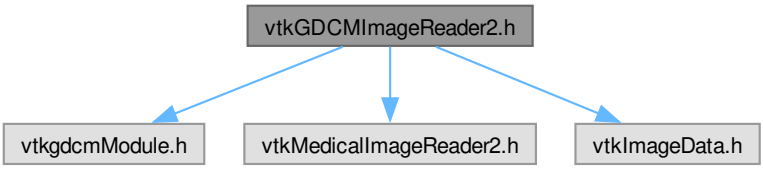
```
  vtkBooleanMacro(ApplyLookupTable,int);

  // Description:
  // Load image as YBR
  vtkGetMacro(ApplyYBRToRGB,int)
  vtkSetMacro(ApplyYBRToRGB,int)
  vtkBooleanMacro(ApplyYBRToRGB,int);

  // Description:
  // Return VTK_LUMINANCE, VTK_INVERSE_LUMINANCE, VTK_RGB, VTK_RGBA, VTK_LOOKUP_TABLE, VTK_YBR or VTK_CMYK
  // or 0 when ImageFormat is not handled.
  // Warning: For color image, PlanarConfiguration need to be taken into account.
  vtkGetMacro(ImageFormat,int);

  // Description:
  // Return the Planar Configuration. This simply means that the internal DICOM image was stored
  // using a particular planar configuration (most of the time: 0)
  // For monochrome image, PlanarConfiguration is always 0
  vtkGetMacro(PlanarConfiguration,int);

  // Description:
  // Return the 'raw' information stored in the DICOM file:
  // In case of a series of multiple files, only the first file is considered. The Image Orientation (Patient)
  // is guaranteed to remain the same, and image Image Position (Patient) in other slice can be computed
  // using the ZSpacing (3rd dimension)
  // (0020,0032) DS [87.774866\-182.908510\168.629671]       #  32, 3 ImagePositionPatient
  // (0020,0037) DS [0.001479\0.999989\-0.004376\-0.002039\-0.004372\-0.999988] #  58, 6 ImageOrientationPatient
  vtkGetVector3Macro(ImagePositionPatient,double);
  vtkGetVector6Macro(ImageOrientationPatient,double);

  // Description:
  // Set/Get the first Curve Data:
  vtkGetObjectMacro(Curve,vtkPolyData);
  virtual void SetCurve(vtkPolyData *pd);

  // Description:
  // \DEPRECATED:
  // Modality LUT
  // Value returned by GetShift/GetScale might be inaccurate since Shift/Scale could be
  // varying along the Series read. Therefore user are advices not to use those functions
  // anymore
  vtkGetMacro(Shift,double);
  vtkGetMacro(Scale,double);

protected:
  vtkGDCMImageReader();
  ~vtkGDCMImageReader();

  vtkSetVector6Macro(ImageOrientationPatient,double);

//BTX
  void FillMedicalImageInformation(const gdcm::ImageReader &reader);
//ETX
  int RequestInformationCompat();
  int RequestDataCompat();

#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
  int ProcessRequest(vtkInformation* request,
                              vtkInformationVector** inputVector,
                              vtkInformationVector* outputVector);
  int RequestInformation(vtkInformation *request,
                         vtkInformationVector **inputVector,
                         vtkInformationVector *outputVector);
  int RequestData(vtkInformation *request,
                  vtkInformationVector **inputVector,
                  vtkInformationVector *outputVector);
#else /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
  void ExecuteInformation();
  void ExecuteData(vtkDataObject *out);
#endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/

protected:
#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
#else
  // Description:
  // Medical Image properties
  vtkMedicalImageProperties *MedicalImageProperties;
#endif
#if (VTK_MAJOR_VERSION > 5) || ( VTK_MAJOR_VERSION == 5 && VTK_MINOR_VERSION > 0 )
```

```
#else
  vtkStringArray *FileNames;
#endif

  vtkMatrix4x4 *DirectionCosines;
  int LoadOverlays;
  int NumberOfOverlays;
  int LoadIconImage;
  int NumberOfIconImages;
  int IconImageDataExtent[6];
  double ImagePositionPatient[3];
  double ImageOrientationPatient[6];
  vtkPolyData *Curve;

  int ImageFormat;
  // the following 3, should remain optional
  int ApplyInverseVideo;
  int ApplyLookupTable;
  int ApplyYBRToRGB;
  // I think that planar configuration need to always be applied as far as VTK is concerned
  int ApplyPlanarConfiguration;
  int ApplyShiftScale;

  int LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen);

  double Shift;
  double Scale;
  int IconDataScalarType;
  int IconNumberOfScalarComponents;
  int PlanarConfiguration;
  int LossyFlag;
  int ForceRescale;

protected:
  // TODO / FIXME
  void SetFilePrefix(const char *) {}
  vtkGetStringMacro(FilePrefix);
  void SetFilePattern(const char *) {}
  vtkGetStringMacro(FilePattern);

private:
  vtkGDCMImageReader(const vtkGDCMImageReader&);  // Not implemented.
  void operator=(const vtkGDCMImageReader&);  // Not implemented.
};
#endif
```

13.607 vtkGDCMImageReader2.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
```

Include dependency graph for vtkGDCMImageReader2.h:



Classes

- class vtkGDCMImageReader2

Namespaces

- namespace gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

13.607.1 Macro Definition Documentation

13.607.1.1 VTK_CMYK

```
#define VTK_CMYK 8
```

13.607.1.2 VTK_INVERSE_LUMINANCE

```
#define VTK_INVERSE_LUMINANCE 5
```

13.607.1.3 VTK_LOOKUP_TABLE

```
#define VTK_LOOKUP_TABLE 6
```

13.607.1.4 VTK_YBR

```
#define VTK_YBR 7
```

13.608 vtkGDCMImageReader2.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkGDCMImageReader2 - read DICOM Image files (Pixel Data)
// .SECTION Description
// vtkGDCMImageReader2 is a source object that reads some DICOM files
// this reader is single threaded.
// .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
// upside down as VTK would expect, use this option only if you know what you are doing.
// .SECTION Implementation note: when reading a series of 2D slices, user is
// expected to provide an ordered list of filenames. No sorting will be applied afterward.
// .SECTION Implementation note: Although 99% of the time the Zspacing as read
// from a tag in a 2D DICOM file should be correct, there has been reports that this
// value can be missing, or incorrect, in which case users are advised to override this
// value using the return value from gdcm::IPPSorter::GetZSpacing() and set it via
// vtkImageChangeInformation on the reader itself.
// .SECTION TODO
// This reader does not handle a series of 3D images, only a single 3D (multi frame) or a
// list of 2D files are supported for now.
// .SECTION TODO
// Did not implement SetFilePattern / SetFilePrefix API, move it to protected section for now.
// .SECTION BUG
// Overlay are assumed to have the same extent as image. Right now if overlay origin is not
// 0,0 the overlay will have an offset...
// Only the very first overlay is loaded at the VTK level, for now (even if there are more than one in the file)
// .SECTION DataOrigin
// When the reader is instantiated with FileLowerLeftOn the DataOrigin and Image Position (Patient) are
// identical. But when FileLowerLeft is Off, we have to reorder the Y-line of the image, and thus the DataOrigin
// is then translated to the other side of the image.
// .SECTION Spacing
// When reading a 3D volume, the spacing along the Z dimension might be negative (so as to respect up-side-down)
// as specified in the Image Orientation (Patient) tag. When Z-spacing is 0, this means the multi-frame object
// contains image which do not represent uniform volume.
// .SECTION Warning
// When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader2
// it is *required* that FileLowerLeft is set to ON as coordinate system
// would be inconsistent in between the two data structures.
// .SECTION Color Space mapping:
// * VTK_LUMINANCE         <-> MONOCHROME2
// * VTK_LUMINANCE_ALPHA   <-> Not supported
// * VTK_RGB               <-> RGB
// * VTK_RGBA              <-> ARGB (deprecated, DICOM 2008)
// * VTK_INVERSE_LUMINANCE <-> MONOCHROME1
// * VTK_LOOKUP_TABLE      <-> PALETTE COLOR
// * VTK_YBR               <-> YBR_FULL
//
// For detailed information on color space transformation and true lossless transformation see:
// http://gdcm.sourceforge.net/wiki/index.php/Color_Space_Transformations

// .SECTION See Also
// vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMPolyDataReader vtkGDCMImageWriter
// vtkDICOMImageReader

#ifndef VTKGDCMIMAGEREADER2_H
#define VTKGDCMIMAGEREADER2_H

#include "vtkgdcmModule.h"
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"

class vtkPolyData;
```

```

// vtkSystemIncludes.h defines:
// #define VTK_LUMINANCE       1
// #define VTK_LUMINANCE_ALPHA 2
// #define VTK_RGB             3
// #define VTK_RGBA            4
#ifndef VTK_INVERSE_LUMINANCE
#define VTK_INVERSE_LUMINANCE 5
#endif
#ifndef VTK_LOOKUP_TABLE
#define VTK_LOOKUP_TABLE 6
#endif
#ifndef VTK_YBR
#define VTK_YBR 7
#endif
#ifndef VTK_CMYK
#define VTK_CMYK 8
#endif

//BTX
namespace gdcm { class ImageReader; }
//ETX
class vtkMatrix4x4;
class VTKGDCM_EXPORT vtkGDCMImageReader2 : public vtkMedicalImageReader2
{
public:
  static vtkGDCMImageReader2 *New();
  vtkTypeMacro(vtkGDCMImageReader2,vtkMedicalImageReader2);
  virtual void PrintSelf(ostream& os, vtkIndent indent);

  // Description: is the given file name a DICOM file containing an image ?
  virtual int CanReadFile(const char* fname);

  // Description:
  // Valid extensions
  virtual const char* GetFileExtensions()
    {
    // I would like to get rid of ACR/NEMA/IMA so only allow dcm extension for now
    return ".dcm .DCM";
    }

  // Description:
  // A descriptive name for this format
  virtual const char* GetDescriptiveName()
    {
    return "DICOM";
    }

  // Description:
  // Get the Image Position (Patient) as stored in the DICOM file
  // This is a read-only data member
  vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);

  virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);

  // Description:
  // Specifically request to load the overlay into the gdcm-VTK layer (gdcm always loads them when found).
  // If no overlay is found in the image, then the vtkImageData for the overlay will be empty.
  vtkGetMacro(LoadOverlays,int);
  vtkSetMacro(LoadOverlays,int);
  vtkBooleanMacro(LoadOverlays,int);

  // Description:
  // Set/Get whether or not to load the Icon as vtkImageData (if found in the DICOM file)
  vtkGetMacro(LoadIconImage,int);
  vtkSetMacro(LoadIconImage,int);
  vtkBooleanMacro(LoadIconImage,int);

  // Description:
  // Set/Get whether or not the image was compressed using a lossy compression algorithm
  vtkGetMacro(LossyFlag,int);
  vtkSetMacro(LossyFlag,int);
  vtkBooleanMacro(LossyFlag,int);

  // Description:
  // Read only: number of overlays as found in this image (multiple overlays per slice is allowed)
  // Only valid when LoadOverlays is true
  vtkGetMacro(NumberOfOverlays,int);

  // Description:
  // Read only: number of icon image (there can only be zero or one icon per file)
```

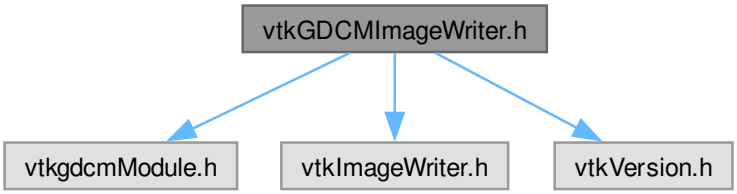
```
  // Only valid when LoadIconImage is true
  vtkGetMacro(NumberOfIconImages,int);

  // Description:
  // Get Overlay/IconImage
  // Remember to ALWAYS use those methods in your code, as the internal number for the output port
  // is not guaranteed to remain the same, as features are added to the reader
  vtkAlgorithmOutput* GetOverlayPort(int index);
  vtkAlgorithmOutput* GetIconImagePort();
  vtkImageData* GetOverlay(int i);
  vtkImageData* GetIconImage();

  // Description:
  // Load image with its associated Lookup Table
  vtkGetMacro(ApplyLookupTable,int);
  vtkSetMacro(ApplyLookupTable,int);
  vtkBooleanMacro(ApplyLookupTable,int);

  // Description:
  // Load image as YBR
  vtkGetMacro(ApplyYBRToRGB,int)
  vtkSetMacro(ApplyYBRToRGB,int)
  vtkBooleanMacro(ApplyYBRToRGB,int);

  // Description:
  // Return VTK_LUMINANCE, VTK_INVERSE_LUMINANCE, VTK_RGB, VTK_RGBA, VTK_LOOKUP_TABLE, VTK_YBR or VTK_CMYK
  // or 0 when ImageFormat is not handled.
  // Warning: For color image, PlanarConfiguration need to be taken into account.
  vtkGetMacro(ImageFormat,int);

  // Description:
  // Return the Planar Configuration. This simply means that the internal DICOM image was stored
  // using a particular planar configuration (most of the time: 0)
  // For monochrome image, PlanarConfiguration is always 0
  vtkGetMacro(PlanarConfiguration,int);

  // Description:
  // Return the 'raw' information stored in the DICOM file:
  // In case of a series of multiple files, only the first file is considered. The Image Orientation (Patient)
  // is guaranteed to remain the same, and image Image Position (Patient) in other slice can be computed
  // using the ZSpacing (3rd dimension)
  // (0020,0032) DS [87.774866\-182.908510\168.629671]       #  32, 3 ImagePositionPatient
  // (0020,0037) DS [0.001479\0.999989\-0.004376\-0.002039\-0.004372\-0.999988] #  58, 6 ImageOrientationPatient
  vtkGetVector3Macro(ImagePositionPatient,double);
  vtkGetVector6Macro(ImageOrientationPatient,double);

  // Description:
  // Set/Get the first Curve Data:
  vtkGetObjectMacro(Curve,vtkPolyData);
  virtual void SetCurve(vtkPolyData *pd);

  // Description:
  // \DEPRECATED:
  // Modality LUT
  // Value returned by GetShift/GetScale might be inaccurate since Shift/Scale could be
  // varying along the Series read. Therefore user are advices not to use those functions
  // anymore
  vtkGetMacro(Shift,double);
  vtkGetMacro(Scale,double);

protected:
  vtkGDCMImageReader2();
  ~vtkGDCMImageReader2();

  vtkSetVector6Macro(ImageOrientationPatient,double);

//BTX
  void FillMedicalImageInformation(const gdcm::ImageReader &reader);
//ETX
  int RequestInformationCompat();
  int RequestDataCompat();

  int ProcessRequest(vtkInformation* request,
                              vtkInformationVector** inputVector,
                              vtkInformationVector* outputVector);
  int RequestInformation(vtkInformation *request,
                         vtkInformationVector **inputVector,
                         vtkInformationVector *outputVector);
  int RequestData(vtkInformation *request,
```

```
                  vtkInformationVector **inputVector,
                  vtkInformationVector *outputVector);

protected:
  vtkMatrix4x4 *DirectionCosines;
  int LoadOverlays;
  int NumberOfOverlays;
  int LoadIconImage;
  int NumberOfIconImages;
  int IconImageDataExtent[6];
  double ImagePositionPatient[3];
  double ImageOrientationPatient[6];
  vtkPolyData *Curve;

  int ImageFormat;
  // the following 3, should remain optional
  int ApplyInverseVideo;
  int ApplyLookupTable;
  int ApplyYBRToRGB;
  // I think that planar configuration need to always be applied as far as VTK is concerned
  int ApplyPlanarConfiguration;
  int ApplyShiftScale;

  int LoadSingleFile(const char *filename, char *pointer, unsigned long &outlen);

  double Shift;
  double Scale;
  int IconDataScalarType;
  int IconNumberOfScalarComponents;
  int PlanarConfiguration;
  int LossyFlag;
  int ForceRescale;

protected:
  // TODO / FIXME
  void SetFilePrefix(const char *) {}
  vtkGetStringMacro(FilePrefix);
  void SetFilePattern(const char *) {}
  vtkGetStringMacro(FilePattern);

private:
  vtkGDCMImageReader2(const vtkGDCMImageReader2&);  // Not implemented.
  void operator=(const vtkGDCMImageReader2&);  // Not implemented.
};
#endif
```

13.609 vtkGDCMImageWriter.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkImageWriter.h"
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



Classes

- class vtkGDCMImageWriter

13.610 vtkGDCMImageWriter.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkGDCMImageWriter - write DICOM files
// .SECTION Description
// vtkGDCMImageWriter is a sink object that write DICOM files
// this writer is single threaded (see vtkGDCMThreadedImageReader2 for multi-thread)
//
// .SECTION Warning: vtkLookupTable from the vtkImageData object taken into account
// only if ImageFormat is set to VTK_LOOKUP_TABLE
//
// .SECTION NOTE We are not using the usual API SetFilePrefix / SetFilePattern,
// but instead a list of filenames: see SetFileNames and class gdcm::FilenameGenerator
//
// .SECTION Warning
// You need to specify the correct ImageFormat (taken from the reader)
// You need to explicitly specify the DirectionCosines (taken from the reader)
// Since VTK 5.4 vtkMedicalImageProperties has its own DirectionCosine (no 's')
// user need to make sure the vtkMatrix4x4 is compatible with the 6-vector DirectionCosine.
//
// .SECTION NOTE Shift/Scale are global to all DICOM frames (=files) written
// as 2D slice, therefore the shift/scale operation might not be optimized for
// all slices. This is not recommended for image with a large dynamic range.
//
// .SECTION See Also
// vtkImageWriter vtkMedicalImageProperties vtkGDCMImageReader

#ifndef VTKGDCMIMAGEWRITER_H
#define VTKGDCMIMAGEWRITER_H

#include "vtkgdcmModule.h"
#include "vtkImageWriter.h"
#include "vtkVersion.h"

class vtkLookupTable;
class vtkMedicalImageProperties;
class vtkMatrix4x4;
class vtkStringArray;
class VTKGDCM_EXPORT vtkGDCMImageWriter : public vtkImageWriter
{
public:
  static vtkGDCMImageWriter *New();
  vtkTypeMacro(vtkGDCMImageWriter,vtkImageWriter);
  virtual void PrintSelf(ostream& os, vtkIndent indent);

  // Description:
  // Pass in the vtkmedicalimageproperties object for medical information
  // to be mapped to DICOM attributes.
  vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
  virtual void SetMedicalImageProperties(vtkMedicalImageProperties*);

  // Description:
  // Pass in the list of filename to be used to write out the DICOM file(s)
  virtual void SetFileNames(vtkStringArray*);
  vtkGetObjectMacro(FileNames, vtkStringArray);

```

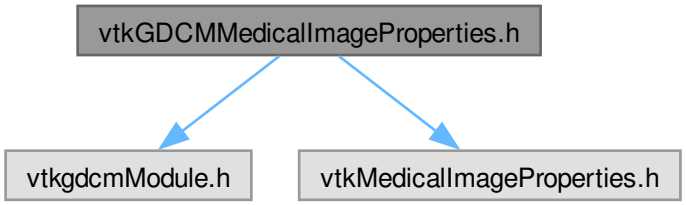
```
  // Description:
  // Set/Get whether or not the image was compressed using a lossy compression algorithm
  vtkGetMacro(LossyFlag,int);
  vtkSetMacro(LossyFlag,int);
  vtkBooleanMacro(LossyFlag,int);

  // I need that...
  virtual void Write();

  // Description:
  // Get the extension for this file format.
  virtual const char* GetFileExtensions() {
    return ".dcm .DCM"; }

  // Description:
  // Get the name of this file format.
  virtual const char* GetDescriptiveName() {
    return "DICOM"; }

  // Description:
  // You need to manually specify the direction the image is in to write a valid DICOM file
  // since vtkImageData do not contains one (eg. MR Image Storage, CT Image Storage...)
  virtual void SetDirectionCosines(vtkMatrix4x4 *matrix);
  vtkGetObjectMacro(DirectionCosines, vtkMatrix4x4);
  virtual void SetDirectionCosinesFromImageOrientationPatient(const double dircos[6]);

  // Description:
  // Modality LUT
  vtkSetMacro(Shift, double);
  vtkGetMacro(Shift, double);
  vtkSetMacro(Scale, double);
  vtkGetMacro(Scale, double);

  // Description:
  // See vtkGDCMImageReader for list of ImageFormat
  vtkGetMacro(ImageFormat,int);
  vtkSetMacro(ImageFormat,int);

  // Description:
  // Set/Get whether the data comes from the file starting in the lower left
  // corner or upper left corner.
  vtkBooleanMacro(FileLowerLeft, int);
  vtkGetMacro(FileLowerLeft, int);
  vtkSetMacro(FileLowerLeft, int);

  // Description:
  // For color image (more than a single comp) you can specify the planar configuration you prefer
  vtkSetMacro(PlanarConfiguration,int);
  vtkGetMacro(PlanarConfiguration,int);

  // Description:
  // Set/Get specific StudyUID / SeriesUID
  vtkSetStringMacro(StudyUID);
  vtkGetStringMacro(StudyUID);
  vtkSetStringMacro(SeriesUID);
  vtkGetStringMacro(SeriesUID);

//BTX
  enum CompressionTypes {
    NO_COMPRESSION = 0,   // raw (default)
    JPEG_COMPRESSION,     // JPEG
    JPEG2000_COMPRESSION, // J2K
    JPEGLS_COMPRESSION,   // JPEG-LS
    RLE_COMPRESSION       // RLE
  };
//ETX
  // Set/Get the compression type
  vtkSetMacro(CompressionType, int);
  vtkGetMacro(CompressionType, int);

  //void SetCompressionTypeFromString(const char *);
  //const char *GetCompressionTypeAsString();

protected:
  vtkGDCMImageWriter();
  ~vtkGDCMImageWriter();

#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
  int FillInputPortInformation(int port, vtkInformation *info);
  int RequestInformation(
    vtkInformation *request,
```

```
    vtkInformationVector **inputVector,
    vtkInformationVector *outputVector);
  int RequestUpdateExtent(
    vtkInformation *request,
    vtkInformationVector **inputVector,
    vtkInformationVector *outputVector);
  int RequestData(
    vtkInformation *request,
    vtkInformationVector **inputVector,
    vtkInformationVector *outputVector);
#else
  void WriteSlice(vtkImageData *data);
#endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
  int WriteGDCMData(vtkImageData *data, int timeStep);

protected:
  virtual /*const*/ char *GetFileName();

private:
  vtkGDCMImageWriter(const vtkGDCMImageWriter&);  // Not implemented.
  void operator=(const vtkGDCMImageWriter&);  // Not implemented.

  // VTK structs:
  //vtkLookupTable *LookupTable;
  vtkMedicalImageProperties *MedicalImageProperties;
  char *StudyUID;
  char *SeriesUID;

  int DataUpdateExtent[6];
  int ImageFormat;

  vtkStringArray *FileNames;
  vtkMatrix4x4 *DirectionCosines;

  double Shift;
  double Scale;
  int FileLowerLeft;
  int PlanarConfiguration;
  int LossyFlag;
  int CompressionType;
};

#endif
```

13.611 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkMedicalImageProperties.h"
```

Include dependency graph for vtkGDCMMedicalImageProperties.h:



Classes

- class vtkGDCMMedicalImageProperties

Namespaces

- namespace gdcm

13.612 vtkGDCMMedicalImageProperties.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkGDCMMedicalImageProperties - some medical image properties.
// .SECTION Description
// vtkGDCMMedicalImageProperties is a helper class that can be used by medical
// image readers and applications to encapsulate medical image/acquisition
// properties. Later on, this should probably be extended to add
// any user-defined property.
// .SECTION See Also
// vtkMedicalImageReader2

#ifndef VTKGDCMMEDICALIMAGEPROPERTIES_H
#define VTKGDCMMEDICALIMAGEPROPERTIES_H

#include "vtkgdcmModule.h"
#include "vtkMedicalImageProperties.h"

class vtkGDCMMedicalImagePropertiesInternals;
//BTX
namespace gdcm { class File; }
//ETX

class VTKGDCM_EXPORT vtkGDCMMedicalImageProperties : public vtkMedicalImageProperties
{
public:
  static vtkGDCMMedicalImageProperties *New();
  vtkTypeMacro(vtkGDCMMedicalImageProperties,vtkMedicalImageProperties);
  void PrintSelf(ostream& os, vtkIndent indent);

  // Description:
  // Convenience method to reset all fields to an empty string/value
  virtual void Clear();

/*
  // Description:
  // Patient name
  // For ex: DICOM (0010,0010) = DOE,JOHN
  vtkSetStringMacro(PatientName);
  vtkGetStringMacro(PatientName);

  // Description:
  // Patient ID
  // For ex: DICOM (0010,0020) = 1933197
  vtkSetStringMacro(PatientID);
  vtkGetStringMacro(PatientID);

  // Description:
  // Patient age
```

```
// Format: nnnD, nnW, nnnM or nnnY (eventually nnD, nnW, nnY)
//         with D (day), M (month), W (week), Y (year)
// For ex: DICOM (0010,1010) = 031Y
vtkSetStringMacro(PatientAge);
vtkGetStringMacro(PatientAge);

// Description:
// Take as input a string in VR=AS (DICOM PS3.5) and extract either
// different fields namely: year month week day
// Return 0 on error, 1 on success
// One can test fields if they are different from -1 upon success
static int GetAgeAsFields(const char *age, int &year, int &month, int &week, int &day);

// For Tcl:
// From C++ use GetPatientAge + GetAgeAsField
// Those function parse a DICOM string, and return the value of the number expressed
// this is either expressed in year, month or days. Thus if a string is expressed in years
// GetPatientAgeDay/GetPatientAgeWeek/GetPatientAgeMonth will return 0
int GetPatientAgeYear();
int GetPatientAgeMonth();
int GetPatientAgeWeek();
int GetPatientAgeDay();

// Description:
// Patient sex
// For ex: DICOM (0010,0040) = M
vtkSetStringMacro(PatientSex);
vtkGetStringMacro(PatientSex);

// Description:
// Patient birth date
// Format: yyyymmdd
// For ex: DICOM (0010,0030) = 19680427
vtkSetStringMacro(PatientBirthDate);
vtkGetStringMacro(PatientBirthDate);

// For Tcl:
// From C++ use GetPatientBirthDate + GetDateAsFields
int GetPatientBirthDateYear();
int GetPatientBirthDateMonth();
int GetPatientBirthDateDay();

// Description:
// Study Date
// Format: yyyymmdd
// For ex: DICOM (0008,0020) = 20030617
vtkSetStringMacro(StudyDate);
vtkGetStringMacro(StudyDate);

// Description:
// Acquisition Date
// Format: yyyymmdd
// For ex: DICOM (0008,0022) = 20030617
vtkSetStringMacro(AcquisitionDate);
vtkGetStringMacro(AcquisitionDate);

// For Tcl:
// From C++ use GetAcquisitionDate + GetDateAsFields
int GetAcquisitionDateYear();
int GetAcquisitionDateMonth();
int GetAcquisitionDateDay();

// Description:
// Study Time
// Format: hhmmss.frac (any trailing component(s) can be omitted)
// For ex: DICOM (0008,0030) = 162552.0705 or 230012, or 0012
vtkSetStringMacro(StudyTime);
vtkGetStringMacro(StudyTime);

// Description:
// Acquisition time
// Format: hhmmss.frac (any trailing component(s) can be omitted)
// For ex: DICOM (0008,0032) = 162552.0705 or 230012, or 0012
vtkSetStringMacro(AcquisitionTime);
vtkGetStringMacro(AcquisitionTime);

// Description:
// Image Date aka Content Date
// Format: yyyymmdd
// For ex: DICOM (0008,0023) = 20030617
vtkSetStringMacro(ImageDate);
```

```
vtkGetStringMacro(ImageDate);

// For Tcl:
// From C++ use GetImageDate + GetDateAsFields
int GetImageDateYear();
int GetImageDateMonth();
int GetImageDateDay();

// Description:
// Take as input a string in ISO 8601 date (YYYY/MM/DD) and extract the
// different fields namely: year month day
// Return 0 on error, 1 on success
static int GetDateAsFields(const char *date, int &year, int &month, int &day);

// Description:
// Take as input a string in ISO 8601 date (YYYY/MM/DD) and construct a
// locale date based on the different fields (see GetDateAsFields to extract
// different fields)
// Return 0 on error, 1 on success
static int GetDateAsLocale(const char *date, char *locale);

// Description:
// Image Time
// Format: hhmmss.frac (any trailing component(s) can be omitted)
// For ex: DICOM (0008,0033) = 162552.0705 or 230012, or 0012
vtkSetStringMacro(ImageTime);
vtkGetStringMacro(ImageTime);

// Description:
// Image number
// For ex: DICOM (0020,0013) = 1
vtkSetStringMacro(ImageNumber);
vtkGetStringMacro(ImageNumber);

// Description:
// Series number
// For ex: DICOM (0020,0011) = 902
vtkSetStringMacro(SeriesNumber);
vtkGetStringMacro(SeriesNumber);

// Description:
// Series Description
// User provided description of the Series
// For ex: DICOM (0008,103e) = SCOUT
vtkSetStringMacro(SeriesDescription);
vtkGetStringMacro(SeriesDescription);

// Description:
// Study ID
// For ex: DICOM (0020,0010) = 37481
vtkSetStringMacro(StudyID);
vtkGetStringMacro(StudyID);

// Description:
// Study description
// For ex: DICOM (0008,1030) = BRAIN/C-SP/FACIAL
vtkSetStringMacro(StudyDescription);
vtkGetStringMacro(StudyDescription);

// Description:
// Modality
// For ex: DICOM (0008,0060)= CT
vtkSetStringMacro(Modality);
vtkGetStringMacro(Modality);

// Description:
// Manufacturer
// For ex: DICOM (0008,0070) = Siemens
vtkSetStringMacro(Manufacturer);
vtkGetStringMacro(Manufacturer);

// Description:
// Manufacturer's Model Name
// For ex: DICOM (0008,1090) = LightSpeed QX/i
vtkSetStringMacro(ManufacturerModelName);
vtkGetStringMacro(ManufacturerModelName);

// Description:
// Station Name
// For ex: DICOM (0008,1010) = LSPD_OC8
vtkSetStringMacro(StationName);
```

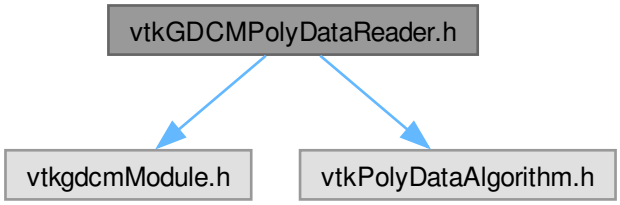
```
vtkGetStringMacro(StationName);

// Description:
// Institution Name
// For ex: DICOM (0008,0080) = FooCity Medical Center
vtkSetStringMacro(InstitutionName);
vtkGetStringMacro(InstitutionName);

// Description:
// Convolution Kernel (or algorithm used to reconstruct the data)
// For ex: DICOM (0018,1210) = Bone
vtkSetStringMacro(ConvolutionKernel);
vtkGetStringMacro(ConvolutionKernel);

// Description:
// Slice Thickness (Nominal reconstructed slice thickness, in mm)
// For ex: DICOM (0018,0050) = 0.273438
vtkSetStringMacro(SliceThickness);
vtkGetStringMacro(SliceThickness);
virtual double GetSliceThicknessAsDouble();

// Description:
// Peak kilo voltage output of the (x-ray) generator used
// For ex: DICOM (0018,0060) = 120
vtkSetStringMacro(KVP);
vtkGetStringMacro(KVP);

// Description:
// Gantry/Detector tilt (Nominal angle of tilt in degrees of the scanning
// gantry.)
// For ex: DICOM (0018,1120) = 15
vtkSetStringMacro(GantryTilt);
vtkGetStringMacro(GantryTilt);
virtual double GetGantryTiltAsDouble();

// Description:
// Echo Time
// (Time in ms between the middle of the excitation pulse and the peak of
// the echo produced)
// For ex: DICOM (0018,0081) = 105
vtkSetStringMacro(EchoTime);
vtkGetStringMacro(EchoTime);

// Description:
// Echo Train Length
// (Number of lines in k-space acquired per excitation per image)
// For ex: DICOM (0018,0091) = 35
vtkSetStringMacro(EchoTrainLength);
vtkGetStringMacro(EchoTrainLength);

// Description:
// Repetition Time
// The period of time in msec between the beginning of a pulse sequence and
// the beginning of the succeeding (essentially identical) pulse sequence.
// For ex: DICOM (0018,0080) = 2040
vtkSetStringMacro(RepetitionTime);
vtkGetStringMacro(RepetitionTime);

// Description:
// Exposure time (time of x-ray exposure in msec)
// For ex: DICOM (0018,1150) = 5
vtkSetStringMacro(ExposureTime);
vtkGetStringMacro(ExposureTime);

// Description:
// X-ray tube current (in mA)
// For ex: DICOM (0018,1151) = 400
vtkSetStringMacro(XRayTubeCurrent);
vtkGetStringMacro(XRayTubeCurrent);

// Description:
// Exposure (The exposure expressed in mAs, for example calculated
// from Exposure Time and X-ray Tube Current)
// For ex: DICOM (0018,1152) = 114
vtkSetStringMacro(Exposure);
vtkGetStringMacro(Exposure);

// Interface to allow insertion of user define values, for instance in DICOM one would want to
// store the Protocol Name (0018,1030), in this case one would do:
// AddUserDefinedValue( "Protocol Name", "T1W/SE/1024" );
void AddUserDefinedValue(const char *name, const char *value);
```

```
  // Get a particular user value
  const char *GetUserDefinedValue(const char *name);
  // Get the number of user defined values
  unsigned int GetNumberOfUserDefinedValues();
  // Get a name/value by index
  const char *GetUserDefinedNameByIndex(unsigned int idx);
  const char *GetUserDefinedValueByIndex(unsigned int idx);

  // Description:
  // Copy the contents of p to this instance.
  virtual void DeepCopy(vtkGDCMMedicalImageProperties *p);

  // Description:
  // Add/Remove/Query the window/level presets that may have been associated
  // to a medical image. Window is also known as 'width', level is also known
  // as 'center'. The same window/level pair can not be added twice.
  // As a convenience, a comment (aka Explanation) can be associated to a preset.
  // For ex: DICOM Window Center (0028,1050) = 00045\000470
  //         DICOM Window Width  (0028,1051) = 0106\03412
  //         DICOM Window Center Width Explanation (0028,1055) = WINDOW1\WINDOW2
  virtual void AddWindowLevelPreset(double w, double l);
  virtual void RemoveWindowLevelPreset(double w, double l);
  virtual void RemoveAllWindowLevelPresets();
  virtual int GetNumberOfWindowLevelPresets();
  virtual int HasWindowLevelPreset(double w, double l);
  virtual int GetNthWindowLevelPreset(int idx, double *w, double *l);
  virtual double* GetNthWindowLevelPreset(int idx);
  virtual void SetNthWindowLevelPresetComment(int idx, const char *comment);
  virtual const char* GetNthWindowLevelPresetComment(int idx);

  // Description:
  // Mapping from a sliceidx within a volumeidx into a DICOM Instance UID
  // Some DICOM reader can populate this structure so that later on from a slice index
  // in a vtkImageData volume we can backtrack and find out which 2d slice it was coming from
  const char *GetInstanceUIDFromSliceID(int volumeidx, int sliceid);
  void SetInstanceUIDFromSliceID(int volumeidx, int sliceid, const char *uid);

  // Description:
  // Provides the inverse mapping. Returns -1 if a slice for this uid is
  // not found.
  int GetSliceIDFromInstanceUID(int &volumeidx, const char *uid);

//BTX
  typedef enum {
    AXIAL = 0,
    CORONAL,
    SAGITTAL
  } OrientationType;
//ETX
  int GetOrientationType(int volumeidx);
  void SetOrientationType(int volumeidx, int orientation);
  static const char *GetStringFromOrientationType(unsigned int type);
*/
protected:
  vtkGDCMMedicalImageProperties();
  ~vtkGDCMMedicalImageProperties();

//BTX
  friend class vtkGDCMImageReader;
  friend class vtkGDCMImageReader2;
  friend class vtkGDCMImageWriter;
  void PushBackFile(gdcm::File const &f);
  gdcm::File const & GetFile(unsigned int t);
//ETX

private:
  vtkGDCMMedicalImagePropertiesInternals *Internals;

  vtkGDCMMedicalImageProperties(const vtkGDCMMedicalImageProperties&); // Not implemented.
  void operator=(const vtkGDCMMedicalImageProperties&); // Not implemented.
};

#endif
```

13.613 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:



Classes

- class vtkGDCMPolyDataReader

Namespaces

- namespace gdcm

13.614 vtkGDCMPolyDataReader.h

Go to the documentation of this file.

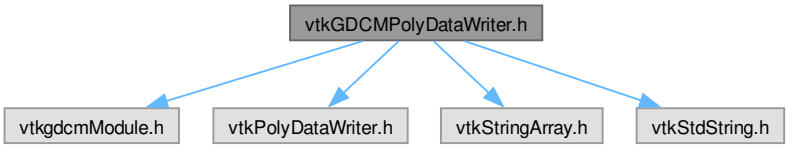
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 // .NAME vtkGDCMPolyDataReader - read DICOM PolyData files (Contour Data...)
00015 // .SECTION Description
00016 // For now only support RTSTRUCT (RT Structure Set Storage)
00017 // .SECTION TODO
00018 // Need to do the same job for DVH Sequence/DVH Data...
00019 // .SECTION Warning
00020 // When using vtkGDCMPolyDataReader in conjunction with vtkGDCMImageReader
00021 // it is *required* that FileLowerLeft is set to ON as coordinate system
00022 // would be inconsistent in between the two data structures.
00023 //
00024 // .SECTION See Also
```

```
// vtkGDCMImageReader vtkGDCMPolyDataWriter vtkRTStructSetProperties


#ifndef VTKGDCMPOLYDATAREADER_H
#define VTKGDCMPOLYDATAREADER_H

#include "vtkgdcmModule.h"
#include "vtkPolyDataAlgorithm.h"

class vtkMedicalImageProperties;
class vtkRTStructSetProperties;
//BTX
namespace gdcm { class Reader; }
//ETX
class VTKGDCM_EXPORT vtkGDCMPolyDataReader : public vtkPolyDataAlgorithm
{
public:
  static vtkGDCMPolyDataReader *New();
  vtkTypeMacro(vtkGDCMPolyDataReader,vtkPolyDataAlgorithm);
  virtual void PrintSelf(ostream& os, vtkIndent indent);

  // Description:
  // Set/Get the filename of the file to be read
  vtkSetStringMacro(FileName);
  vtkGetStringMacro(FileName);

  // Description:
  // Get the medical image properties object
  vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);

  vtkGetObjectMacro(RTStructSetProperties, vtkRTStructSetProperties);

protected:
  vtkGDCMPolyDataReader();
  ~vtkGDCMPolyDataReader();

  char *FileName;
  vtkMedicalImageProperties *MedicalImageProperties;
  vtkRTStructSetProperties *RTStructSetProperties;
//BTX
  void FillMedicalImageInformation(const gdcm::Reader &reader);
//ETX

  int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *);
  int RequestInformation(
    vtkInformation *vtkNotUsed(request),
    vtkInformationVector **vtkNotUsed(inputVector),
    vtkInformationVector *outputVector);
//BTX
  int RequestInformation_RTStructureSetStorage(gdcm::Reader const & reader);
  int RequestData_RTStructureSetStorage(gdcm::Reader const &reader, vtkInformationVector *outputVector);
  int RequestInformation_HemodynamicWaveformStorage(gdcm::Reader const & reader);
  int RequestData_HemodynamicWaveformStorage(gdcm::Reader const &reader, vtkInformationVector
*outputVector);
//ETX

private:
  vtkGDCMPolyDataReader(const vtkGDCMPolyDataReader&);  // Not implemented.
  void operator=(const vtkGDCMPolyDataReader&);  // Not implemented.
};

#endif
```

13.615 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkPolyDataWriter.h"
#include "vtkStringArray.h"
#include "vtkStdString.h"
```

Include dependency graph for vtkGDCMPolyDataWriter.h:



Classes

- class vtkGDCMPolyDataWriter

Namespaces

- namespace gdcm

13.616 vtkGDCMPolyDataWriter.h

Go to the documentation of this file.

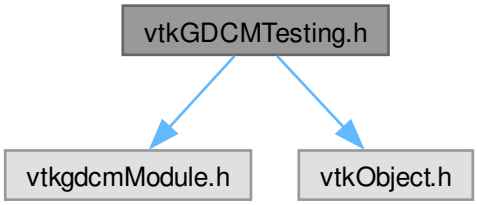
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkGDCMPolyDataWriter - writer DICOM PolyData files (Contour Data...)
// .SECTION Description
// For now only support RTSTRUCT (RT Structure Set Storage)
// .SECTION TODO
// Need to do the same job for DVH Sequence/DVH Data...
// .SECTION Warning
//
// .SECTION See Also
// vtkGDCMImageReader vtkGDCMPolyDataReader vtkRTStructSetProperties


#ifndef VTKGDCMPOLYDATAWRITER_H
#define VTKGDCMPOLYDATAWRITER_H

#include "vtkgdcmModule.h"
#include "vtkPolyDataWriter.h"
#include "vtkStringArray.h"
#include "vtkStdString.h"


class vtkMedicalImageProperties;
class vtkRTStructSetProperties;
//BTX
namespace gdcm { class File; }
```

```
//ETX
class VTKGDCM_EXPORT vtkGDCMPolyDataWriter : public vtkPolyDataWriter
{
public:
  static vtkGDCMPolyDataWriter *New();
  vtkTypeMacro(vtkGDCMPolyDataWriter,vtkPolyDataWriter);
  virtual void PrintSelf(ostream& os, vtkIndent indent);

  // Description:
  // Set/Get the filename of the file to be read
//  vtkSetStringMacro(FileName);
//  vtkGetStringMacro(FileName);

  // Description:
  // Get the medical image properties object
//  vtkGetObjectMacro(MedicalImageProperties, vtkMedicalImageProperties);
  virtual void SetMedicalImageProperties(vtkMedicalImageProperties *pd);

  virtual void SetRTStructSetProperties(vtkRTStructSetProperties *pd);


  //this function will initialize the contained rtstructset with
  //the inputs of the writer and the various extra information
  //necessary for writing a complete rtstructset.
  //NOTE: inputs must be set BEFORE calling this function!
  //NOTE: the number of outputs for the appendpolydata MUST MATCH the ROI vectors!
  void InitializeRTStructSet(vtkStdString inDirectory,
     vtkStdString inStructLabel, vtkStdString inStructName,
     vtkStringArray* inROINames,
     vtkStringArray* inROIAlgorithmName,
     vtkStringArray* inROIType);

  // make parent class public...
  void SetNumberOfInputPorts(int n);

protected:
  vtkGDCMPolyDataWriter();
  ~vtkGDCMPolyDataWriter();

  vtkMedicalImageProperties *MedicalImageProperties;
  vtkRTStructSetProperties *RTStructSetProperties;

  void WriteData();
//BTX
  void WriteRTSTRUCTInfo(gdcm::File &file);
  void WriteRTSTRUCTData(gdcm::File &file, int num);
//ETX

private:
  vtkGDCMPolyDataWriter(const vtkGDCMPolyDataWriter&);  // Not implemented.
  void operator=(const vtkGDCMPolyDataWriter&);  // Not implemented.
};

#endif
```

13.617 vtkGDCMTesting.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkObject.h"
```

Include dependency graph for vtkGDCMTesting.h:



Classes

- class vtkGDCMTesting

13.618 vtkGDCMTesting.h

Go to the documentation of this file.

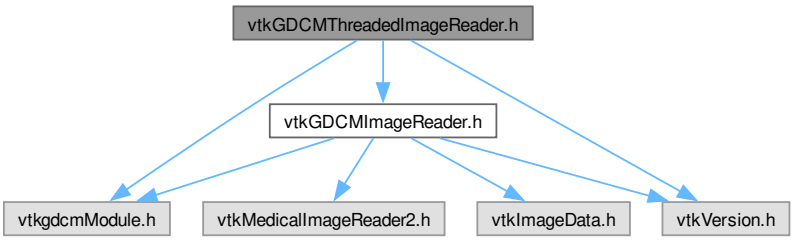
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkGDCMTesting - GDCM Testing
// .SECTION Description
// GDCM Testing

// .SECTION See Also
// vtkTesting

#ifndef VTKGDCMTESTING_H
#define VTKGDCMTESTING_H

#include "vtkgdcmModule.h"
#include "vtkObject.h"

class VTKGDCM_EXPORT vtkGDCMTesting : public vtkObject
{
public:
  static vtkGDCMTesting *New();
  vtkTypeMacro(vtkGDCMTesting,vtkObject);
  void PrintSelf(ostream& os, vtkIndent indent);

  static const char *GetVTKDataRoot();
  static const char *GetGDCMDataRoot();

//BTX
  typedef const char* const (*MD5MetaImagesType)[3];
  static const char * const * GetMD5MetaImage(unsigned int file);
```

```
//ETX
  static unsigned int GetNumberOfMD5MetaImages();

  static const char * GetMHDMD5FromFile(const char *filepath);
  static const char * GetRAWMD5FromFile(const char *filepath);

protected:
  vtkGDCMTesting();
  ~vtkGDCMTesting();

private:
  vtkGDCMTesting(const vtkGDCMTesting&);  // Not implemented.
  void operator=(const vtkGDCMTesting&);  // Not implemented.
};

#endif
```

13.619 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkGDCMImageReader.h"
#include "vtkVersion.h"
```

Include dependency graph for vtkGDCMThreadedImageReader.h:



Classes

- class vtkGDCMThreadedImageReader

13.620 vtkGDCMThreadedImageReader.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.
```

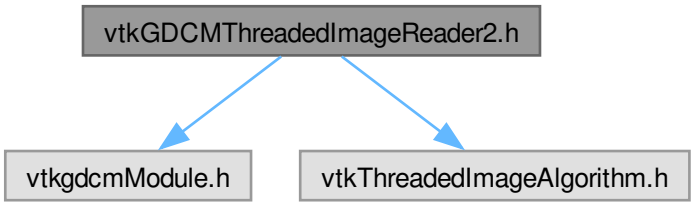
```
=========================================================================*/
// .NAME vtkGDCMThreadedImageReader - read DICOM files with multiple threads
// .SECTION Description
// vtkGDCMThreadedImageReader is a source object that reads some DICOM files
// This reader is threaded. Meaning that on a multiple core CPU with N cpu, it will
// read approx N times faster than when reading in a single thread.
//
// .SECTION Warning: Advanced users only. Do not use this class in the general case,
// you have to understand how physically medium works first (sequential reading for
// instance) before playing with this class
//
// .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
// upside down as VTK would expect, use this option only if you know what you are doing
//
// .SECTION FIXME: need to implement the other mode where FileLowerLeft is set to OFF
//
// .SECTION FIXME: you need to call SetFileName when reading a volume file (multiple slices DICOM)
// since SetFileNames expect each single file to be single slice (see parent class)
//
// .SECTION BUG: you should really consider using vtkGDCMThreadedImageReader2 instead !
//
// .SECTION See Also
// vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMThreadedImageReader2

#ifndef VTKGDCMTHREADEDIMAGEREADER_H
#define VTKGDCMTHREADEDIMAGEREADER_H

#include "vtkgdcmModule.h"
#include "vtkGDCMImageReader.h"
#include "vtkVersion.h"

class VTKGDCM_EXPORT vtkGDCMThreadedImageReader : public vtkGDCMImageReader
{
public:
  static vtkGDCMThreadedImageReader *New();
  vtkTypeMacro(vtkGDCMThreadedImageReader,vtkGDCMImageReader);
  virtual void PrintSelf(ostream& os, vtkIndent indent);

  // Description:
  // Explicitly set the Rescale Intercept (0028,1052)
  vtkSetMacro(Shift,double);

  // Description:
  // Explicitly get/set the Rescale Slope (0028,1053)
  vtkSetMacro(Scale,double);

  // Description:
  // Determine whether or not reader should use value from Shift/Scale
  // Default is 1
  vtkSetMacro(UseShiftScale,int);
  vtkGetMacro(UseShiftScale,int);
  vtkBooleanMacro(UseShiftScale,int);

  // Within this class this is allowed to set the Number of Overlays from outside
  //vtkSetMacro(NumberOfOverlays,int);

protected:
  vtkGDCMThreadedImageReader();
  ~vtkGDCMThreadedImageReader();

#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
  int RequestInformation(vtkInformation *request,
                         vtkInformationVector **inputVector,
                         vtkInformationVector *outputVector);
  int RequestData(vtkInformation *request,
                  vtkInformationVector **inputVector,
                  vtkInformationVector *outputVector);
#else /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/
  void ExecuteInformation();
  void ExecuteData(vtkDataObject *out);
#endif /*(VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )*/

  void ReadFiles(unsigned int nfiles, const char *filenames[]);
  void RequestDataCompat();

private:
  vtkGDCMThreadedImageReader(const vtkGDCMThreadedImageReader&);  // Not implemented.
  void operator=(const vtkGDCMThreadedImageReader&);  // Not implemented.

  int UseShiftScale;
```

```
};

#endif
```

13.621 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkGDCMThreadedImageReader2.h:



Classes

- class vtkGDCMThreadedImageReader2

13.622 vtkGDCMThreadedImageReader2.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkGDCMThreadedImageReader2 - read DICOM files with multiple threads
// .SECTION Description
// vtkGDCMThreadedImageReader2 is a source object that reads some DICOM files
// This reader is threaded. Meaning that on a multiple core CPU with N cpu, it will
// read approx N times faster than when reading in a single thread assuming the IO is
// not a bottleneck operation.
// If looking for a single threaded class see: vtkGDCMImageReader
//
// .SECTION Warning: Advanced users only. Do not use this class in the general case,
// you have to understand how physically medium works first (sequential reading for
// instance) before playing with this class
```

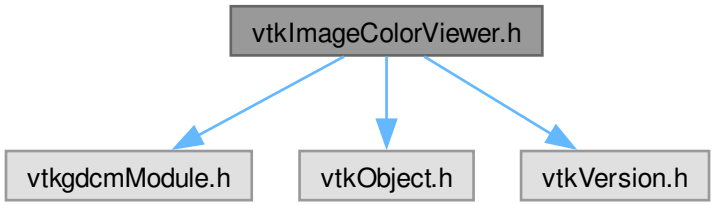
```
//
// .SECTION Implementation note: when FileLowerLeft is set to on the image is not flipped
// upside down as VTK would expect, use this option only if you know what you are doing
//
// .SECTION FIXME: need to implement the other mode where FileLowerLeft is set to OFF
//
// .SECTION FIXME: need to implement reading of series of 3D files
//
// .SECTION Implementation note: this class is meant to supersede vtkGDCMThreadedImageReader
// because it had support for ProgressEvent support even from python layer. There is a
// subtle trick down in the threading mechanism in VTK were the main thread (talking to the
// python interpreter) is also part of the execution process (and the N-1 other thread
// are just there to execute the remaining of ThreadedRequestData), this separation into
// two types of thread is necessary to achieve a working implementation of UpdateProgress

// .SECTION See Also
// vtkMedicalImageReader2 vtkMedicalImageProperties vtkGDCMImageReader

#ifndef VTKGDCMTHREADEDIMAGEREADER2_H
#define VTKGDCMTHREADEDIMAGEREADER2_H

#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"

class vtkStringArray;
class VTKGDCM_EXPORT vtkGDCMThreadedImageReader2 : public vtkThreadedImageAlgorithm
{
public:
  static vtkGDCMThreadedImageReader2 *New();
  vtkTypeMacro(vtkGDCMThreadedImageReader2,vtkThreadedImageAlgorithm);
  virtual void PrintSelf(ostream& os, vtkIndent indent);

  vtkGetMacro(FileLowerLeft,int);
  vtkSetMacro(FileLowerLeft,int);
  vtkBooleanMacro(FileLowerLeft,int);

  vtkGetMacro(NumberOfOverlays,int);

  vtkSetMacro(DataScalarType,int);
  vtkGetMacro(DataScalarType,int);

  vtkSetMacro(NumberOfScalarComponents,int);
  vtkGetMacro(NumberOfScalarComponents,int);

  vtkGetMacro(LoadOverlays,int);
  vtkSetMacro(LoadOverlays,int);
  vtkBooleanMacro(LoadOverlays,int);

  vtkSetVector6Macro(DataExtent,int);
  vtkGetVector6Macro(DataExtent,int);

  vtkSetVector3Macro(DataOrigin,double);
  vtkGetVector3Macro(DataOrigin,double);

  vtkSetVector3Macro(DataSpacing,double);
  vtkGetVector3Macro(DataSpacing,double);

  //vtkGetStringMacro(FileName);
  //vtkSetStringMacro(FileName);
  virtual const char *GetFileName(int i = 0);
  virtual void SetFileName(const char *filename);

  virtual void SetFileNames(vtkStringArray*);
  vtkGetObjectMacro(FileNames, vtkStringArray);

  int SplitExtent(int splitExt[6], int startExt[6],
                  int num, int total);

  // Description:
  // Explicitly set the Rescale Intercept (0028,1052)
  vtkSetMacro(Shift,double);
  vtkGetMacro(Shift,double);

  // Description:
  // Explicitly get/set the Rescale Slope (0028,1053)
  vtkSetMacro(Scale,double);
  vtkGetMacro(Scale,double);

  // Description:
  // Determine whether or not reader should use value from Shift/Scale
  // Default is 1
```

```
  vtkSetMacro(UseShiftScale,int);
  vtkGetMacro(UseShiftScale,int);
  vtkBooleanMacro(UseShiftScale,int);

protected:
  vtkGDCMThreadedImageReader2();
  ~vtkGDCMThreadedImageReader2();

  int RequestInformation(vtkInformation *request,
                         vtkInformationVector **inputVector,
                         vtkInformationVector *outputVector);

protected:
  void ThreadedRequestData (
    vtkInformation * request,
    vtkInformationVector** inputVector,
    vtkInformationVector * outputVector,
    vtkImageData ***inData,
    vtkImageData **outData,
    int outExt[6], int id);

private:
  int FileLowerLeft;
  char *FileName;
  vtkStringArray *FileNames;
  int LoadIconImage;
  int DataExtent[6];
  int LoadOverlays;
  int NumberOfOverlays;
  int DataScalarType;

  int NumberOfScalarComponents;
  double DataSpacing[3];
  double DataOrigin[3];
  int IconImageDataExtent[6];

  double Shift;
  double Scale;
  int UseShiftScale;

private:
  vtkGDCMThreadedImageReader2(const vtkGDCMThreadedImageReader2&);  // Not implemented.
  void operator=(const vtkGDCMThreadedImageReader2&);  // Not implemented.
};

#endif
```

13.623 vtkImageColorViewer.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkObject.h"
#include "vtkVersion.h"
```

Include dependency graph for vtkImageColorViewer.h:



Classes

- class vtkImageColorViewer

13.624 vtkImageColorViewer.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkImageColorViewer - Display a 2D image.
// .SECTION Description
// vtkImageColorViewer is a convenience class for displaying a 2D image.  It
// packages up the functionality found in vtkRenderWindow, vtkRenderer,
// vtkImageActor and vtkImageMapToWindowLevelColors into a single easy to use
// class.  This class also creates an image interactor style
// (vtkInteractorStyleImage) that allows zooming and panning of images, and
// supports interactive window/level operations on the image. Note that
// vtkImageColorViewer is simply a wrapper around these classes.
//
// vtkImageColorViewer uses the 3D rendering and texture mapping engine
// to draw an image on a plane.  This allows for rapid rendering,
// zooming, and panning. The image is placed in the 3D scene at a
// depth based on the z-coordinate of the particular image slice. Each
// call to SetSlice() changes the image data (slice) displayed AND
// changes the depth of the displayed slice in the 3D scene. This can
// be controlled by the AutoAdjustCameraClippingRange ivar of the
// InteractorStyle member.
//
// It is possible to mix images and geometry, using the methods:
//
// viewer->SetInput( myImage );
// viewer->GetRenderer()->AddActor( myActor );
//
// This can be used to annotate an image with a PolyData of "edges" or
// or highlight sections of an image or display a 3D isosurface
// with a slice from the volume, etc. Any portions of your geometry
// that are in front of the displayed slice will be visible; any
// portions of your geometry that are behind the displayed slice will
// be obscured. A more general framework (with respect to viewing
// direction) for achieving this effect is provided by the
// vtkImagePlaneWidget .
//
// Note that pressing 'r' will reset the window/level and pressing
// shift+'r' or control+'r' will reset the camera.
//
// .SECTION See Also
// vtkRenderWindow vtkRenderer vtkImageActor vtkImageMapToWindowLevelColors

#ifndef VTKIMAGECOLORVIEWER_H
#define VTKIMAGECOLORVIEWER_H

#include "vtkgdcmModule.h"
#include "vtkObject.h"
#include "vtkVersion.h"

class vtkAlgorithm;
class vtkAlgorithmOutput;
class vtkImageActor;
class vtkImageData;
class vtkImageMapToWindowLevelColors2;
class vtkInformation;
class vtkInteractorStyleImage;
```

```
00067 class vtkRenderWindow;
00068 class vtkRenderer;
00069 class vtkRenderWindowInteractor;
00070 class vtkPolyData;
00071
00072 class VTKGDCM_EXPORT vtkImageColorViewer : public vtkObject
00073 {
00074 public:
00075   static vtkImageColorViewer *New();
00076   vtkTypeMacro(vtkImageColorViewer,vtkObject);
00077   void PrintSelf(ostream& os, vtkIndent indent);
00078
00079   // Description:
00080   // Get the name of rendering window.
00081   virtual const char *GetWindowName();
00082
00083   // Description:
00084   // Render the resulting image.
00085   virtual void Render(void);
00086
00087   // Description:
00088   // Set/Get the input image to the viewer.
00089 #if (VTK_MAJOR_VERSION >= 6)
00090   virtual void SetInputData(vtkImageData *in);
00091 #else
00092   virtual void SetInput(vtkImageData *in);
00093 #endif
00094   virtual vtkImageData *GetInput();
00095   virtual void SetInputConnection(vtkAlgorithmOutput* input);
00096   virtual void AddInputConnection(vtkAlgorithmOutput* input);
00097   virtual void AddInput(vtkImageData * input);
00098   //virtual void AddInput(vtkPolyData * input);
00099
00100   double GetOverlayVisibility();
00101   void SetOverlayVisibility(double vis);
00102
00103   // Description:
00104   // Set/get the slice orientation
00105   //BTX
00106   enum
00107   {
00108     SLICE_ORIENTATION_YZ = 0,
00109     SLICE_ORIENTATION_XZ = 1,
00110     SLICE_ORIENTATION_XY = 2
00111   };
00112   //ETX
00113   vtkGetMacro(SliceOrientation, int);
00114   virtual void SetSliceOrientation(int orientation);
00115   virtual void SetSliceOrientationToXY()
00116     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_XY); };
00117   virtual void SetSliceOrientationToYZ()
00118     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_YZ); };
00119   virtual void SetSliceOrientationToXZ()
00120     { this->SetSliceOrientation(vtkImageColorViewer::SLICE_ORIENTATION_XZ); };
00121
00122   // Description:
00123   // Set/Get the current slice to display (depending on the orientation
00124   // this can be in X, Y or Z).
00125   vtkGetMacro(Slice, int);
00126   virtual void SetSlice(int s);
00127
00128   // Description:
00129   // Update the display extent manually so that the proper slice for the
00130   // given orientation is displayed. It will also try to set a
00131   // reasonable camera clipping range.
00132   // This method is called automatically when the Input is changed, but
00133   // most of the time the input of this class is likely to remain the same,
00134   // i.e. connected to the output of a filter, or an image reader. When the
00135   // input of this filter or reader itself is changed, an error message might
00136   // be displayed since the current display extent is probably outside
00137   // the new whole extent. Calling this method will ensure that the display
00138   // extent is reset properly.
00139   virtual void UpdateDisplayExtent();
00140
00141   // Description:
00142   // Return the minimum and maximum slice values (depending on the orientation
00143   // this can be in X, Y or Z).
00144   virtual int GetSliceMin();
00145   virtual int GetSliceMax();
00146   virtual void GetSliceRange(int range[2])
00147     { this->GetSliceRange(range[0], range[1]); }
```

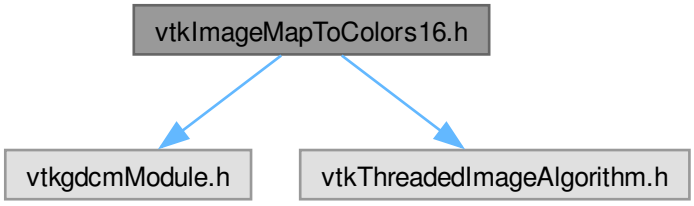
```
  virtual void GetSliceRange(int &min, int &max);
  virtual int* GetSliceRange();

  // Description:
  // Set window and level for mapping pixels to colors.
  virtual double GetColorWindow();
  virtual double GetColorLevel();
  virtual void SetColorWindow(double s);
  virtual void SetColorLevel(double s);

  // Description:
  // These are here when using a Tk window.
  virtual void SetDisplayId(void *a);
  virtual void SetWindowId(void *a);
  virtual void SetParentId(void *a);

  // Description:
  // Set/Get the position in screen coordinates of the rendering window.
  virtual int* GetPosition();
  virtual void SetPosition(int a,int b);
  virtual void SetPosition(int a[2]) { this->SetPosition(a[0],a[1]); }

  // Description:
  // Set/Get the size of the window in screen coordinates in pixels.
  virtual int* GetSize();
  virtual void SetSize(int a, int b);
  virtual void SetSize(int a[2]) { this->SetSize(a[0],a[1]); }

  // Description:
  // Get the internal render window, renderer, image actor, and
  // image map instances.
  vtkGetObjectMacro(RenderWindow,vtkRenderWindow);
  vtkGetObjectMacro(Renderer, vtkRenderer);
  vtkGetObjectMacro(ImageActor,vtkImageActor);
  vtkGetObjectMacro(WindowLevel,vtkImageMapToWindowLevelColors2);
  vtkGetObjectMacro(InteractorStyle,vtkInteractorStyleImage);

  // Description:
  // Set your own renderwindow and renderer
  virtual void SetRenderWindow(vtkRenderWindow *arg);
  virtual void SetRenderer(vtkRenderer *arg);

  // Description:
  // Attach an interactor for the internal render window.
  virtual void SetupInteractor(vtkRenderWindowInteractor*);

  // Description:
  // Create a window in memory instead of on the screen. This may not
  // be supported for every type of window and on some windows you may
  // need to invoke this prior to the first render.
  virtual void SetOffScreenRendering(int);
  virtual int GetOffScreenRendering();
  vtkBooleanMacro(OffScreenRendering,int);

protected:
  vtkImageColorViewer();
  ~vtkImageColorViewer();

  virtual void InstallPipeline();
  virtual void UnInstallPipeline();

  vtkImageMapToWindowLevelColors2  *WindowLevel;
  vtkRenderWindow                 *RenderWindow;
  vtkRenderer                     *Renderer;
  vtkImageActor                   *ImageActor;
  vtkImageActor                   *OverlayImageActor;
  vtkRenderWindowInteractor       *Interactor;
  vtkInteractorStyleImage         *InteractorStyle;

  int SliceOrientation;
  int FirstRender;
  int Slice;

  virtual void UpdateOrientation();

#if (VTK_MAJOR_VERSION >= 6)
  vtkAlgorithm* GetInputAlgorithm();
  vtkInformation* GetInputInformation();
#endif

  friend class vtkImageColorViewerCallback;
```

```
00229
00230 private:
00231   vtkImageColorViewer(const vtkImageColorViewer&);  // Not implemented.
00232   void operator=(const vtkImageColorViewer&);  // Not implemented.
00233 };
00234
00235 #endif
```

13.625 vtkImageMapToColors16.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageMapToColors16.h:



Classes

- class vtkImageMapToColors16

13.626 vtkImageMapToColors16.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 /*=========================================================================
00015
00016   Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018   Program:   Visualization Toolkit
00019   Module:    $RCSfile: vtkImageMapToColors16.h,v $
00020
```

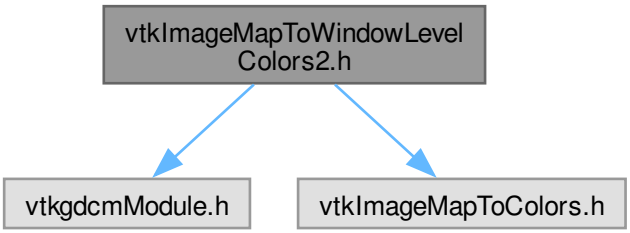
```
  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
  All rights reserved.
  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkImageMapToColors16 - map the input image through a lookup table
// .SECTION Description
// The vtkImageMapToColors16 filter will take an input image of any valid
// scalar type, and map the first component of the image through a
// lookup table.  The result is an image of type VTK_UNSIGNED_CHAR.
// If the lookup table is not set, or is set to NULL, then the input
// data will be passed through if it is already of type VTK_UNSIGNED_CHAR.

// .SECTION See Also
// vtkLookupTable vtkScalarsToColors

#ifndef VTKIMAGEMAPTOCOLORS16_H
#define VTKIMAGEMAPTOCOLORS16_H


#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"

class vtkScalarsToColors;

class VTKGDCM_EXPORT vtkImageMapToColors16 : public vtkThreadedImageAlgorithm
{
public:
  static vtkImageMapToColors16 *New();
  vtkTypeMacro(vtkImageMapToColors16,vtkThreadedImageAlgorithm);
  void PrintSelf(ostream& os, vtkIndent indent);

  // Description:
  // Set the lookup table.
  virtual void SetLookupTable(vtkScalarsToColors*);
  vtkGetObjectMacro(LookupTable,vtkScalarsToColors);

  // Description:
  // Set the output format, the default is RGBA.
  vtkSetMacro(OutputFormat,int);
  vtkGetMacro(OutputFormat,int);
  void SetOutputFormatToRGBA() { this->OutputFormat = VTK_RGBA; };
  void SetOutputFormatToRGB() { this->OutputFormat = VTK_RGB; };
  void SetOutputFormatToLuminanceAlpha() { this->OutputFormat = VTK_LUMINANCE_ALPHA; };
  void SetOutputFormatToLuminance() { this->OutputFormat = VTK_LUMINANCE; };

  // Description:
  // Set the component to map for multi-component images (default: 0)
  vtkSetMacro(ActiveComponent,int);
  vtkGetMacro(ActiveComponent,int);

  // Description:
  // Use the alpha component of the input when computing the alpha component
  // of the output (useful when converting monochrome+alpha data to RGBA)
  vtkSetMacro(PassAlphaToOutput,int);
  vtkBooleanMacro(PassAlphaToOutput,int);
  vtkGetMacro(PassAlphaToOutput,int);

  // Description:
  // We need to check the modified time of the lookup table too.
#ifdef VTK_HAS_MTIME_TYPE
  virtual vtkMTimeType GetMTime();
#else
  virtual unsigned long GetMTime();
#endif

protected:
  vtkImageMapToColors16();
  ~vtkImageMapToColors16();

  virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *);

  void ThreadedRequestData(vtkInformation *request,
                           vtkInformationVector **inputVector,
                           vtkInformationVector *outputVector,
                           vtkImageData ***inData, vtkImageData **outData,
                           int extent[6], int id);
```

```
00102
00103   virtual int RequestData(vtkInformation *request,
00104                           vtkInformationVector **inputVector,
00105                           vtkInformationVector *outputVector);
00106
00107   vtkScalarsToColors *LookupTable;
00108   int OutputFormat;
00109
00110   int ActiveComponent;
00111   int PassAlphaToOutput;
00112
00113   int DataWasPassed;
00114 private:
00115   vtkImageMapToColors16(const vtkImageMapToColors16&);  // Not implemented.
00116   void operator=(const vtkImageMapToColors16&);  // Not implemented.
00117 };
00118
00119 #endif
```

13.627 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkImageMapToColors.h"
```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



Classes

- class vtkImageMapToWindowLevelColors2

13.628 vtkImageMapToWindowLevelColors2.h

Go to the documentation of this file.

```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
```

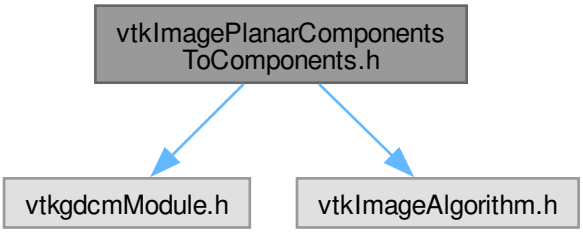
```
     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*=========================================================================

  Portions of this file are subject to the VTK Toolkit Version 3 copyright.

  Program:   Visualization Toolkit
  Module:    $RCSfile: vtkImageMapToWindowLevelColors2.h,v $

  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
  All rights reserved.
  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkImageMapToWindowLevelColors2 - map the input image through a lookup table and window / level it
// .SECTION Description
// The vtkImageMapToWindowLevelColors2 filter will take an input image of any
// valid scalar type, and map the first component of the image through a
// lookup table.  This resulting color will be modulated with value obtained
// by a window / level operation. The result is an image of type
// VTK_UNSIGNED_CHAR. If the lookup table is not set, or is set to NULL, then
// the input data will be passed through if it is already of type
// UNSIGNED_CHAR.
//
// .SECTION See Also
// vtkLookupTable vtkScalarsToColors

#ifndef VTKIMAGEMAPTOWINDOWLEVELCOLORS2_H
#define VTKIMAGEMAPTOWINDOWLEVELCOLORS2_H

#include "vtkgdcmModule.h"
#include "vtkImageMapToColors.h"

class VTKGDCM_EXPORT vtkImageMapToWindowLevelColors2 : public vtkImageMapToColors
{
public:
  static vtkImageMapToWindowLevelColors2 *New();
  vtkTypeMacro(vtkImageMapToWindowLevelColors2,vtkImageMapToColors);
  void PrintSelf(ostream& os, vtkIndent indent);

  // Description:
  // Set / Get the Window to use -> modulation will be performed on the
  // color based on (S - (L - W/2))/W where S is the scalar value, L is
  // the level and W is the window.
  vtkSetMacro( Window, double );
  vtkGetMacro( Window, double );

  // Description:
  // Set / Get the Level to use -> modulation will be performed on the
  // color based on (S - (L - W/2))/W where S is the scalar value, L is
  // the level and W is the window.
  vtkSetMacro( Level, double );
  vtkGetMacro( Level, double );

protected:
  vtkImageMapToWindowLevelColors2();
  ~vtkImageMapToWindowLevelColors2();

  virtual int RequestInformation (vtkInformation *, vtkInformationVector **, vtkInformationVector *);
  void ThreadedRequestData(vtkInformation *request,
                           vtkInformationVector **inputVector,
                           vtkInformationVector *outputVector,
                           vtkImageData ***inData, vtkImageData **outData,
                           int extent[6], int id);
  virtual int RequestData(vtkInformation *request,
                          vtkInformationVector **inputVector,
                          vtkInformationVector *outputVector);

  double Window;
  double Level;

private:
  vtkImageMapToWindowLevelColors2(const vtkImageMapToWindowLevelColors2&);  // Not implemented.
```

```
00089   void operator=(const vtkImageMapToWindowLevelColors2&);  // Not implemented.
00090 };
00091
00092 #endif
```

13.629 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



Classes

- class vtkImagePlanarComponentsToComponents

13.630 vtkImagePlanarComponentsToComponents.h

Go to the documentation of this file.

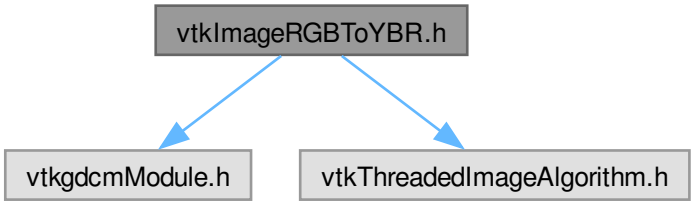
```
00001 /*=========================================================================
00002
00003   Program: GDCM (Grassroots DICOM). A DICOM library
00004
00005   Copyright (c) 2006-2011 Mathieu Malaterre
00006   All rights reserved.
00007   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
00008
00009      This software is distributed WITHOUT ANY WARRANTY; without even
00010      the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
00011      PURPOSE.  See the above copyright notice for more information.
00012
00013 =========================================================================*/
00014 /*=========================================================================
00015
00016   Portions of this file are subject to the VTK Toolkit Version 3 copyright.
00017
00018   Program:   Visualization Toolkit
00019   Module:    $RCSfile: vtkImagePlanarComponentsToComponents.h,v $
00020
00021   Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
00022   All rights reserved.
```

```
  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkImagePlanarComponentsToComponents - Converts planar comp to pixel comp
// .SECTION Description

// .SECTION See Also
// TODO: Can I make this filter threaded ?
// TODO: How do I handle the VTK-flipping (FileLowerLeft)?

#ifndef VTKIMAGEPLANARCOMPONENTSTOCOMPONENTS_H
#define VTKIMAGEPLANARCOMPONENTSTOCOMPONENTS_H

#include "vtkgdcmModule.h"
#include "vtkImageAlgorithm.h"

// everything is now handled within the vtkGDCMImageReader as Planar Configuration can not
// be externalized (conflict with file lower left)

#error do not use this class

//class VTKGDCM_EXPORT vtkImagePlanarComponentsToComponents : public vtkThreadedImageAlgorithm
class VTKGDCM_EXPORT vtkImagePlanarComponentsToComponents : public vtkImageAlgorithm
{
public:
  static vtkImagePlanarComponentsToComponents *New();
  //vtkTypeMacro(vtkImagePlanarComponentsToComponents,vtkThreadedImageAlgorithm);
  vtkTypeMacro(vtkImagePlanarComponentsToComponents,vtkImageAlgorithm);

  void PrintSelf(ostream& os, vtkIndent indent);

protected:
  vtkImagePlanarComponentsToComponents();
  ~vtkImagePlanarComponentsToComponents() {};

//  void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
//                        int ext[6], int id);
//  virtual int RequestInformation (vtkInformation *, vtkInformationVector**, vtkInformationVector *);
  virtual int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector *);

private:
  vtkImagePlanarComponentsToComponents(const vtkImagePlanarComponentsToComponents&);  // Not implemented.
  void operator=(const vtkImagePlanarComponentsToComponents&);  // Not implemented.
};

#endif
```

13.631 vtkImageRGBToYBR.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:



Classes

- class vtkImageRGBToYBR

13.632 vtkImageRGBToYBR.h

Go to the documentation of this file.

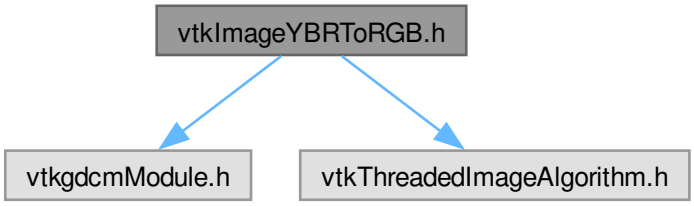
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*=========================================================================

  Portions of this file are subject to the VTK Toolkit Version 3 copyright.

  Program:   Visualization Toolkit
  Module:    $RCSfile: vtkImageRGBToYBR.h,v $

  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
  All rights reserved.
  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkImageRGBToYBR - Converts YBR components to RGB.
// .SECTION Description
// For each pixel with hue, saturation and value components this filter
// outputs the color coded as red, green, blue.  Output type must be the same
// as input type.

// .SECTION See Also
// vtkImageRGBToHSV

#ifndef VTKIMAGERGBTOYBR_H
```

```
#define VTKIMAGERGBTOYBR_H

#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"

class VTKGDCM_EXPORT vtkImageRGBToYBR : public vtkThreadedImageAlgorithm
{
public:
  static vtkImageRGBToYBR *New();
  vtkTypeMacro(vtkImageRGBToYBR,vtkThreadedImageAlgorithm);

  void PrintSelf(ostream& os, vtkIndent indent);

protected:
  vtkImageRGBToYBR();
  ~vtkImageRGBToYBR() {};

  void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
                        int ext[6], int id);
private:
  vtkImageRGBToYBR(const vtkImageRGBToYBR&);  // Not implemented.
  void operator=(const vtkImageRGBToYBR&);  // Not implemented.
};

#endif
```

13.633 vtkImageYBRToRGB.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageYBRToRGB.h:



Classes

- class vtkImageYBRToRGB

13.634 vtkImageYBRToRGB.h

Go to the documentation of this file.

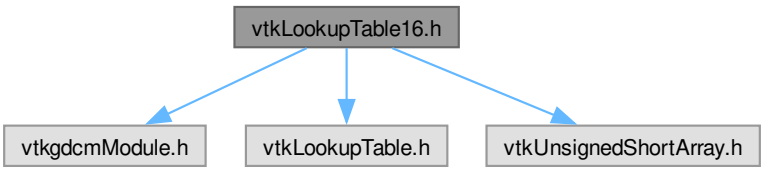
```
/*=========================================================================

```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*=========================================================================

  Portions of this file are subject to the VTK Toolkit Version 3 copyright.

  Program:   Visualization Toolkit
  Module:    $RCSfile: vtkImageYBRToRGB.h,v $

  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
  All rights reserved.
  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkImageYBRToRGB - Converts YBR components to RGB.
// .SECTION Description
// For each pixel with hue, saturation and value components this filter
// outputs the color coded as red, green, blue.  Output type must be the same
// as input type.

// .SECTION See Also
// vtkImageRGBToHSV

#ifndef VTKIMAGEYBRTORGB_H
#define VTKIMAGEYBRTORGB_H

#include "vtkgdcmModule.h"
#include "vtkThreadedImageAlgorithm.h"

class VTKGDCM_EXPORT vtkImageYBRToRGB : public vtkThreadedImageAlgorithm
{
public:
  static vtkImageYBRToRGB *New();
  vtkTypeMacro(vtkImageYBRToRGB,vtkThreadedImageAlgorithm);

  void PrintSelf(ostream& os, vtkIndent indent);

protected:
  vtkImageYBRToRGB();
  ~vtkImageYBRToRGB() {};

  void ThreadedExecute (vtkImageData *inData, vtkImageData *outData,
                        int ext[6], int id);
private:
  vtkImageYBRToRGB(const vtkImageYBRToRGB&);  // Not implemented.
  void operator=(const vtkImageYBRToRGB&);  // Not implemented.
};

#endif
```

13.635 vtkLookupTable16.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkLookupTable.h"
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



Classes

- class vtkLookupTable16

13.636 vtkLookupTable16.h

Go to the documentation of this file.

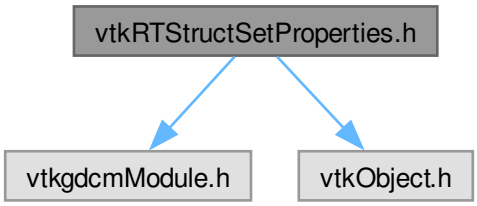
```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*=========================================================================

  Portions of this file are subject to the VTK Toolkit Version 3 copyright.

  Program:   Visualization Toolkit
  Module:    $RCSfile: vtkLookupTable16.h,v $

  Copyright (c) Ken Martin, Will Schroeder, Bill Lorensen
  All rights reserved.
  See Copyright.txt or http://www.kitware.com/Copyright.htm for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkLookupTable16 -
// .SECTION Description
//
// .SECTION Caveats
//
// .SECTION See Also
// vtkLookupTable

#ifndef VTKLOOKUPTABLE16_H
#define VTKLOOKUPTABLE16_H

#include "vtkgdcmModule.h"
#include "vtkLookupTable.h"
#include "vtkUnsignedShortArray.h"
```

```
00044
00045 class VTKGDCM_EXPORT vtkLookupTable16 : public vtkLookupTable
00046 {
00047 public:
00048   static vtkLookupTable16 *New();
00049
00050   vtkTypeMacro(vtkLookupTable16,vtkLookupTable);
00051   void PrintSelf(ostream& os, vtkIndent indent);
00052
00053   void Build();
00054
00055   void SetNumberOfTableValues(vtkIdType number);
00056
00057   unsigned char *WritePointer(const vtkIdType id, const int number);
00058
00059   unsigned short *GetPointer(const vtkIdType id) {
00060     return this->Table16->GetPointer(4*id); };
00061
00062 protected:
00063   vtkLookupTable16(int sze=256, int ext=256);
00064   ~vtkLookupTable16();
00065
00066   vtkUnsignedShortArray *Table16;
00067
00068 void MapScalarsThroughTable2(void *input,
00069                                             unsigned char *output,
00070                                             int inputDataType,
00071                                             int numberOfValues,
00072                                             int inputIncrement,
00073                                             int outputFormat);
00074
00075 private:
00076   vtkLookupTable16(const vtkLookupTable16&);  // Not implemented.
00077   void operator=(const vtkLookupTable16&);  // Not implemented.
00078 };
00079
00080 //-----------------------------------------------------------------------------
00081 inline unsigned char *vtkLookupTable16::WritePointer(const vtkIdType id,
00082                                                      const int number)
00083 {
00084   //this->InsertTime.Modified();
00085   return (unsigned char*)this->Table16->WritePointer(4*id,4*number);
00086 }
00087
00088 #endif
```

13.637 vtkRTStructSetProperties.h File Reference

```
#include "vtkgdcmModule.h"
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



Classes

- class vtkRTStructSetProperties

13.638 vtkRTStructSetProperties.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// .NAME vtkRTStructSetProperties - some rtstruct properties.
// .SECTION Description
//
// .SECTION See Also
// vtkGDCMPolyDataReader vtkGDCMPolyDataWriter

#ifndef VTKRTSTRUCTSETPROPERTIES_H
#define VTKRTSTRUCTSETPROPERTIES_H

#include "vtkgdcmModule.h"
#include "vtkObject.h"

class vtkRTStructSetPropertiesInternals;

class VTKGDCM_EXPORT vtkRTStructSetProperties : public vtkObject
{
public:
  static vtkRTStructSetProperties *New();
  vtkTypeMacro(vtkRTStructSetProperties,vtkObject);
  void PrintSelf(ostream& os, vtkIndent indent);

  // Description:
  // Convenience method to reset all fields to an empty string/value
  virtual void Clear();

  // Description:
  //
  vtkSetStringMacro(StructureSetLabel);
  vtkGetStringMacro(StructureSetLabel);

  vtkSetStringMacro(StructureSetName);
  vtkGetStringMacro(StructureSetName);

  vtkSetStringMacro(StructureSetDate);
  vtkGetStringMacro(StructureSetDate);

  vtkSetStringMacro(StructureSetTime);
  vtkGetStringMacro(StructureSetTime);

  vtkSetStringMacro(SOPInstanceUID);
  vtkGetStringMacro(SOPInstanceUID);

  vtkSetStringMacro(StudyInstanceUID);
  vtkGetStringMacro(StudyInstanceUID);

  vtkSetStringMacro(SeriesInstanceUID);
  vtkGetStringMacro(SeriesInstanceUID);

  vtkSetStringMacro(ReferenceSeriesInstanceUID);
  vtkGetStringMacro(ReferenceSeriesInstanceUID);

  vtkSetStringMacro(ReferenceFrameOfReferenceUID);
  vtkGetStringMacro(ReferenceFrameOfReferenceUID);
```

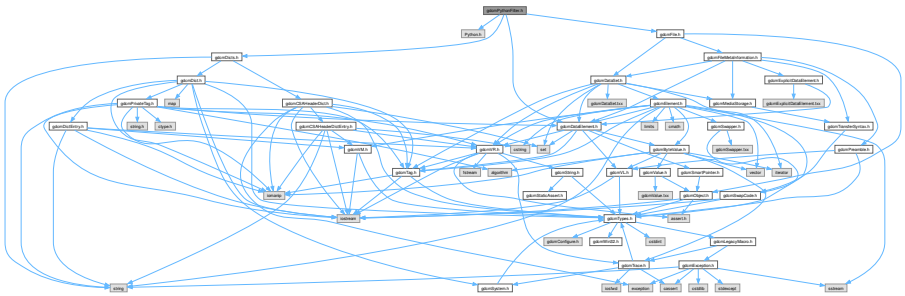
```

  // Description:
  // Copy the contents of p to this instance.
  virtual void DeepCopy(vtkRTStructSetProperties *p);

  void AddContourReferencedFrameOfReference( vtkIdType pdnum, const char *classuid , const char *
instanceuid );
  const char *GetContourReferencedFrameOfReferenceClassUID( vtkIdType pdnum, vtkIdType id );
  const char *GetContourReferencedFrameOfReferenceInstanceUID( vtkIdType pdnum, vtkIdType id );
  vtkIdType GetNumberOfContourReferencedFrameOfReferences();
  vtkIdType GetNumberOfContourReferencedFrameOfReferences(vtkIdType pdnum);

  void AddReferencedFrameOfReference( const char *classuid , const char * instanceuid );
  const char *GetReferencedFrameOfReferenceClassUID( vtkIdType id );
  const char *GetReferencedFrameOfReferenceInstanceUID( vtkIdType id );
  vtkIdType GetNumberOfReferencedFrameOfReferences();

  void AddStructureSetROI( int roinumber,
    const char* reframerefuid,
    const char* roiname,
    const char* ROIGenerationAlgorithm,
    const char* ROIDescription = 0
  );
  void AddStructureSetROIObservation( int refnumber,
    int observationnumber,
    const char *rtroiinterpretedtype,
    const char *roiinterpreter,
    const char *roiobservationlabel = 0
  );

  vtkIdType GetNumberOfStructureSetROIs();
  int GetStructureSetObservationNumber(vtkIdType id);
  int GetStructureSetROINumber(vtkIdType id);
  const char *GetStructureSetROIRefFrameRefUID(vtkIdType);
  const char *GetStructureSetROIName(vtkIdType);
  const char *GetStructureSetROIGenerationAlgorithm(vtkIdType);
  const char *GetStructureSetROIDescription(vtkIdType id);
  const char *GetStructureSetRTROIInterpretedType(vtkIdType id);
  const char *GetStructureSetROIObservationLabel(vtkIdType id);

protected:
  vtkRTStructSetProperties();
  ~vtkRTStructSetProperties();

  char *StructureSetLabel;
  char *StructureSetName;
  char *StructureSetDate;
  char *StructureSetTime;

  char *SOPInstanceUID;
  char *StudyInstanceUID;
  char *SeriesInstanceUID;

  char *ReferenceSeriesInstanceUID;
  char *ReferenceFrameOfReferenceUID;

  // Description:
  // PIMPL Encapsulation for STL containers
  //BTX
  vtkRTStructSetPropertiesInternals *Internals;
  //ETX

private:
  vtkRTStructSetProperties(const vtkRTStructSetProperties&); // Not implemented.
  void operator=(const vtkRTStructSetProperties&); // Not implemented.
};

#endif
```

13.639 gdcmPythonFilter.h File Reference

```
#include <Python.h>
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
```

`#include "gdcmFile.h"`

Include dependency graph for gdcmPythonFilter.h:



Classes

- class gdcm::PythonFilter

  PythonFilter PythonFilter is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a DataElement into a string, typically this is a nice feature to have for wrapped language.

Namespaces

- namespace gdcm

13.640 gdcmPythonFilter.h

Go to the documentation of this file.

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#ifndef GDCMPYTHONFILTER_H
#define GDCMPYTHONFILTER_H

#include <Python.h>

#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"

namespace gdcm
{

class GDCM_EXPORT PythonFilter
{
public:
  PythonFilter();
  ~PythonFilter();

```

```
00037   void UseDictAlways(bool ) {}
00038
00039   // Allow user to pass in there own dicts
00040   void SetDicts(const Dicts &dicts);
00041
00042   // Convert to string the ByteValue contained in a DataElement
00043   PyObject *ToPyObject(const Tag& t) const;
00044
00045   void SetFile(const File& f);
00046   File &GetFile();
00047   const File &GetFile() const;
00048
00049 private:
00050   SmartPointer<File> F;
00051 };
00052
00053 } // end namespace gdcm
00054
00055 #endif //GDCMPYTHONFILTER_H
```

Chapter 14

Examples

14.1 TestByteSwap.cxx

This is a C++ example on how to use gdcm::ByteSwap

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.h"

#include <cstring> // memcpy

int myfunc()
{
  char vl_str[4];
  const char raw[] = "\000\000\000\004";
  memcpy(vl_str, raw, 4);
  uint32_t vl;
  memcpy(&vl, vl_str, 4);
  gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem(&vl, gdcm::SwapCode::BigEndian, 1);
  if( vl != 0x00000004 )
    {
    std::cerr « std::hex « "vl: " « vl « std::endl;
    return 1;
    }

  gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(vl, gdcm::SwapCode::LittleEndian);
  if( vl != 0x00000004 )
    {
    std::cerr « std::hex « "vl: " « vl « std::endl;
    return 1;
    }

  gdcm::ByteSwap<uint32_t>::SwapFromSwapCodeIntoSystem(vl, gdcm::SwapCode::BigEndian);
  if( vl != 0x4000000 )
    {
    std::cerr « std::hex « "vl: " « vl « std::endl;
    return 1;
    }

  return 0;
```

```
}

int TestByteSwap(int , char *[])
{
  gdcm::SwapCode sc = gdcm::SwapCode::Unknown;
  if ( gdcm::ByteSwap<uint16_t>::SystemIsBigEndian() )
    {
    sc = gdcm::SwapCode::BigEndian;
    }
  else if ( gdcm::ByteSwap<uint16_t>::SystemIsLittleEndian() )
    {
    sc = gdcm::SwapCode::LittleEndian;
    }
  if( sc == gdcm::SwapCode::Unknown )
    {
    std::cerr « "unk" « std::endl;
    return 1;
    }

  //std::cout « "sc: " « sc « std::endl;

  uint16_t t = 0x1234;
  gdcm::ByteSwap<uint16_t>::SwapFromSwapCodeIntoSystem(t, sc);
  if( sc == gdcm::SwapCode::BigEndian )
    {
    if( t != 0x3412 )
      {
      std::cerr « std::hex « "t: " « t « std::endl;
      return 1;
      }
    // ok test pass rest value to old one
    t = 0x1234;
    }
  else if ( sc == gdcm::SwapCode::LittleEndian )
    {
    if( t != 0x1234 )
      {
      std::cerr « std::hex « "t: " « t « std::endl;
      return 1;
      }
    }

  union { char n[2]; uint16_t tn; } u16;
  memcpy(u16.n, &t, 2 );
  gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&u16.tn, sc, 1);
  uint16_t tn = u16.tn;
  if( sc == gdcm::SwapCode::BigEndian )
    {
    if( tn != 0x3412 )
      {
      std::cerr « std::hex « "tn: " « tn « std::endl;
      return 1;
      }
    // ok test pass rest value to old one
    t = 0x1234;
    }
  else if ( sc == gdcm::SwapCode::LittleEndian )
    {
    if( tn != 0x1234 )
      {
      std::cerr « std::hex « "tn: " « tn « std::endl;
      return 1;
      }
    }
  gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(&u16.tn, gdcm::SwapCode::BigEndian, 1);
  tn = u16.tn;
  if( sc == gdcm::SwapCode::LittleEndian )
    {
    if( tn != 0x3412 )
      {
      std::cerr « std::hex « "tn: " « tn « std::endl;
      return 1;
      }
    }
  else if ( sc == gdcm::SwapCode::BigEndian )
    {
    if( tn != 0x1234 )
      {
      std::cerr « std::hex « "tn: " « tn « std::endl;
      return 1;
      }
```

```
    }

  if( myfunc() )
    {
    return 1;
    }

  uint16_t array[] = { 0x1234 };
  gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem(array,
    gdcm::SwapCode::BigEndian,1);
   if ( array[0] != 0x3412 )
     {
     std::cerr « std::hex « "array: " « array[0] « std::endl;
     return 1;
     }

  return 0;
}
```

14.2 PatchFile.cxx

This is a C++ example on how to use gdcm::Attribute

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * The image was a broken file where the Pixel Data element was 8 times too big
 * Apparently multiplying the BitsAllocated to 4 and multiplying the number of
 * frames by 2 would solve the problem
 *
 * This C++ code can be used to patch the header.
 */

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *f = argv[1];
  const char *out = argv[2];
  gdcm::Reader r;
  r.SetFileName( f );
  if( !r.Read() )
    {
    return 1;
    }

  gdcm::File &file = r.GetFile();
  gdcm::DataSet& ds = file.GetDataSet();
  // (0028,0100) US 16                                       #   2, 1 BitsAllocated
  // (0028,0101) US 16                                       #   2, 1 BitsStored
  // (0028,0102) US 15                                       #   2, 1 HighBit
  //
    {
    gdcm::Attribute<0x28,0x100> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    if( at.GetValue() != 8 )
```

```
      {
      return 1;
      }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
    }
    {
    gdcm::Attribute<0x28,0x101> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    if( at.GetValue() != 8 )
      {
      return 1;
      }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
    }
    {
    gdcm::Attribute<0x28,0x102> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    if( at.GetValue() != 7 )
      {
      return 1;
      }
    at.SetValue( 31 );
    ds.Replace( at.GetAsDataElement() );
    }
  // (0028,0008) IS [56]                                    #   2, 1 NumberOfFrames

    {
    gdcm::Attribute<0x28,0x8> at;
    at.SetFromDataElement( ds.GetDataElement( at.GetTag() ) );
    at.SetValue( at.GetValue() * 2 );
    ds.Replace( at.GetAsDataElement() );
    }

  gdcm::Writer w;
  w.SetFile( file );
  w.SetCheckFileMetaInformation( false );
  w.SetFileName( out );
  if( !w.Write() )
    {
    return 1;
    }

  // Now let's see if we can read it as an image:
  gdcm::ImageReader ir;
  ir.SetFileName( out );
  if(!ir.Read())
    {
    return 1;
    }
  gdcm::Image &image = ir.GetImage();
  unsigned long len = image.GetBufferLength();
  const gdcm::ByteValue *bv = ir.GetFile().GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010)
      ).GetByteValue();
  if( !bv || len != bv->GetLength() )
    {
    return 1;
    }
  std::cout « bv->GetLength() « " " « len « std::endl;

  std::cout « "Success to rewrite image !" « std::endl;
  image.Print( std::cout );
  return 0;
}
```

14.3 SimplePrint.cs

This is a C# example on how to use gdcm::SWIGDataSet

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
  Convertor convertor = new Convertor();
  int a = convertor.Convert<int>( some_int_blob );
  double b = convertor.Convert<double>( some_double_blob );
*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrint.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class SimplePrint
{
  public static void RecurseDataSet(File f, DataSet ds, string indent)
    {
    CSharpDataSet cds = new CSharpDataSet(ds);
    while(!cds.IsAtEnd())
      {
      DataElement de = cds.GetCurrent();
      // Compute VR from the toplevel file, and the currently processed dataset:
      VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

      if( vr.Compatible( new VR(VR.VRType.SQ) ) )
        {
        uint uvl = (uint)de.GetVL(); // Test cast is ok
        System.Console.WriteLine( indent + de.GetTag().toString() + ":" + uvl ); // why not ?
        //SequenceOfItems sq = de.GetSequenceOfItems();
        // GetValueAsSQ handle more cases than GetSequenceOfItems
        SmartPtrSQ sq = de.GetValueAsSQ();
        uint n = sq.GetNumberOfItems();
        for( uint i = 1; i <= n; i++) // item starts at 1, not 0
          {
          Item item = sq.GetItem( i );
          DataSet nested = item.GetNestedDataSet();
          RecurseDataSet( f, nested, indent + "  " );
          }
        }
      else
        {
        System.Console.WriteLine( indent + de.toString() );
        }
      cds.Next();
      }
    }

  public static int Main(string[] args)
    {
    string filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();

    RecurseDataSet( f, ds, "" );

    return 0;
    }
}
```

14.4 TestReader.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmFile.h"
#include "gdcmTesting.h"
#include "gdcmMediaStorage.h"

int TestRead(const char* filename, bool verbose = false)
{
  if( verbose )
    std::cout « "TestRead: " « filename « std::endl;

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if ( !reader.Read() )
    {
    std::cerr « "TestReadError: Failed to read: " « filename « std::endl;
    return 1;
    }

//commenting out the fmi and ds to avoid warnings
  //const gdcm::FileMetaInformation &h = reader.GetFile().GetHeader();
  //std::cout « h « std::endl;

  //const gdcm::DataSet &ds = reader.GetFile().GetDataSet();
  //std::cout « ds « std::endl;

  const char *ref = gdcm::Testing::GetMediaStorageFromFile(filename);
  gdcm::MediaStorage ms;
  ms.SetFromFile( reader.GetFile() );
  if( !ref )
    {
    std::cerr « "TestReadError: Missing MediaStorage: " « filename « std::endl;
    std::cerr « "It should be: " « ms « std::endl;
    return 1;
    }

  if( ms.IsUndefined() && ref && *ref != 0 )
    {
    std::cerr « "TestReadError: MediaStorage: " « filename « std::endl;
    std::cerr « "It should be instead: " « ref « std::endl;
    return 1;
    }

  // Make sure it is the right one:

  if( ref && *ref != 0 && ms != gdcm::MediaStorage::GetMSType(ref) )
    {
    std::cerr « "Error: Found MediaStorage: " « ms « " for " « filename « std::endl;
    std::cerr « "It should be instead: " « ref « std::endl;
    return 1;
    }

  return 0;
}

int TestReader(int argc, char *argv[])
{
  if( argc == 2 )
    {
    const char *filename = argv[1];
    return TestRead(filename, true);
    }

  // else
  gdcm::Trace::DebugOff();
```

```
  gdcm::Trace::WarningOff();
  int r = 0, i = 0;
  const char *filename;
  const char * const *filenames = gdcm::Testing::GetFileNames();
  while( (filename = filenames[i]) )
    {
    r += TestRead( filename );
    ++i;
    }

  return r;
}
```

14.5 TestReader.py

This is a C++ example on how to use gdcm::Reader

```
00001
00014
00015 import os,sys
00016 import gdcm
00017
00018 def TestRead(filename, verbose = False):
00019   r = gdcm.Reader()
00020   r.SetFileName( filename )
00021   success = r.Read()
00022   #if verbose: print r.GetFile()
00023   if verbose: print(r.GetFile().GetDataSet())
00024   return success
00025
00026 if __name__ == "__main__":
00027   success = 0
00028   try:
00029     filename = os.sys.argv[1]
00030     success += TestRead( filename, True )
00031   except:
00032     # loop over all files:
00033     gdcm.Trace.DebugOff()
00034     gdcm.Trace.WarningOff()
00035     t = gdcm.Testing()
00036     nfiles = t.GetNumberOfFileNames()
00037     for i in range(0,nfiles):
00038       filename = t.GetFileName(i)
00039       success += TestRead( filename )
00040
00041
00042   # Test succeed ?
00043   sys.exit(success == 0)
```

14.6 DecompressJPEGFile.cs

This is a C# example on how to use gdcm::SequenceOfFragments

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
```

```
 * $ mono bin/DecompressJPEGFile.exe somejpegfile.jpg
 */
using System;
using gdcm;

public class DecompressJPEGFile
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    System.IO.FileStream infile =
      new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
    uint fsize = gdcm.PosixEmulation.FileSize(file1);

    byte[] jstream  = new byte[fsize];
    infile.Read(jstream, 0 , jstream.Length);

    Trace.DebugOn();
    Image image = new Image();
    image.SetNumberOfDimensions( 2 ); // important for now
    DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

    // DO NOT set a ByteValue here, JPEG is a particular kind of encapsulated syntax
    // in which can one cannot use a simple byte array for storage. Instead, see
    // gdcm.SequenceOfFragments
    //pixeldata.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length ) );

    // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
    SmartPtrFrag sq = SequenceOfFragments.New();
    Fragment frag = new Fragment();
    frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
    // Single file => single fragment
    sq.AddFragment( frag );
    // Pass by reference:
    pixeldata.SetValue( sq.__ref__() );

    // insert:
    image.SetDataElement( pixeldata );

    // JPEG use YBR to achieve better compression ratio by default (not RGB)
    // FIXME hardcoded:
    PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.YBR_FULL );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(3,8,8,7);
    image.SetPixelFormat( pixeltype );

    // FIXME hardcoded:
    image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 ) );
    image.SetDimension(0, 692);
    image.SetDimension(1, 721);

    // Decompress !
    byte[] decompressedData = new byte[(int)image.GetBufferLength()];
    image.GetBuffer(decompressedData);

    // Write out the decompressed bytes
    System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.raw",
        System.IO.FileMode.Create))
      {
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write(decompressedData);
      }


    return 0;
    }
}
```

14.7 ManipulateFile.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```
  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ManipulateFile.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class ManipulateFile
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    string file2 = args[1];
    Reader reader = new Reader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    Anonymizer ano = new Anonymizer();
    ano.SetFile( reader.GetFile() );
    ano.RemovePrivateTags();
    ano.RemoveGroupLength();
    Tag t = new Tag(0x10,0x10);
    ano.Replace( t, "GDCM^Csharp^Test^Hello^World" );

    UIDGenerator g = new UIDGenerator();
    ano.Replace( new Tag(0x0008,0x0018), g.Generate() );
    ano.Replace( new Tag(0x0020,0x000d), g.Generate() );
    ano.Replace( new Tag(0x0020,0x000e), g.Generate() );
    ano.Replace( new Tag(0x0020,0x0052), g.Generate() );

    Writer writer = new Writer();
    writer.SetFileName( file2 );
    writer.SetFile( ano.GetFile() );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }

    return 0;
    }
}
```

14.8 ClinicalTrialIdentificationWorkflow.cs

This is a C# example on how to use Anonymizer

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.
```

```
==========================================================================*/

/*
 * Typical usage on UNIX:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ClinicalTrialIdentificationWorkflow.exe input_dir output_dir
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void StartFilter() {
    System.Console.WriteLine( "This is my start" );
  }
  protected override void EndFilter(){
    System.Console.WriteLine( "This is my end" );
  }
  protected override void ShowProgress(Subject caller, Event evt){
    ProgressEvent pe = ProgressEvent.Cast(evt);
    System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
  }
  protected override void ShowIteration(){
    System.Console.WriteLine( "This is my iteration" );
  }
  protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 *  http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 *  System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 *  System.Type type = evt.GetType();
 *  System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 *  System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
 *  System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
    AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
    if( ae != null )
      {
      Tag t = ae.GetTag();
      System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
      }
    else
      {
      System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
      }
  }
  protected override void ShowAbort(){
    System.Console.WriteLine( "This is my abort" );
  }
}

public class ClinicalTrialIdentificationWorkflow
{
  public static bool ProcessOneFile( gdcm.Anonymizer ano , string filename, string outfilename )
    {
    Reader reader = new Reader();
    reader.SetFileName( filename );
    bool ret = reader.Read();
    if( !ret )
      {
      return false;
      }
    // Pass in the file:
    ano.SetFile( reader.GetFile() );

    // First step, let's protect all Patient information as per
    // PS 3.15 / E.1 / Basic Application Level Confidentiality Profile
    if( !ano.BasicApplicationLevelConfidentialityProfile() )
      {
      return false;
      }

    // Now let's pass in all Clinical Trial fields
    // PS 3.3 - 2008 / C.7.1.3 Clinical Trial Subject Module
    /*
    Clinical Trial Sponsor Name (0012,0010) 1 The name of the clinical trial sponsor. See C.7.1.3.1.1.
    Clinical Trial Protocol ID (0012,0020) 1 Identifier for the noted protocol. See C.7.1.3.1.2.
    Clinical Trial Protocol Name (0012,0021) 2 The name of the clinical trial protocol. See C.7.1.3.1.3.
    Clinical Trial Site ID (0012,0030) 2 The identifier of the site responsible for submitting clinical trial
```

```
    data. See C.7.1.3.1.4.
  Clinical Trial Site Name (0012,0031) 2 Name of the site responsible for submitting clinical trial data. See
    C.7.1.3.1.5
  Clinical Trial Subject ID (0012,0040) 1C The assigned identifier for the clinical trial subject. See
    C.7.1.3.1.6. Shall be present if Clinical Trial Subject Reading ID (0012,0042) is absent. May be present
    otherwise.
  Clinical Trial Subject Reading ID (0012,0042) 1C Identifies the subject for blinded evaluations. Shall be
    present if Clinical Trial Subject ID (0012,0040) is absent.  May be present otherwise. See C.7.1.3.1.7.
   */
  ano.Replace( new gdcm.Tag(0x0012,0x0010), "MySponsorName");
  ano.Replace( new gdcm.Tag(0x0012,0x0020), "MyProtocolID");
  ano.Replace( new gdcm.Tag(0x0012,0x0021), "MyProtocolName");
  ano.Replace( new gdcm.Tag(0x0012,0x0030), "MySiteId");
  ano.Replace( new gdcm.Tag(0x0012,0x0031), "MySiteName");
  ano.Replace( new gdcm.Tag(0x0012,0x0040), "MySponsorId");
  ano.Replace( new gdcm.Tag(0x0012,0x0050), "MyTPId");
  ano.Replace( new gdcm.Tag(0x0012,0x0051), "MyTPDescription");

  // The following two are not required as they are guaranteed to be filled in by the
  // Basic Application Level Confidentiality Profile. Only override if you understand what
  // you are doing
  //ano.Replace( new gdcm.Tag(0x0012,0x0062), "YES");
  //ano.Replace( new gdcm.Tag(0x0012,0x0063), "My Super Duper Anonymization Overload");

  // We might be generating a subdirectory. Let's make sure the subdir exist:
  gdcm.Filename fn = new gdcm.Filename( outfilename );
  string subdir = fn.GetPath();
  if( !gdcm.PosixEmulation.MakeDirectory( subdir ) )
    {
    return false;
    }

  gdcm.FileMetaInformation fmi = ano.GetFile().GetHeader();
  // The following three lines make sure to regenerate any value:
  fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
  fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
  fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

  Writer writer = new Writer();
  writer.SetFileName( outfilename );
  writer.SetFile( ano.GetFile() );
  ret = writer.Write();
  if( !ret )
    {
    return false;
    }

  return true;
  }

public static int Main(string[] args)
  {
  gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My ClinicalTrial App" );

  // http://www.oid-info.com/get/1.3.6.1.4.17434
  string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
  gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
  System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

  gdcm.Global global = gdcm.Global.GetInstance();
  if( !global.LoadResourcesFiles() )
    {
    System.Console.WriteLine( "Could not LoadResourcesFiles" );
    return 1;
    }

  if( args.Length != 2 )
    {
    System.Console.WriteLine( "Usage:" );
    System.Console.WriteLine( "ClinicalTrialIdentificationWorkflow input_dir output_dir" );
    return 1;
    }
  string dir1 = args[0];
  string dir2 = args[1];

  // Check input is valid:
  if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
    {
    System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
    return 1;
    }
```

```
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
      {
      System.Console.WriteLine( "Output directory: " + dir2 + " does not exist. Sorry" );
      return 1;
      }

    // Recursively search all file within this toplevel directory:
    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;

    // Let's use the pre-shipped certificate of GDCM.
    string certpath = gdcm.Filename.Join(gdcm.Testing.GetSourceDirectory(),
      "/Testing/Source/Data/certificate.pem" );
    gdcm.CryptoFactory fact = gdcm.CryptoFactory.GetFactoryInstance();
    gdcm.CryptographicMessageSyntax cms = fact.CreateCMSProvider();
    if( !cms.ParseCertificateFile( certpath ) )
      {
      System.Console.WriteLine( "PEM Certificate : " + certpath + " could not be read. Sorry" );
      return 1;
      }

    //Anonymizer ano = new Anonymizer();
    // A reference to an actual C++ instance is required here:
    SmartPtrAno sano = Anonymizer.New();
    Anonymizer ano = sano.__ref__();

    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano, "Anonymizer");
    MyWatcher watcher = new MyWatcher(ano);

    // Explicitly specify the Cryptographic Message Syntax to use:
    ano.SetCryptographicMessageSyntax( cms );

    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
      {
      string filename = filenames[ (int)i ];
      string outfilename = filename.Replace( dir1, dir2 );
      System.Console.WriteLine( "Filename: " + filename );
      System.Console.WriteLine( "Out Filename: " + outfilename );
      if( !ProcessOneFile( ano , filename, outfilename ) )
        {
        System.Console.WriteLine( "Could not process filename: " + filename );
        return 1;
        }
      }

    return 0;
    }
}
```

14.9 GenerateDICOMDIR.cs

This is a C# example on how to use DICOMDIRGenerator

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example to show how to use DICOMDIRGenerator
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
```

```
 * $ mono bin/GenerateDICOMDIR.exe path output_filename
 */
using System;
using gdcm;

public class GenerateDICOMDIR
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    string outfilename = args[1];

    Directory d = new Directory();
    uint nfiles = d.Load( directory, true );
    if(nfiles == 0) return 1;
    //System.Console.WriteLine( "Files:\n" + d.toString() );

    // Implement fast path ?
    // Scanner s = new Scanner();

    string descriptor = "My_Descriptor";
    FilenamesType filenames = d.GetFilenames();

    gdcm.DICOMDIRGenerator gen = new DICOMDIRGenerator();
    gen.SetFilenames( filenames );
    gen.SetDescriptor( descriptor );
    if( !gen.Generate() )
      {
      return 1;
      }

    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "GenerateDICOMDIR" );
    gdcm.Writer writer = new Writer();
    writer.SetFile( gen.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
      {
      return 1;
      }

    return 0;
    }
}
```

14.10 GenFakeImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmFileDerivation.h"
#include "gdcmUIDGenerator.h"
//#include "gdcmImageChangePhotometricInterpretation.h"

/*
 * This example shows two things:
 * 1. How to create an image ex-nihilo
 * 2. How to use the gdcm.FileDerivation filter. This filter is meant to create "DERIVED" image
 * object. FileDerivation has a simple API where you can reference *all* the input image that have been
 * used to generate the image. The API also allows user to specify the purpose of reference (see CID 7202,
 * PS 3.16 - 2008), and the image derivation type (CID 7203, PS 3.16 - 2008).
 */
int main(int, char *[])
{
  // Step 1: Fake Image
```

```
gdcm::SmartPointer<gdcm::Image> im = new gdcm::Image;

char * buffer = new char[ 256 * 256 * 3];
char * p = buffer;
int b = 128;
//int ybr[3];
int ybr2[3];
//int rgb[3];

for(int r = 0; r < 256; ++r)
  for(int g = 0; g < 256; ++g)
    //for(int b = 0; b < 256; ++b)
    {
    //rgb[0] = r;
    //rgb[1] = g;
    //rgb[1] = 128;
    //rgb[2] = b;
    //ybr[0] = r;
    //ybr[1] = g;
    //ybr[1] = 128;
    //ybr[2] = b;

    ybr2[0] = r;
    ybr2[1] = g;
    ybr2[1] = 128;
    ybr2[2] = b;
    //gdcm::ImageChangePhotometricInterpretation::YBR2RGB(rgb, ybr);
    //gdcm::ImageChangePhotometricInterpretation::RGB2YBR(ybr2, rgb);
    *p++ = (char)ybr2[0];
    *p++ = (char)ybr2[1];
    *p++ = (char)ybr2[2];
    }

im->SetNumberOfDimensions( 2 );
im->SetDimension(0, 256 );
im->SetDimension(1, 256 );

im->GetPixelFormat().SetSamplesPerPixel(3);
//im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::RGB );
im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::YBR_FULL );

unsigned long l = im->GetBufferLength();
if( l != 256 * 256 * 3 )
  {
  return 1;
  }
gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
pixeldata.SetByteValue( buffer, (uint32_t)l );
delete[] buffer;
im->SetDataElement( pixeldata );

gdcm::UIDGenerator uid; // helper for uid generation

gdcm::SmartPointer<gdcm::File> file = new gdcm::File; // empty file

// Step 2: DERIVED object
gdcm::FileDerivation fd;
// For the purpose of this exercise we will pretend that this image is referencing
// two source image (we need to generate fake UID for that).
const char ReferencedSOPClassUID[] = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

// Again for the purpose of the exercise we will pretend that the image is a
// multiplanar reformat (MPR):
// CID 7202 Source Image Purposes of Reference
// {"DCM",121322,"Source image for image processing operation"},
fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
// CID 7203 Image Derivation
// { "DCM",113072,"Multiplanar reformatting" },
fd.SetDerivationCodeSequenceCodeValue( 113072 );
fd.SetFile( *file );
// If all Code Value are ok the filter will execute properly
if( !fd.Derive() )
  {
  std::cerr « "Sorry could not derive using input info" « std::endl;
  return 1;
  }

// We pass both :
// 1. the fake generated image
```

```
  // 2. the 'DERIVED' dataset object
  // to the writer.
  gdcm::ImageWriter w;
  w.SetImage( *im );
  w.SetFile( fd.GetFile() );

  // Set the filename:
  w.SetFileName( "ybr2.dcm" );
  if( !w.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.11 ReformatFile.cs

This is a C++ example on how to use FileDerivation

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ReformatFile.exe input.dcm output.dcm
 */
using System;
using gdcm;

public class ReformatFile
{
  public static int Main(string[] args)
    {
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Reformat App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

    string filename = args[0];
    string outfilename = args[1];

    Reader reader = new Reader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read: " + filename );
      return 1;
      }

    UIDGenerator uid = new UIDGenerator(); // helper for uid generation
    FileDerivation fd = new FileDerivation();
    // For the purpose of this exercise we will pretend that this image is referencing
    // two source image (we need to generate fake UID for that).
    string ReferencedSOPClassUID = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
    fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
    fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

    // Again for the purpose of the exercise we will pretend that the image is a
```

```
    // multiplanar reformat (MPR):
    // CID 7202 Source Image Purposes of Reference
    // {"DCM",121322,"Source image for image processing operation"},
    fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
    // CID 7203 Image Derivation
    // { "DCM",113072,"Multiplanar reformatting" },
    fd.SetDerivationCodeSequenceCodeValue( 113072 );
    fd.SetFile( reader.GetFile() );
    // If all Code Value are ok the filter will execute properly
    if( !fd.Derive() )
      {
      return 1;
      }

    gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

    Writer writer = new Writer();
    writer.SetFileName( outfilename );
    writer.SetFile( fd.GetFile() );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write: " + outfilename );
      return 1;
      }


    return 0;
    }
}
```

14.12 DecompressImage.cs

This is a C# example on how to use Image

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcmData/012345.002.050.dcm decompress.dcm
 */
using System;
using gdcm;

public class DecompressImage
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    string file2 = args[1];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    // check that one can access a Fragment from C#:
```

```
    var de = reader.GetFile().GetDataSet().GetDataElement(new Tag(0x7fe0, 0x0010));
    var sq = de.GetSequenceOfFragments();
    sq.GetFragment(0);

    Image image = new Image();
    Image ir = reader.GetImage();

    image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );

    //Just for fun:
    //int dircos =  ir.GetDirectionCosines();
    //t = gdcm.Orientation.GetType(dircos);
    //int l = gdcm.Orientation.GetLabel(t);
    //System.Console.WriteLine( "Orientation label:" + l );

    // Set the dimensions,
    // 1. either one at a time
    //image.SetDimension(0, ir.GetDimension(0) );
    //image.SetDimension(1, ir.GetDimension(1) );

    // 2. the array at once
    uint[] dims = {0, 0};
    // Just for fun let's invert the dimensions:
    dims[0] = ir.GetDimension(1);
    dims[1] = ir.GetDimension(0);
    ir.SetDimensions( dims );

    PixelFormat pixeltype = ir.GetPixelFormat();
    image.SetPixelFormat( pixeltype );

    PhotometricInterpretation pi = ir.GetPhotometricInterpretation();
    image.SetPhotometricInterpretation( pi );

    DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
    byte[] str1 = new byte[ ir.GetBufferLength()];
    ir.GetBuffer( str1 );
    //System.Console.WriteLine( ir.GetBufferLength() );
    pixeldata.SetByteValue( str1, new VL( (uint)str1.Length ) );
    //image.SetDataElement( pixeldata );
    ir.SetDataElement( pixeldata );


    ImageWriter writer = new ImageWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( ir );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }

    return 0;
    }
}
```

14.13 StandardizeFiles.cs

This is a C++ example on how to use ImageChangeTransferSyntax

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
```

```
 * Simple C# example to show how one would 'Standardize' a DICOM File-Set
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/StandardizeFiles.exe input_path output_path
 */
using System;
using gdcm;

public class StandardizeFiles
{
  public static bool ProcessOneFile( string filename, string outfilename )
    {
    PixmapReader reader = new PixmapReader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read: " + filename );
      return false;
      }

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetForce( false ); // do we really want to recompress when input is already compressed in same alg ?
    change.SetCompressIconImage( false ); // Keep it simple
    change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEG2000Lossless ) );
    change.SetInput( reader.GetPixmap() );
    if( !change.Change() )
      {
      System.Console.WriteLine( "Could not change: " + filename );
      return false;
      }

    gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
    // The following three lines make sure to regenerate any value:
    fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
    fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

    PixmapWriter writer = new PixmapWriter();
    writer.SetFileName( outfilename );
    writer.SetFile( reader.GetFile() );
    gdcm.Pixmap pixout = ((PixmapToPixmapFilter)change).GetOutput();

    writer.SetPixmap( pixout );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write: " + outfilename );
      return false;
      }

    return true;
    }

  public static int Main(string[] args)
    {
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Standardize App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
      {
      System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
      return 1;
      }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
      {
      System.Console.WriteLine( "Output directory: " + dir2 + " does not exist. Sorry" );
      return 1;
      }

    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
```

```
    if(nfiles == 0) return 1;

    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
      {
      string filename = filenames[ (int)i ];
      string outfilename = filename.Replace( dir1, dir2 );
      System.Console.WriteLine( "Filename: " + filename );
      System.Console.WriteLine( "Out Filename: " + outfilename );
      if( !ProcessOneFile( filename, outfilename ) )
        {
        System.Console.WriteLine( "Could not process filename: " + filename );
        //return 1;
        }
      }


    return 0;
    }
}
```

14.14 ScanDirectory.cs

This is a C# example on how to use Scanner

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ bin/ScanDirectory.exe /path/to/gdcmData/
 */
using System;
using gdcm;

// We will print each filename being processed
public class MyWatcher : SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void ShowFileName(Subject caller, Event evt){
    FileNameEvent fne = FileNameEvent.Cast(evt);
    if( fne != null )
      {
      string fn = fne.GetFileName();
      System.Console.WriteLine( "This is my Scanner. Processing FileName: " + fn );
      }
    else
      {
      System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
      }
  }
}

public class ScanDirectory
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    Tag t = new Tag(0x8,0x80);

    Directory d = new Directory();
    uint nfiles = d.Load( directory );
    if(nfiles == 0) return 1;
```

```
    //System.Console.WriteLine( "Files:\n" + d.toString() );

    // Use a StrictScanner, need to use a reference to pass the C++ pointer to
    // MyWatcher implementation
    SmartPtrStrictScan sscan = StrictScanner.New();
    StrictScanner s = sscan.__ref__();
    MyWatcher watcher = new MyWatcher(s);

    s.AddTag( t );
    bool b = s.Scan( d.GetFilenames() );
    if(!b) return 1;

    for(int i = 0; i < (int)nfiles; ++i)
      {
      if( !s.IsKey( d.GetFilenames()[i] ) )
        {
        System.Console.WriteLine( "File is not DICOM or could not be read: " + d.GetFilenames()[i] );
        }
      }

    System.Console.WriteLine( "Scan:\n" + s.toString() );

    System.Console.WriteLine( "success" );
    return 0;
    }
}
```

14.15 BasicAnonymizer.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/BasicAnonymizer.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void StartFilter() {
    System.Console.WriteLine( "This is my start" );
  }
  protected override void EndFilter(){
    System.Console.WriteLine( "This is my end" );
  }
  protected override void ShowProgress(Subject caller, Event evt){
    ProgressEvent pe = ProgressEvent.Cast(evt);
    System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
  }
  protected override void ShowIteration(){
    System.Console.WriteLine( "This is my iteration" );
  }
  protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 *  http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 *  System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 *  System.Type type = evt.GetType();
 *  System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 *  System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
```

```
 *  System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
    AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
    if( ae != null )
      {
      Tag t = ae.GetTag();
      System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
      }
    else
      {
      System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
      }
  }
  protected override void ShowAbort(){
    System.Console.WriteLine( "This is my abort" );
  }
}

public class BasicAnonymizer
{
  public static int Main(string[] args)
    {
    gdcm.Global global = gdcm.Global.GetInstance();
    if( !global.LoadResourcesFiles() )
      {
      System.Console.WriteLine( "Could not LoadResourcesFiles" );
      return 1;
      }

    string file1 = args[0];
    string file2 = args[1];
    Reader reader = new Reader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    string certpath = gdcm.Filename.Join(gdcm.Testing.GetSourceDirectory(),
      "/Testing/Source/Data/certificate.pem" );
    gdcm.CryptoFactory fact = gdcm.CryptoFactory.GetFactoryInstance();
    gdcm.CryptographicMessageSyntax cms = fact.CreateCMSProvider();
    if( !cms.ParseCertificateFile( certpath ) )
      {
      return 1;
      }

    //Anonymizer ano = new Anonymizer();
    SmartPtrAno sano = Anonymizer.New();
    Anonymizer ano = sano.__ref__();

    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(ano, "Anonymizer");
    MyWatcher watcher = new MyWatcher(ano);

    ano.SetFile( reader.GetFile() );
    ano.SetCryptographicMessageSyntax( cms );
    if( !ano.BasicApplicationLevelConfidentialityProfile() )
      {
      return 1;
      }

    Writer writer = new Writer();
    writer.SetFileName( file2 );
    writer.SetFile( ano.GetFile() );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }

    return 0;
    }
}
```

14.16 BasicImageAnonymizer.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 */
using System;
using gdcm;

public class BasicImageAnonymizer
{
  public static int Main(string[] args)
    {
    string filename = args[0];

    // instantiate the reader:
    gdcm.ImageReader reader = new gdcm.ImageReader();
    reader.SetFileName( filename );

    if (!reader.Read()) return 1;

    Image ir = reader.GetImage();

    uint[] dims = {0, 0, 0};
    dims[0] = ir.GetDimension(0);
    dims[1] = ir.GetDimension(1);
    dims[2] = ir.GetDimension(2);
    System.Console.WriteLine( "Dim:" + dims[0] );
    System.Console.WriteLine( "Dim:" + dims[1] );
    System.Console.WriteLine( "Dim:" + dims[2] );

    // buffer to get the pixels
    byte[] buffer = new byte[ ir.GetBufferLength()];
    System.Console.WriteLine( "Dim:" + ir.GetBufferLength() );
    ir.GetBuffer( buffer );

    for (uint z = 0; z < dims[2]; z++)
      {
      for (uint y = 0; y < dims[1] / 2; y++) // only half Y
        {
        for (uint x = 0; x < dims[0] / 2; x++) // only half X
          {
          buffer[ (z * dims[1] + y) * dims[0] + x ] = 0; // works when pixel type == UINT8
          }
        }
      }

    DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
    pixeldata.SetByteValue( buffer, new VL( (uint)buffer.Length ) );
    ir.SetDataElement( pixeldata );
    ir.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.ExplicitVRLittleEndian ) );

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLSLossless ) );
    change.SetInput( ir );
    if( !change.Change() )
      {
      System.Console.WriteLine( "Could not change: " + filename );
      return 1;
      }

    ImageWriter writer = new ImageWriter();
    writer.SetFileName( "out.dcm" );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( change.GetOutput() );
    bool ret = writer.Write();
    if( !ret )
      {
```

```
      return 1;
      }


    return 0;
    }
}
```

14.17 Cleaner.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/Cleaner.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
  public MyWatcher(Subject s):base(s,"Override String"){}
  protected override void StartFilter() {
    System.Console.WriteLine( "This is my start" );
  }
  protected override void EndFilter(){
    System.Console.WriteLine( "This is my end" );
  }
  protected override void ShowProgress(Subject caller, Event evt){
    ProgressEvent pe = ProgressEvent.Cast(evt);
    System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
  }
  protected override void ShowIteration(){
    System.Console.WriteLine( "This is my iteration" );
  }
  protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 *  http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 *  System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 *  System.Type type = evt.GetType();
 *  System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 *  System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
 *  System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
    AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
    if( ae != null )
      {
      Tag t = ae.GetTag();
      System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
      }
    else
      {
      System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
      }
  }
  protected override void ShowAbort(){
    System.Console.WriteLine( "This is my abort" );
  }
}

public class Cleaner
```

```
{
  public static int Main(string[] args)
    {
    gdcm.Global global = gdcm.Global.GetInstance();
    if( !global.LoadResourcesFiles() )
      {
      System.Console.WriteLine( "Could not LoadResourcesFiles" );
      return 1;
      }

    string file1 = args[0];
    string file2 = args[1];
    Reader reader = new Reader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    SmartPtrCleaner scleaner = gdcm.Cleaner.New();
    gdcm.Cleaner cleaner = scleaner.__ref__();

    //SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(cleaner, "Anonymizer");
    MyWatcher watcher = new MyWatcher(cleaner);

    cleaner.SetFile( reader.GetFile() );
    cleaner.Empty( new gdcm.VR(gdcm.VR.VRType.PN) );
    gdcm.DPath dpath = new gdcm.DPath();
    dpath.ConstructFromString( "/0010,0010" );
    cleaner.Preserve( dpath );
    gdcm.Tag t1 = new gdcm.Tag(0x10, 0x30);
    cleaner.Empty( t1 );
    gdcm.PrivateTag pt0 = new gdcm.PrivateTag( new gdcm.Tag(0x29,0x60), "SIEMENS MEDCOM HEADER2" );
    cleaner.Remove( pt0 );
    gdcm.PrivateTag pt1 = new gdcm.PrivateTag( new gdcm.Tag(0x29,0x10), "SIEMENS CSA HEADER" );
    gdcm.PrivateTag pt2 = new gdcm.PrivateTag( new gdcm.Tag(0x29,0x20), "SIEMENS CSA HEADER" );
    cleaner.Scrub( pt1 );
    cleaner.Scrub( pt2 );
    if( !cleaner.Clean() )
      {
      return 1;
      }

    Writer writer = new Writer();
    writer.SetFileName( file2 );
    writer.SetFile( cleaner.GetFile() );
    ret = writer.Write();
    if( !ret )
      {
      return 1;
      }

    return 0;
    }
}
```

14.18 CompressLossyJPEG.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
 * $ mono bin/CompressLossyJPEG.exe input.dcm output.dcm
```

```
 */

using System;
using gdcm;

public class CompressLossyJPEG
{
  public static int Main(string[] args)
    {
    if( args.Length < 2 )
      {
      System.Console.WriteLine( " input.dcm output.dcm" );
      return 1;
      }
    string filename = args[0];
    string outfilename = args[1];

    ImageReader reader = new ImageReader();
    reader.SetFileName( filename );
    if( !reader.Read() )
      {
      System.Console.WriteLine( "Could not read: " + filename );
      return 1;
      }

    // The output of gdcm::Reader is a gdcm::File
    File file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    DataSet ds = file.GetDataSet();

    Image image = reader.GetImage();
    //image.Print( cout );

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    TransferSyntax targetts =  new TransferSyntax( TransferSyntax.TSType.JPEGBaselineProcess1 );
    change.SetTransferSyntax( targetts );

    // Setup our JPEGCodec, warning it should be compatible with JPEGBaselineProcess1
    JPEGCodec jpegcodec = new JPEGCodec();
    if( !jpegcodec.CanCode( targetts ) )
      {
      System.Console.WriteLine( "Something went really wrong, JPEGCodec cannot handle JPEGBaselineProcess1" );
      return 1;
      }
    jpegcodec.SetLossless( false );
    jpegcodec.SetQuality( 50 ); // poor quality !
    change.SetUserCodec( jpegcodec ); // specify the codec to use to the ImageChangeTransferSyntax

    change.SetInput( image );
    bool b = change.Change();
    if( !b )
      {
      System.Console.WriteLine( "Could not change the Transfer Syntax" );
      return 1;
      }

    ImageWriter writer = new ImageWriter();
    writer.SetImage( (gdcm.Image)change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write: " + outfilename );
      return 1;
      }

    return 0;
    }

}
```

14.19 DecompressImageMultiframe.cs

```
/*=========================================================================
```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
$ gdcminfo ~/Desktop/angiogram-06.dcm
MediaStorage is 1.2.840.10008.5.1.4.1.1.12.1 [X-Ray Angiographic Image Storage]
TransferSyntax is 1.2.840.10008.1.2.4.50 [JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8
      Bit Image Compression]
NumberOfDimensions: 3
Dimensions: (512,512,355)
Origin: (0,0,0)
Spacing: (1,1,40)
DirectionCosines: (1,0,0,0,1,0)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel    :1
BitsAllocated      :8
BitsStored         :8
HighBit            :7
PixelRepresentation:0
ScalarType found   :UINT8
PhotometricInterpretation: MONOCHROME2
PlanarConfiguration: 0
TransferSyntax: 1.2.840.10008.1.2.4.50
Orientation Label: AXIAL
*/

/*
 * Description:
 *
 * Assume we have a file angiogram-06.dcm as described above.
 * the following program will decompress directly from the extracted jpeg stream.
 *
 * First step extract the jpeg stream (but not the Basic Offset Table):
 *
 * $ gdcmraw -i angiogram-06.dcm -o /tmp/output/chris --split-frags --pattern %d.jpg
 *
 * Check that indeed there are 355 files, while there are 356 fragments in the original DICOM file, since
 * gdcmraw always skip the first fragment (Basic Offset Table).
 *
 * Now from those individual jpeg stream, recreate a fake gdcm.DataElement...
 *
 * Usage:
 *
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono ./bin/DecompressImageMultiframe.exe /tmp/output
 */
using System;
using gdcm;

public class DecompressImageMultiframe
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    gdcm.Directory dir = new gdcm.Directory();
    uint nfiles = dir.Load(directory);
    //System.Console.WriteLine(dir.toString());
    gdcm.FilenamesType filenames = dir.GetFilenames();

    Image image = new Image();
    image.SetNumberOfDimensions( 3 ); // important for now
    DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

    // Create a new SequenceOfFragments C++ object, store it as a SmartPointer :
    SmartPtrFrag sq = SequenceOfFragments.New();

    // Yeah, the files are not guaranteed to be in order, please adapt...
    for(uint i = 0; i < nfiles; ++i)
      {
      System.Console.WriteLine( filenames[(int)i] );
      string file = filenames[(int)i];
      System.IO.FileStream infile =
```

```
        new System.IO.FileStream(file, System.IO.FileMode.Open, System.IO.FileAccess.Read);
      uint fsize = gdcm.PosixEmulation.FileSize(file);

      byte[] jstream  = new byte[fsize];
      infile.Read(jstream, 0 , jstream.Length);

      Fragment frag = new Fragment();
      frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
      sq.AddFragment( frag );
      }

    // Pass by reference:
    pixeldata.SetValue( sq.__ref__() );

    // insert:
    image.SetDataElement( pixeldata );

    // JPEG use YBR to achieve better compression ratio by default (not RGB)
    // FIXME hardcoded:
    PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2
      );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(1,8,8,7);
    image.SetPixelFormat( pixeltype );

    // FIXME hardcoded:
    image.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 ) );
    image.SetDimension(0, 512);
    image.SetDimension(1, 512);
    image.SetDimension(2, 355);

    // Decompress !
    byte[] decompressedData = new byte[(int)image.GetBufferLength()];
    image.GetBuffer(decompressedData);

    // Write out the decompressed bytes
    System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.raw",
        System.IO.FileMode.Create))
      {
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write(decompressedData);
      }


    return 0;
    }
}
```

14.20 DumpCSA.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ bin/DumpCSA.exe input.dcm
 */
using System;
using gdcm;

public class DumpCSA
{
```

```
  public static int Main(string[] args)
    {
    string filename = args[0];

    gdcm.Reader reader = new gdcm.Reader();
    reader.SetFileName( filename );
    if (!reader.Read()) return 1;

    gdcm.File f = reader.GetFile();
    gdcm.DataSet ds = f.GetDataSet();

    string[] expectedSiemensTags = new string[] { "B_value" , "AcquisitionMatrixText" };
    using (PrivateTag gtag = CSAHeader.GetCSAImageHeaderInfoTag())
    {
        if (ds.FindDataElement(gtag))
        {
            using (DataElement de = ds.GetDataElement(gtag))
            {
                if (de != null && !de.IsEmpty())
                {
                    using (CSAHeader csa = new CSAHeader())
                    {
                        if (csa.LoadFromDataElement(de))
                        {
                            foreach (string str in expectedSiemensTags)
                            {
                                if (csa.FindCSAElementByName(str))
                                {
                                    using (CSAElement elem = csa.GetCSAElementByName(str))
                                    {
                                        if (elem != null)
                                        {
                                            System.Console.WriteLine( elem.toString() );
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }


    return 0;
    }
}
```

14.21 ExtractEncapsulatedFile.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This example shows how one from C# context can extract a binary blob
 * and write out as a file.
 * This example is meant for pdf encapsulated file, but can be adapted for other type
 * of binary blob.
 *
 * DICOM file is:
 * ...
 * (0042,0010) ST (no value available)                     #   0, 0 DocumentTitle
 * (0042,0011) OB 25\50\44\46\2d\31\2e\32\20\0d\25\e2\e3\cf\d3\20\0d\31\30\20\30\20... # 40718, 1
      EncapsulatedDocument
```

```
 * (0042,0012) LO [application/pdf]                        #  16, 1 MIMETypeOfEncapsulatedDocument
 * ...
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ExtractEncapsulatedFile.exe some_pdf_encapsulated.dcm
 */
using System;
using gdcm;

public class ExtractEncapsulatedFile
{
  public static int Main(string[] args)
    {
    string file = args[0];
    Reader reader = new Reader();
    reader.SetFileName( file );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();
    Tag tencapsulated_stream = new Tag(0x0042,0x0011); // Encapsulated Document
    if( !ds.FindDataElement( tencapsulated_stream ) )
      {
      return 1;
      }
    // else
    DataElement de = ds.GetDataElement( tencapsulated_stream );
    ByteValue bv = de.GetByteValue();
    uint len = bv.GetLength();
    byte[] encapsulated_stream = new byte[len];
    bv.GetBuffer( encapsulated_stream, len );

    // Write out the decompressed bytes
    //System.Console.WriteLine(image.toString());
    using (System.IO.Stream stream =
      System.IO.File.Open(@"/tmp/dd.pdf",
        System.IO.FileMode.Create))
      {
      System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
      writer.Write( encapsulated_stream );
      }


    return 0;
    }
}
```

14.22 ExtractImageRegion.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 *
 * Usage:
 * $ bin/ExtractImageRegion.exe input.dcm
 *
```

```
 * Example:
 * $ bin/ExtractImageRegion.exe gdcmData/012345.002.050.dcm
 * $ md5sum /tmp/frame.raw
 * d594a5e2fde12f32b6633ca859b4d4a6  /tmp/frame.raw
 * $ gdcminfo --md5sum gdcmData/012345.002.050.dcm
 * [...]
 * md5sum: d594a5e2fde12f32b6633ca859b4d4a6
 */
using System;
using gdcm;

public class ExtractImageRegion
{
  public static int Main(string[] args)
    {
    string filename = args[0];

    uint file_size = gdcm.PosixEmulation.FileSize(filename);

    // instantiate the reader:
    gdcm.ImageRegionReader reader = new gdcm.ImageRegionReader();
    reader.SetFileName( filename );

    // pull DICOM info:
    if (!reader.ReadInformation()) return 1;

    // store current offset:
    uint cur_pos = reader.GetStreamCurrentPosition();

    uint remaining = file_size - cur_pos;

    Console.WriteLine("Remaining bytes to read (Pixel Data): " + remaining.ToString() );

    // Get file infos
    gdcm.File f = reader.GetFile();

    // get some info about image
    UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
    PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
    int pixelsize = pf.GetPixelSize();
    PhotometricInterpretation pi = ImageHelper.GetPhotometricInterpretationValue(f);
    Console.WriteLine( pi.toString() );

    // buffer to get the pixels
    byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];

    // define a simple box region.
    BoxRegion box = new BoxRegion();
    for (uint z = 0; z < dims[2]; z++)
      {
      // Define that I want the image 0, full size (dimx x dimy pixels)
      // and do that for each z:
      box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
      //System.Console.WriteLine( box.toString() );
      reader.SetRegion( box );

      // reader will try to load the uncompressed image region into buffer.
      // the call returns an error when buffer.Length is too small. For instance
      // one can call:
      // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
      // to get the exact size of minimum buffer
      if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
        {
        using (System.IO.Stream stream =
          System.IO.File.Open(@"/tmp/frame.raw",
            System.IO.FileMode.Create))
          {
          System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
          writer.Write(buffer);
          }
        }
      else
        {
        throw new Exception("can’t read pixels error");
        }
      }

    return 0;
    }
}
```

14.23 ExtractImageRegionWithLUT.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 * Furthermore we are applying the LUT on this image.
 * Special care should be taken in case the image is not PALETTE COLOR
 *
 * Usage:
 * $ bin/ExtractImageRegionWithLUT.exe input.dcm
 *
 * Example:
 * $ bin/ExtractImageRegionWithLUT.exe gdcmData/rle16loo.dcm
 * $ md5sum /tmp/frame_rgb.raw
 * 73bf61325fdb6e2830244a2b7b0c4ae2  /tmp/frame_rgb.raw
 * $ gdcmimg --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
 * $ gdcmviewer rgb.dcm
 */
using System;
using gdcm;

public class ExtractImageRegion
{
  public static int Main(string[] args)
    {
    string filename = args[0];

    // instantiate the reader:
    gdcm.ImageRegionReader reader = new gdcm.ImageRegionReader();
    reader.SetFileName( filename );

    // pull DICOM info:
    if (!reader.ReadInformation()) return 1;
    // Get file infos
    gdcm.File f = reader.GetFile();

    gdcm.LookupTable lut = reader.GetImage().GetLUT();

    // get some info about image
    UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
    PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
    int pixelsize = pf.GetPixelSize();

    // buffer to get the pixels
    byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];

    // output buffer for the RGB decoded image:
    byte[] buffer2 = new byte[ dims[0] * dims[1] * pixelsize * 3 ];

    // define a simple box region.
    BoxRegion box = new BoxRegion();
    for (uint z = 0; z < dims[2]; z++)
      {
      // Define that I want the image 0, full size (dimx x dimy pixels)
      // and do that for each z:
      box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
      //System.Console.WriteLine( box.toString() );
      reader.SetRegion( box );

      // reader will try to load the uncompressed image region into buffer.
      // the call returns an error when buffer.Length is too small. For instance
      // one can call:
      // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
      // to get the exact size of minimum buffer
      if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
```

```
        {
        if( !lut.Decode( buffer2, (uint)buffer2.Length, buffer, (uint)buffer.Length ) )
          {
          throw new Exception("can't decode");
          }

        using (System.IO.Stream stream =
          System.IO.File.Open(@"/tmp/frame_rgb.raw",
            System.IO.FileMode.Create))
          {
          System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
          writer.Write(buffer2);
          }
        }
      else
        {
        throw new Exception("can't read pixels error");
        }
      }

    return 0;
    }
}
```

14.24 ExtractOneFrame.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This small code shows how to use the gdcm.StreamImageReader API
 * to read a single (whole) frame at a time
 * The API allow extracting a smaller extent of the frame of course.
 * It will write out the extracted frame in /tmp/frame.raw
 *
 * Usage:
 * $ bin/ExtractOneFrame.exe input.dcm
 */
using System;
using gdcm;

public class ExtractOneFrame
{
  public static int Main(string[] args)
    {
    string filename = args[0];

    gdcm.StreamImageReader reader = new gdcm.StreamImageReader();

    reader.SetFileName( filename );

    if (!reader.ReadImageInformation()) return 1;
    // Get file infos
    gdcm.File f = reader.GetFile();

    // get some info about image
    UIntArrayType extent = ImageHelper.GetDimensionsValue(f);
    //System.Console.WriteLine( extent[0] );
    uint dimx = extent[0];
    //System.Console.WriteLine( extent[1] );
    uint dimy = extent[1];
    //System.Console.WriteLine( extent[2] );
    uint dimz = extent[2];
    PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
    int pixelsize = pf.GetPixelSize();
```

```
    //System.Console.WriteLine( pixelsize );

    // buffer to get the pixels
    byte[] buffer = new byte[ dimx * dimy * pixelsize ];

    for (int i = 0; i < dimz; i++)
      {
      // Define that I want the image 0, full size (dimx x dimy pixels)
      reader.DefinePixelExtent(0, (ushort)dimx, 0, (ushort)dimy, (ushort)i, (ushort)(i+1));
      uint buf_len = reader.DefineProperBufferLength(); // take into account pixel size
      //System.Console.WriteLine( buf_len );
      if( buf_len > buffer.Length )
        {
        throw new Exception("buffer is too small for target");
        }

      if (reader.Read(buffer, (uint)buffer.Length))
        {
        using (System.IO.Stream stream =
          System.IO.File.Open(@"/tmp/frame.raw",
            System.IO.FileMode.Create))
          {
          System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
          writer.Write(buffer);
          }
        }
      else
        {
        throw new Exception("can't read pixels error");
        }
      }

    return 0;
    }
}
```

14.25 FileAnonymize.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example
 *
 * Usage:
 * $ mono bin/FileAnonymize.exe input.dcm output.dcm
 */
using System;
using gdcm;

public class FileAnonymize
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];

    gdcm.FileAnonymizer fa = new gdcm.FileAnonymizer();
    fa.SetInputFileName( filename );
    fa.SetOutputFileName( outfilename );

    // Empty Operations
    // It will create elements, since those tags are non-registered public elements (2011):
    fa.Empty( new Tag(0x0008,0x1313) );
    fa.Empty( new Tag(0x0008,0x1317) );
```

```
    // Remove Operations
    // The following Tag are actually carefully chosen, since they refer to SQ:
    fa.Remove( new Tag(0x0008,0x2112) );
    fa.Remove( new Tag(0x0008,0x9215) );
    // Replace Operations
    // do not call replace operation on SQ attribute !
    fa.Replace( new Tag(0x0018,0x5100), "MYVALUE " );
    fa.Replace( new Tag(0x0008,0x1160), "MYOTHERVAL" );

    if( !fa.Write() )
      {
      System.Console.WriteLine( "Could not write" );
      return 1;
      }

    return 0;
    }
}
```

14.26 FileChangeTS.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example
 *
 * Shows multiple steps:
 * Steps 1.
 * Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
 * image to be able to create the volume in memory Of course you can use any
 * existing DICOM instead
 *
 * Step 2.
 * Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
 * At this point in time this makes the DICOM file invalid (truncated). But the
 * next step will fix this.
 *
 * Step 3.
 * Use C# to create a binary data which will represent our source object for
 * image.
 *
 * Step 4.
 * We use gdcm.FileStreamer to merge the template DICOM file from Step 2, with
 * the binary data from Step 3. We decide to read a scanline at a time, but
 * this can be read with any number of bytes. AppendToDataElement() will always
 * do the proper computation.
 *
 * Step 5.
 * We compress this gigantic file, into [JPEG Lossless, Non-Hierarchical,
 * First-Order Prediction (Process 14 [Selection Value 1])]
 *
 * Usage:
 * $ mono bin/FileChangeTS.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
 */
using System;
using System.IO;
using gdcm;

public class FileChangeTS
{
  public static byte[] StrToByteArray(string str)
    {
    System.Text.ASCIIEncoding  encoding=new System.Text.ASCIIEncoding();
    return encoding.GetBytes(str);
```

```
  }
// Create a 256 x 256 Secondary Capture Image Storage
static private void CreateSmallDICOM(string fileName)
  {
  using( var writer = new gdcm.PixmapWriter() )
    {
    gdcm.Pixmap img = writer.GetImage();
    img.SetNumberOfDimensions( 3 );
    img.SetDimension(0, 512 );
    img.SetDimension(1, 512 );
    img.SetDimension(2, 2 ); // fake a 3d volume
    PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2
    );
    img.SetPhotometricInterpretation( pi );
    gdcm.DataElement pixeldata = new gdcm.DataElement( new gdcm.Tag(0x7fe0,0x0010) );
    byte[] buffer = new byte[ 512 * 512 * 2 ];
    pixeldata.SetByteValue( buffer, new gdcm.VL((uint)buffer.Length) );
    img.SetDataElement( pixeldata );

    gdcm.File file = writer.GetFile();
    gdcm.DataSet ds = file.GetDataSet();
    gdcm.DataElement ms = new gdcm.DataElement(new gdcm.Tag(0x0008,0x0016));
    string mediastorage = "1.2.840.10008.5.1.4.1.1.7.2"; // Multi-frame Grayscale Byte Secondary Capture Image
    Storage
    byte[] val = StrToByteArray(mediastorage);
    ms.SetByteValue( val, new gdcm.VL( (uint)val.Length) );
    ds.Insert( ms );

    writer.SetFileName( fileName );
    writer.Write();
    }
  }
static private void CreateBigDICOM(string fileName, string outfilename)
  {
  using( var ano = new gdcm.FileAnonymizer() )
    {
    // The following is somewhat dangerous, do not try at home:
    string nframes = "1000";
    ano.Replace( new gdcm.Tag(0x0028,0x0008), nframes );
    ano.SetInputFileName(fileName);
    ano.SetOutputFileName(outfilename);
    ano.Write(); // at this point the DICOM is invalid !
    }
  }
static private void CreateDummyFile(string fileName, long length)
  {
  using (var fileStream = new FileStream(fileName, FileMode.Create, FileAccess.Write, FileShare.None))
    {
    // Looks like C# always init to 0 (fallocate ?)
    // For the purpose of the test we could add some random noise
    fileStream.SetLength(length);
    }
  }
static private void ReadBytesIntoArray( byte[] array, FileStream source )
  {
  int numBytesToRead = array.Length;
  int numBytesRead = 0;
  while (numBytesToRead > 0)
    {
    // According to spec: Read() may return anything from 0 to numBytesToRead.
    int n = source.Read(array, numBytesRead, numBytesToRead);

    // Break when the end of the file is reached.
    if (n == 0)
      break;

    numBytesRead += n;
    numBytesToRead -= n;
    }
  }
static private void AssembleDICOMAndRaw(string dicomfn, string rawdata, string outfn)
  {
  using ( var fs = new gdcm.FileStreamer() )
    {
    fs.SetTemplateFileName(dicomfn);
    fs.SetOutputFileName(outfn);
    gdcm.Tag pixeldata = new gdcm.Tag(0x7fe0, 0x0010);
    // FileStreamer support automatic checking of pixel data length
    // based on DICOM attributes, only if we say so:
    fs.CheckDataElement( pixeldata );
    // Declare we are working on Pixel Data attribute:
```

```
      fs.StartDataElement( pixeldata );
      using (FileStream rawSource = new FileStream(rawdata,
          FileMode.Open, FileAccess.Read))
        {
        byte[] bytes = new byte[512];
        // Only read one scanline at a time
        // We could have been reading more at once, if this is more efficient,
        // AppendToDataElement will do the logic in all cases.
        for( int i = 0; i < 512 * 1000; ++i )
          {
          // Read the source file into a byte array.
          ReadBytesIntoArray( bytes, rawSource );
          fs.AppendToDataElement( pixeldata, bytes, (uint)bytes.Length );
          }
        }
      if( !fs.StopDataElement( pixeldata ) )
        {
        // Most likely an issue with Pixel Data Length computation:
        throw new Exception("StopDataElement failed");
        }
      }
    }
  static private void CompressIntoJPEG(string rawdicom, string jpegdicom)
    {
    using( var sfcts = FileChangeTransferSyntax.New() )
      {
      // Need to retrieve the actual C++ reference, to pass to
      // SimpleSubjectWatcher:
      FileChangeTransferSyntax fcts = sfcts.__ref__();
      SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(fcts, "FileChangeTransferSyntax");
      gdcm.TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.JPEGLosslessProcess14_1 );
      fcts.SetTransferSyntax( ts );
      fcts.SetInputFileName( rawdicom );
      fcts.SetOutputFileName( jpegdicom );
      fcts.Change();
      }
    }
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    string rawfilename = args[2];
    string mergefn = args[3];
    string jpegfn = args[4];

    CreateSmallDICOM(filename);
    CreateBigDICOM(filename, outfilename);
    CreateDummyFile(rawfilename, 512 * 512 * 1000 );
    AssembleDICOMAndRaw(outfilename, rawfilename, mergefn);
    CompressIntoJPEG(mergefn, jpegfn);

    return 0;
    }
}
```

14.27 FileChangeTSLossy.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example
 *
 * Shows multiple steps:
 * Steps 1.
```

```
 * Create a fake (dummy) DICOM file, with size 512 x 512 x 2 We use a small
 * image to be able to create the volume in memory Of course you can use any
 * existing DICOM instead
 *
 * Step 2.
 * Hack the DICOM file to pretend the number of frames is 1000 (instead of 2)
 * At this point in time this makes the DICOM file invalid (truncated). But the
 * next step will fix this.
 *
 * Step 3.
 * Use C# to create a binary data which will represent our source object for
 * image.
 *
 * Step 4.
 * We use gdcm.FileStreamer to merge the template DICOM file from Step 2, with
 * the binary data from Step 3. We decide to read a scanline at a time, but
 * this can be read with any number of bytes. AppendToDataElement() will always
 * do the proper computation.
 *
 * Step 5.
 * We compress this gigantic file, into [JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit
     Image Compression]
 *
 * Usage:
 * $ bin/FileChangeTSLossy.exe small.dcm big.dcm raw.data merge.dcm jpeg.dcm
 */
using System;
using System.IO;
using gdcm;

public class FileChangeTS
{
  public static byte[] StrToByteArray(string str)
    {
    System.Text.ASCIIEncoding  encoding=new System.Text.ASCIIEncoding();
    return encoding.GetBytes(str);
    }
  // Create a 256 x 256 Secondary Capture Image Storage
  static private void CreateSmallDICOM(string fileName)
    {
    using( var writer = new gdcm.PixmapWriter() )
      {
      gdcm.Pixmap img = writer.GetImage();
      img.SetNumberOfDimensions( 3 );
      img.SetDimension(0, 512 );
      img.SetDimension(1, 512 );
      img.SetDimension(2, 2 ); // fake a 3d volume
      PhotometricInterpretation pi = new PhotometricInterpretation( PhotometricInterpretation.PIType.MONOCHROME2
      );
      img.SetPhotometricInterpretation( pi );
      gdcm.DataElement pixeldata = new gdcm.DataElement( new gdcm.Tag(0x7fe0,0x0010) );
      byte[] buffer = new byte[ 512 * 512 * 2 ];
      pixeldata.SetByteValue( buffer, new gdcm.VL((uint)buffer.Length) );
      img.SetDataElement( pixeldata );

      gdcm.File file = writer.GetFile();
      gdcm.DataSet ds = file.GetDataSet();
      gdcm.DataElement ms = new gdcm.DataElement(new gdcm.Tag(0x0008,0x0016));
      string mediastorage = "1.2.840.10008.5.1.4.1.1.7.2"; // Multi-frame Grayscale Byte Secondary Capture Image
      Storage
      byte[] val = StrToByteArray(mediastorage);
      ms.SetByteValue( val, new gdcm.VL( (uint)val.Length) );
      ds.Insert( ms );

      writer.SetFileName( fileName );
      writer.Write();
      }
    }
  static private void CreateBigDICOM(string fileName, string outfilename)
    {
    using( var ano = new gdcm.FileAnonymizer() )
      {
      // The following is somewhat dangerous, do not try at home:
      string nframes = "1000";
      ano.Replace( new gdcm.Tag(0x0028,0x0008), nframes );
      ano.SetInputFileName(fileName);
      ano.SetOutputFileName(outfilename);
      ano.Write(); // at this point the DICOM is invalid !
      }
    }
  static private void CreateDummyFile(string fileName, long length)
```

```
  {
  using (var fileStream = new FileStream(fileName, FileMode.Create, FileAccess.Write, FileShare.None))
    {
    // Looks like C# always init to 0 (fallocate ?)
    // For the purpose of the test we could add some random noise
    fileStream.SetLength(length);
    }
  }
static private void ReadBytesIntoArray( byte[] array, FileStream source )
  {
  int numBytesToRead = array.Length;
  int numBytesRead = 0;
  while (numBytesToRead > 0)
    {
    // According to spec: Read() may return anything from 0 to numBytesToRead.
    int n = source.Read(array, numBytesRead, numBytesToRead);

    // Break when the end of the file is reached.
    if (n == 0)
      break;

    numBytesRead += n;
    numBytesToRead -= n;
    }
  }
static private void AssembleDICOMAndRaw(string dicomfn, string rawdata, string outfn)
  {
  using ( var fs = new gdcm.FileStreamer() )
    {
    fs.SetTemplateFileName(dicomfn);
    fs.SetOutputFileName(outfn);
    gdcm.Tag pixeldata = new gdcm.Tag(0x7fe0, 0x0010);
    // FileStreamer support automatic checking of pixel data length
    // based on DICOM attributes, only if we say so:
    fs.CheckDataElement( pixeldata );
    // Declare we are working on Pixel Data attribute:
    fs.StartDataElement( pixeldata );
    using (FileStream rawSource = new FileStream(rawdata,
        FileMode.Open, FileAccess.Read))
      {
      byte[] bytes = new byte[512];
      // Only read one scanline at a time
      // We could have been reading more at once, if this is more efficient,
      // AppendToDataElement will do the logic in all cases.
      for( int i = 0; i < 512 * 1000; ++i )
        {
        // Read the source file into a byte array.
        ReadBytesIntoArray( bytes, rawSource );
        fs.AppendToDataElement( pixeldata, bytes, (uint)bytes.Length );
        }
      }
    if( !fs.StopDataElement( pixeldata ) )
      {
      // Most likely an issue with Pixel Data Length computation:
      throw new Exception("StopDataElement failed");
      }
    }
  }
static private void CompressIntoJPEG(string rawdicom, string jpegdicom)
  {
  using( var sfcts = FileChangeTransferSyntax.New() )
    {
    // Need to retrieve the actual C++ reference, to pass to
    // SimpleSubjectWatcher:
    FileChangeTransferSyntax fcts = sfcts.__ref__();
    SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(fcts, "FileChangeTransferSyntax");
    gdcm.TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.JPEGBaselineProcess1 );
    fcts.SetTransferSyntax( ts );
    ImageCodec ic = fcts.GetCodec();
    JPEGCodec jpeg = JPEGCodec.Cast( ic );
    jpeg.SetLossless( false );
    jpeg.SetQuality( 50 ); // poor quality !

    fcts.SetInputFileName( rawdicom );
    fcts.SetOutputFileName( jpegdicom );
    fcts.Change();
    }
  }
public static int Main(string[] args)
  {
  string filename = args[0];
```

```
    string outfilename = args[1];
    string rawfilename = args[2];
    string mergefn = args[3];
    string jpegfn = args[4];

    CreateSmallDICOM(filename);
    CreateBigDICOM(filename, outfilename);
    CreateDummyFile(rawfilename, 512 * 512 * 1000 );
    AssembleDICOMAndRaw(outfilename, rawfilename, mergefn);
    CompressIntoJPEG(mergefn, jpegfn);

    return 0;
    }
}
```

14.28 FileStreaming.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Simple C# example
 *
 * Usage:
 * $ mono bin/FileStreaming.exe gdcmData/CT_16b_signed-UsedBits13.dcm output.dcm
 *
 * The class will take care of group handling and will use the first available group:
 * (0009,0012) ?? (LO) [MYTEST]                          # 6,1 Private Creator
 */
using System;
using gdcm;

public class FileStreaming
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];

    gdcm.PrivateTag pt = new gdcm.PrivateTag( new gdcm.Tag(0x9,0x10), "MYTEST" );

    gdcm.FileStreamer fs = new gdcm.FileStreamer();
    fs.SetTemplateFileName( filename );
    fs.SetOutputFileName( outfilename );

    byte[] buffer = new byte[ 8192 ];
    uint len = (uint)buffer.Length;

    // In this example, we want that each newly created Private Attribute
    // contains at most 1000 bytes of incoming dataset.
    // We are also calling the function twice to check that appending mode is
    // working from one call to the other. The last element will have a length
    // of (2 * 8192) % 1000 = 384
    if( !fs.StartGroupDataElement( pt, 1000, 1 )
      || !fs.AppendToGroupDataElement( pt, buffer, len )
      || !fs.AppendToGroupDataElement( pt, buffer, len )
      || !fs.StopGroupDataElement( pt ) )
      {
      System.Console.WriteLine( "Could not change private group" );
      return 1;
      }

    return 0;
    }
}
```

14.29 GetArray.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/GetArray.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class GetArray
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    Image image = reader.GetImage();

    PixelFormat pixeltype = image.GetPixelFormat();

    if( image.GetNumberOfDimensions() != 2 )
      {
      // For the purpose of the test, exit early on
      return 1;
      }
    uint dimx = image.GetDimension(0);
    uint dimy = image.GetDimension(1);
    uint npixels = dimx * dimy;
    //LookupTable lut = image.GetLUT();
    //uint rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
    //byte[] rbuf = new byte[ rl ];
    //uint rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
    //assert rl == rl2;

    //byte[] str1 = new byte[ image.GetBufferLength()];
    //image.GetBuffer( str1 );
    if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT8 )
      {
      System.Console.WriteLine( "Processing UINT8 image type" );
      byte[] str1 = new byte[ npixels ];
      image.GetArray( str1 );
      }
    else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.INT16 )
      {
      System.Console.WriteLine( "Processing INT16 image type" );
      short[] str1 = new short[ npixels ];
      image.GetArray( str1 );
      }
    else if( pixeltype.GetScalarType() == PixelFormat.ScalarType.UINT16 )
      {
      System.Console.WriteLine( "Processing UINT16 image type" );
      ushort[] str1 = new ushort[ npixels ];
      image.GetArray( str1 );
      }
    else
      {
      //System.Console.WriteLine( "Default (unhandled pixel format): " + pixeltype.toString() );
      System.Console.WriteLine( "Default (unhandled pixel format): " + pixeltype.GetScalarTypeAsString() );
      // Get bytes
```

```
      byte[] str1 = new byte[ image.GetBufferLength()];
      image.GetBuffer( str1 );
      }

    return 0;
    }
}
```

14.30 MpegVideoInfo.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This examples takes in a MPEG2 and write out a Video Endoscopic Imagae Storage
 * encoded using MPEG2 @ Main Profile
 * ref: http://chrisa.wordpress.com/2007/11/21/decoding-mpeg2-information/
 * See also:
 * http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html#gop
 * http://cvs.linux.hr/cgi-bin/viewcvs.cgi/mpeg_mod/README.infompeg?view=markup
 * http://www.guru-group.fi/~too/sw/m2vmp2cut/mpeg2info.c
 */

/*
 * Provides information about an MPEG2 file, including the duration, frame rate, aspect
 * ratio, and resolution.  Good information about the MPEG2 file structure that helps
 * explain parts of the code can be found here:
 * http://dvd.sourceforge.net/dvdinfo/mpeghdrs.html#gop
 *
 * Copyright (c) 2007 Chris Anderson (chrisa@wordpress.com)
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 */
using System;
using System.IO;
using gdcm;

public class Mpeg2VideoInfo
{
    #region Member Variables
    private TimeSpan        m_startTime     = TimeSpan.Zero;
    private TimeSpan        m_endTime       = TimeSpan.Zero;
    private TimeSpan        m_duration      = TimeSpan.Zero;
    private eAspectRatios   m_aspectRatio   = eAspectRatios.Invalid;
    private eFrameRates     m_frameRate     = 0;
    private int             m_pictureWidth  = 0;
    private int             m_pictureHeight = 0;
    #endregion

    #region Constants
    private const byte PADDING_PACKET = 0xBE;
    private const byte VIDEO_PACKET = 0xE0;
    private const byte AUDIO_PACKET = 0xC0;
    private const byte SYSTEM_PACKET = 0xBB;
    private const byte TIMESTAMP_PACKET = 0xB8;
```

```
private const byte HEADER_PACKET = 0xB3;

private const int BUFFER_SIZE = 8162; // 8K buffer

private readonly static TimeSpan EMPTY_TIMESPAN = new TimeSpan(0, 0, -1);
#endregion

#region Enumerations
public enum eFrameRates
{
    Invalid,
    PulldownNTSC,           // 24000d/1001d = 23.976 Hz
    Film,                   // 24 Hz
    PAL,                    // 25 Hz
    NTSC,                   // 30000d/1001d = 29.97 Hz
    DropFrameNTSC,          // 30 Hz
    DoubleRatePAL,          // 50 Hz
    DoubleRateNTSC,         // 59.97 Hz
    DoubleRateDropFrameNTSC // 60 Hz
}

public enum eAspectRatios
{
    Invalid,
    VGA,        // 1/1
    StandardTV, // 4/3
    LargeTV,    // 16/9
    Cinema      // 2.21/1
}
#endregion

#region Constructor
public Mpeg2VideoInfo(string file)
{
    ParseMpeg(file);
}
#endregion

#region Public Properties
public TimeSpan StartTime
{
    get { return m_startTime; }
}

public TimeSpan EndTime
{
    get { return m_endTime; }
}

public TimeSpan Duration
{
    get { return m_duration; }
}

public eAspectRatios AspectRatio
{
    get { return m_aspectRatio; }
}

public eFrameRates FrameRate
{
    get { return m_frameRate; }
}

public int PictureWidth
{
    get { return m_pictureWidth; }
}

public int PictureHeight
{
    get { return m_pictureHeight; }
}
#endregion

#region Private Functions
private void ParseMpeg(string file)
{
    FileStream fs = new FileStream(file, FileMode.Open, FileAccess.Read, FileShare.ReadWrite);
```

```
        BinaryReader br = new BinaryReader(fs);

        m_startTime = GetStartTimeStampInfo(br);
        m_endTime = GetEndTimeStampInfo(br);

        m_duration = m_endTime.Subtract(m_startTime);

        GetHeaderInfo(br);

        br.Close();
        fs.Close();
    }

    private TimeSpan GetStartTimeStampInfo(BinaryReader br)
    {
        TimeSpan startTime = EMPTY_TIMESPAN;
        byte[] buffer = new byte[BUFFER_SIZE];

        br.BaseStream.Seek(0, SeekOrigin.Begin);

        while (startTime == EMPTY_TIMESPAN && br.BaseStream.Position < br.BaseStream.Length)
        {
            int readBytes = br.Read(buffer, 0, BUFFER_SIZE);

            for (int offset = 0; offset < readBytes - 8; offset++)
            {
                if (IsStreamMarker(ref buffer, offset, TIMESTAMP_PACKET))
                {
                    offset += 4; // Move to the data position which follows the stream header
                    uint timeStampEncoded = GetData(ref buffer, offset);
                    startTime = DecodeTimeStamp(timeStampEncoded);

                    if (startTime != EMPTY_TIMESPAN)
                        break;
                }
            }
        }

        return startTime;
    }

    private TimeSpan GetEndTimeStampInfo(BinaryReader br)
    {
        TimeSpan endTime = EMPTY_TIMESPAN;
        byte[] buffer = new byte[BUFFER_SIZE];

        br.BaseStream.Seek(-BUFFER_SIZE, SeekOrigin.End);

        while (endTime == EMPTY_TIMESPAN && br.BaseStream.Position > BUFFER_SIZE)
        {
            int readBytes = br.Read(buffer, 0, BUFFER_SIZE);

            for (int offset = readBytes - 8; offset >= 0; offset--)
            {
                if (IsStreamMarker(ref buffer, offset, TIMESTAMP_PACKET))
                {
                    offset += 4; // Move to the data position which follows the stream header
                    uint timeStampEncoded = GetData(ref buffer, offset);
                    endTime = DecodeTimeStamp(timeStampEncoded);

                    if (endTime != EMPTY_TIMESPAN)
                        break;
                }
            }

            br.BaseStream.Seek(-BUFFER_SIZE * 2, SeekOrigin.Current);
        }

        return endTime;
    }

    private TimeSpan DecodeTimeStamp(uint timeStampEncoded)
    {
        TimeSpan timeStamp = EMPTY_TIMESPAN;

        // Mask out the bits containing the property we are after, then
        // shift the data to the right to get its value
        int hour = (int)(timeStampEncoded & 0x7C000000) » 26;   // Bits 31 -> 27
        int minute = (int)(timeStampEncoded & 0x03F00000) » 20; // Bits 26 -> 21
        int second = (int)(timeStampEncoded & 0x0007E000) » 13; // Bits 19 -> 14
        int frame = (int)(timeStampEncoded & 0x00001F80) » 7;   // Bits 13 -> 8 - not used, but included for
```

```
    completeness

      timeStamp = new TimeSpan(hour, minute, second);
      return timeStamp;
  }

  private void GetHeaderInfo(BinaryReader br)
  {
      byte[] buffer = new byte[BUFFER_SIZE];

      br.BaseStream.Seek(0, SeekOrigin.Begin);
      br.Read(buffer, 0, BUFFER_SIZE);

      for (int offset = 0; offset < buffer.Length - 4; offset++)
      {
          if (IsStreamMarker(ref buffer, offset, HEADER_PACKET))
          {
              offset += 4; // Move to the data position which follows the stream header
              uint headerData = GetData(ref buffer, offset);

              // Mask out the bits containing the property we are after, then
              // shift the data to the right to get its value
              m_pictureWidth = (int)(headerData & 0xFFF00000) » 20;
              m_pictureHeight = (int)(headerData & 0x000FFF00) » 8;

              uint aspectRatioIndex = (headerData & 0x000000F0) » 4;
              uint fpsIndex = headerData & 0x0000000F;

              m_aspectRatio = (eAspectRatios)fpsIndex;
              m_frameRate = (eFrameRates)fpsIndex;

              break;
          }
      }
  }

  private uint GetData(ref byte[] buffer, int offset)
  {
      return (uint)  ((buffer[offset] « 24) |
                      (buffer[offset + 1] « 16) |
                      (buffer[offset + 2] « 8) |
                      (buffer[offset + 3]));
  }

  private bool IsStreamMarker(ref byte[] buffer, int offset, byte markerType)
  {
      return (buffer[offset] == 0x00 &&
              buffer[offset + 1] == 0x00 &&
              buffer[offset + 2] == 0x01 &&
              buffer[offset + 3] == markerType);
  }
  #endregion
public static int Main(string[] args)
  {
  string file1 = args[0];
  Mpeg2VideoInfo info = new Mpeg2VideoInfo(file1);
  System.Console.WriteLine( info.StartTime );
  System.Console.WriteLine( info.EndTime );
  System.Console.WriteLine( info.Duration );
  System.Console.WriteLine( info.AspectRatio );
  System.Console.WriteLine( info.FrameRate );
  System.Console.WriteLine( info.PictureWidth );
  System.Console.WriteLine( info.PictureHeight );

  ImageReader r = new ImageReader();
  //Image image = new Image();
  Image image = r.GetImage();
  image.SetNumberOfDimensions( 3 );
  DataElement pixeldata = new DataElement( new gdcm.Tag(0x7fe0,0x0010) );

  System.IO.FileStream infile =
    new System.IO.FileStream(file1, System.IO.FileMode.Open, System.IO.FileAccess.Read);
  uint fsize = gdcm.PosixEmulation.FileSize(file1);

  byte[] jstream  = new byte[fsize];
  infile.Read(jstream, 0 , jstream.Length);

  SmartPtrFrag sq = SequenceOfFragments.New();
  Fragment frag = new Fragment();
  frag.SetByteValue( jstream, new gdcm.VL( (uint)jstream.Length) );
  sq.AddFragment( frag );
```

```
    pixeldata.SetValue( sq.__ref__() );

    // insert:
    image.SetDataElement( pixeldata );

    PhotometricInterpretation pi = new PhotometricInterpretation(
      PhotometricInterpretation.PIType.YBR_PARTIAL_420 );
    image.SetPhotometricInterpretation( pi );
    // FIXME hardcoded:
    PixelFormat pixeltype = new PixelFormat(3,8,8,7);
    image.SetPixelFormat( pixeltype );

    // FIXME hardcoded:
    TransferSyntax ts = new TransferSyntax( TransferSyntax.TSType.MPEG2MainProfile);
    image.SetTransferSyntax( ts );

    image.SetDimension(0, (uint)info.PictureWidth);
    image.SetDimension(1, (uint)info.PictureHeight);
    image.SetDimension(2, 721);

    ImageWriter writer = new ImageWriter();
    gdcm.File file = writer.GetFile();
    file.GetHeader().SetDataSetTransferSyntax( ts );
    Anonymizer anon = new Anonymizer();
    anon.SetFile( file );

    MediaStorage ms = new MediaStorage( MediaStorage.MSType.VideoEndoscopicImageStorage);

    UIDGenerator gen = new UIDGenerator();
    anon.Replace( new Tag(0x0008,0x16), ms.GetString() );
    anon.Replace( new Tag(0x0018,0x40), "25" );
    anon.Replace( new Tag(0x0018,0x1063), "40.000000" );
    anon.Replace( new Tag(0x0028,0x34), "4\\3" );
    anon.Replace( new Tag(0x0028,0x2110), "01" );

    writer.SetImage( image );
    writer.SetFileName( "dummy.dcm" );
    if( !writer.Write() )
      {
      System.Console.WriteLine( "Could not write" );
      return 1;
      }

    return 0;
    }
}
```

14.31 NewSequence.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/NewSequence.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
//using gdcm;

public class NewSequence
{
  public static byte[] StrToByteArray(string str)
    {
    System.Text.ASCIIEncoding  encoding=new System.Text.ASCIIEncoding();
```

```
    return encoding.GetBytes(str);
    }

  public static int Main(string[] argv)
    {
    string file1 = argv[0];
    string file2 = argv[1];

    gdcm.Reader r = new gdcm.Reader();
    r.SetFileName( file1 );
    if ( ! r.Read() )
      {
      return 1;
      }

    gdcm.File f = r.GetFile();
    gdcm.DataSet ds = f.GetDataSet();
    // tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence

    // Create a dataelement
    gdcm.DataElement de = new gdcm.DataElement(new gdcm.Tag(0x0010, 0x2180));
    string occ = "Occupation";
    de.SetByteValue( StrToByteArray(occ), new gdcm.VL((uint)occ.Length));
    de.SetVR(new gdcm.VR(gdcm.VR.VRType.SH));

    // Create an item
    gdcm.Item it = new gdcm.Item();
    it.SetVLToUndefined();      // Needed to not popup error message
    //it.InsertDataElement(de)
    gdcm.DataSet nds = it.GetNestedDataSet();
    nds.Insert(de);

    // Create a Sequence
    gdcm.SmartPtrSQ sq = gdcm.SequenceOfItems.New();
    sq.SetLengthToUndefined();
    sq.AddItem(it);

    // Insert sequence into data set
    gdcm.DataElement des = new gdcm.DataElement(new gdcm.Tag(0x0400,0x0550));
    des.SetVR(new gdcm.VR(gdcm.VR.VRType.SQ));
    des.SetValue(sq.__ref__());
    des.SetVLToUndefined();

    ds.Insert(des);

    gdcm.Writer w = new gdcm.Writer();
    w.SetFile( f );
    w.SetFileName( file2 );
    if ( !w.Write() )
      return 1;

    return 0;
    }
}
```

14.32 RescaleImage.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcmData/012345.002.050.dcm rescaled.dcm
 */
```

```
using System;
using gdcm;

public class DecompressImage
{
  public static int Main(string[] args)
    {
    string file1 = args[0];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    bool ret = reader.Read();
    if( !ret )
      {
      return 1;
      }

    Image image = reader.GetImage();
    PixelFormat pixeltype = image.GetPixelFormat();

    Rescaler r = new Rescaler();
    r.SetIntercept( 0 );
    r.SetSlope( 1.2 );
    r.SetPixelFormat( pixeltype );
    PixelFormat outputpt = new PixelFormat( r.ComputeInterceptSlopePixelType() );

    System.Console.WriteLine( "pixeltype" );
    System.Console.WriteLine( pixeltype.toString() );
    System.Console.WriteLine( "outputpt" );
    System.Console.WriteLine( outputpt.toString() );

    uint len = image.GetBufferLength();
    short[] input = new short[ len / 2 ]; // sizeof(short) == 2
    image.GetArray( input );

    double[] output = new double[ len / 2 ];
    r.Rescale( output, input, len );

    // First Pixel is:
    System.Console.WriteLine( "Input:" );
    System.Console.WriteLine( input[0] );

    System.Console.WriteLine( "Output:" );
    System.Console.WriteLine( output[0] );

    return 0;
    }
}
```

14.33 SendFileSCU.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm-gcc/bin
 * $ mono bin/SendFileSCU.exe server port input.dcm
 */
using System;
using gdcm;

public class SendFileSCU
{
  public static int Main(string[] args)
    {
```

```
    string server = args[0];
    ushort port = ushort.Parse(args[1]);
    string filename = args[2];

    bool b = CompositeNetworkFunctions.CEcho( server, port );
    if( !b ) return 1;

    FilenamesType files = new FilenamesType();
    files.Add( filename );
    b = CompositeNetworkFunctions.CStore( server, port, files );
    if( !b ) return 1;

    return 0;
    }
}
```

14.34 SimplePrintPatientName.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrintPatientName.exe gdcmData/012345.002.050.dcm
 */
/*
 This example was provided by Jonathan Morra /jonmorra gmail com/
 on the gdcm mailing list (Fri, 28 May 2010)
*/
using System;
using gdcm;

namespace GDCMTest
{
  class SimplePrintPatientName
    {
    static int Main(string[] args)
      {
      if (args.Length != 1)
        {
        Console.WriteLine("This program prints the patient name of a dicom file with gdcm");
        Console.WriteLine("Usage: [input.dcm]");
        return 1;
        }

      gdcm.Reader reader = new gdcm.Reader();
      reader.SetFileName(args[0]);
      bool ret = reader.Read();
      //TagSetType tst = new TagSetType();
      //tst.Add( new Tag(0x7fe0,0x10) );
      //bool ret = reader.ReadUpToTag( new Tag(0x88,0x200), tst );
      if( !ret )
        {
        return 1;
        }

      gdcm.File file = reader.GetFile();

      gdcm.StringFilter filter = new gdcm.StringFilter();
      filter.SetFile(file);
      string value = filter.ToString(new gdcm.Tag(0x0010, 0x0010));

      Console.WriteLine("Patient Name: " + value);
      return 0;
      }
```

```
    }
}
```

14.35 SortImage2.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SortImage.exe gdcmData/012345.002.050.dcm out.dcm
 */
using System;
using gdcm;

public class SortImage2
{
  bool mysort(DataSet ds1, DataSet ds2)
    {
    return false;
    }

  public static int Main(string[] args)
    {
    Sorter sorter = new Sorter();
    sorter.SetSortFunction( mysort );

    return 0;
    }
}
```

14.36 CStoreQtProgress.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This small example show how one can use the virtual function
 * mechanism of the SimpleSubjectWatcher class to redirect progress
 * report to a custom Qt classes
 *
 * http://doc.qt.nokia.com/latest/qprogressdialog.html
 *
 * Usage:
 * CStoreQtProgress dicom.example.com 11112 gdcmData/MR_Spectroscopy_SIEMENS_OF.dcm
 *
 */

#include "gdcmServiceClassUser.h"
#include "gdcmSimpleSubjectWatcher.h"
```

```
#include "gdcmProgressEvent.h"
#include "gdcmDirectory.h"
#include "gdcmPresentationContextGenerator.h"

#include <QApplication>
#include <QProgressDialog>
#include <QVBoxLayout>

namespace gdcm {
/*
 * This class is a little more complicated than what this example demonstrate
 * This watcher is capable of handling nested progress. Since the Progress
 * grows from [0 to 1] on a per file basis and we only have one instance of a
 * watcher per association, we need some calculation to compute the global
 * (total) progress
 * In fact we simply divide the per-file progress by the number of files.
 *
 * This QtWatcher class will then update the progress bar according to the
 * progress.
 */
class MyQtWatcher : public SimpleSubjectWatcher
{
  size_t nfiles;
  double progress;
  size_t index;
  double refprogress;
  QWidget* win;
  QProgressDialog* qtprogress;
public:
  MyQtWatcher(Subject * s, const char *comment = "", QWidget *w = NULL, QProgressDialog* p = NULL, size_t n =
      1):
    SimpleSubjectWatcher(s,comment),nfiles(n),progress(0),index(0),refprogress(0),win(w),qtprogress(p){}
  void ShowIteration()
    {
    index++;
    assert( index <= nfiles );
    // update refprogess (we are moving to the next file)
    refprogress = progress;
    }
  void ShowProgress(Subject *, const Event &evt)
    {
    // Retrieve the ProgressEvent:
    const ProgressEvent &pe = dynamic_cast<const ProgressEvent&>(evt);
    // compute global progress:
    progress = refprogress + (1. / (double)nfiles ) * pe.GetProgress();
    // Print Global and local progress to stdout:
    std::cout « "Global Progress: " « progress « " per file progress " « pe.GetProgress() « std::endl;
    //set progress value in the QtProgress bar
    int i = (int)(progress * 100 + 0.5); // round to next int
    qtprogress->setValue(i);
    win->show();
    }
  virtual void ShowDataSet(Subject *caller, const Event &evt)
    {
    (void)caller;
    (void)evt;
    }
};
} // end namespace gdcm

int main(int argc, char *argv[])
{
  if( argc < 4 )
    {
    std::cerr « argv[0] « " remote_server port filename" « std::endl;
    return 1;
    }
  QApplication a(argc, argv);

  std::ostringstream error_log;
  gdcm::Trace::SetErrorStream( error_log );

  const char *remote = argv[1];
  int portno = atoi(argv[2]);
  const char *filename = argv[3];

  QVBoxLayout* layout = new QVBoxLayout;
  QWidget*  win = new QWidget;

  QProgressDialog* progress = new QProgressDialog("Sending data...", "Cancel", 0, 100);
```

```
  progress->setWindowModality(Qt::WindowModal);

  layout->addWidget(progress,Qt::AlignCenter);
  win->setLayout(layout);

  gdcm::SmartPointer<gdcm::ServiceClassUser> scup = new gdcm::ServiceClassUser;
  gdcm::ServiceClassUser &scu = *scup;
  //gdcm::SimpleSubjectWatcher w( &scu, "TestServiceClassUser" );
  // let's use a more complicated progress reported in this example
  gdcm::MyQtWatcher w( &scu, "QtWatcher", win, progress );

  scu.SetHostname( remote );
  scu.SetPort( (uint16_t)portno );
  scu.SetTimeout( 1000 );
  scu.SetCalledAETitle( "GDCM_STORE" );

  if( !scu.InitializeConnection() )
    {
    std::cerr « "Could not InitializeConnection" « std::endl;
    return 1;
    }

  gdcm::Directory::FilenamesType filenames;
  filenames.push_back( filename );

  // setup the PC(s) based on the filenames:
  gdcm::PresentationContextGenerator generator;
  if( !generator.GenerateFromFilenames(filenames) )
    {
    std::cerr « "Could not GenerateFromFilenames" « std::endl;
    return 1;
    }

  // Setup PresentationContext(s)
  scu.SetPresentationContexts( generator.GetPresentationContexts() );

  // Start ASSOCIATION
  if( !scu.StartAssociation() )
    {
    std::cerr « "Could not Start" « std::endl;
    return 1;
    }

  // Send C-STORE
  if( !scu.SendStore( filename ) )
    {
    std::cerr « "Could not Store" « std::endl;
    std::cerr « "Error log is:" « std::endl;
    std::cerr « error_log.str() « std::endl;
    return 1;
    }

  // Stop ASSOCIATION
  if( !scu.StopAssociation() )
    {
    std::cerr « "Could not Stop" « std::endl;
    return 1;
    }

  win->show();

  return a.exec();
}
```

14.37 ChangePrivateTags.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmPrivateTag.h"

int main(int argc, char* argv[] )
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " path/to/05148044-mr-siemens-avanto-syngo.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if (! reader.Read() )
    {
    return 1;
    }

  // (0029,0010) LO [SIEMENS CSA HEADER]                              # 18,1 Private Creator
  // (0029,0011) LO [SIEMENS MEDCOM HEADER ]                          # 22,1 Private Creator
  // (0029,0012) LO [SIEMENS MEDCOM HEADER2]                          # 22,1 Private Creator
  // [...]
  // (0029,1018) CS [MR]                                              # 2,1 CSA Series Header Type
  // (0029,1134) CS [DB TO DICOM ]                                    # 12,1 PMTF Information 4
  // (0029,1260) LO [com ]                                            # 4,1 Series Workflow Status

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  // Declare private tag we need to find:
  gdcm::PrivateTag pt1( 0x29,0x18, "SIEMENS CSA HEADER" );
  gdcm::PrivateTag pt2( 0x29,0x34, "SIEMENS MEDCOM HEADER" );
  gdcm::PrivateTag pt3( 0x29,0x60, "SIEMENS MEDCOM HEADER2" );

  const char str1[] = "GDCM was here 3!";
  if( !ds.FindDataElement( pt1 ) ) return 1;
  gdcm::DataElement de1 = ds.GetDataElement( pt1 ); // Convert Private tag, into actual DataElement
  std::cout « de1 « std::endl;
  de1.SetByteValue( str1, (uint32_t)strlen(str1) );
  ds.Replace( de1 );

  const char str2[] = "GDCM was here 2!";
  if( !ds.FindDataElement( pt2 ) ) return 1;
  gdcm::DataElement de2 = ds.GetDataElement( pt2 );
  std::cout « de2 « std::endl;
  de2.SetByteValue( str2, (uint32_t)strlen(str2) );
  ds.Replace( de2 );

  const char str3[] = "GDCM was here 3!";
  if( !ds.FindDataElement( pt3 ) ) return 1;
  gdcm::DataElement de3 = ds.GetDataElement( pt3 );
  std::cout « de3 « std::endl;
  de3.SetByteValue( str3, (uint32_t)strlen(str3) );
  ds.Replace( de3 );

  gdcm::Writer writer;
  writer.SetFile( file );
  writer.SetFileName( outfilename );
  if ( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.38 ChangeSequenceUltrasound.cxx

```
/*=========================================================================
```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmSmartPointer.h"
#include "gdcmDataSetHelper.h"

/*
 ./ChangeSequenceUltrasound gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

 This is the exact C++ translation of the original python example: ManipulateSequence.py
 */

int main(int argc, char* argv[] )
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if (! reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  gdcm::Tag tsis(0x0008,0x2112); // SourceImageSequence
  if ( ds.FindDataElement( tsis ) )
    {
    const gdcm::DataElement &sis = ds.GetDataElement( tsis );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqsis = sis.GetValueAsSQ();
    if ( sqsis && sqsis->GetNumberOfItems() )
      {
      gdcm::Item &item1 = sqsis->GetItem(1);
      gdcm::DataSet &nestedds = item1.GetNestedDataSet();
      gdcm::Tag tprcs(0x0040,0xa170); // PurposeOfReferenceCodeSequence
      if( nestedds.FindDataElement( tprcs ) )
        {
        const gdcm::DataElement &prcs = nestedds.GetDataElement( tprcs );
        gdcm::SmartPointer<gdcm::SequenceOfItems> sqprcs = prcs.GetValueAsSQ();
        if ( sqprcs && sqprcs->GetNumberOfItems() )
          {
          gdcm::Item &item2 = sqprcs->GetItem(1);
          gdcm::DataSet &nestedds2 = item2.GetNestedDataSet();
          // (0008,0104) LO [Uncompressed predecessor]               #  24, 1 CodeMeaning
          gdcm::Tag tcm(0x0008,0x0104);
          if( nestedds2.FindDataElement( tcm ) )
            {
            gdcm::DataElement cm = nestedds2.GetDataElement( tcm );
            std::string mystr = "GDCM was here";
            cm.SetByteValue( mystr.c_str(), (uint32_t)mystr.size() );
            nestedds2.Replace( cm );
            }
          }
        }
      }
    }

  gdcm::Writer writer;
  writer.SetFile( file );
  writer.SetFileName( outfilename );
  if ( !writer.Write() )
    {
    return 1;
```

```
    }

  return 0;
}
```

14.39 CheckBigEndianBug.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * WARNING: This is a dev tool, do not use !
 *
 * Usage: after a gdcmconv, you would like to know if the conversion process is acceptable
 * sometime a vbindiff is acceptable, sometime it is not. In the case of the famous Philips
 * Little/Big Endian Explicit Transfer Syntax it is not easy to compare two files. However
 * this only impact byte ordering, thus we can compute byte-indenpendant information to still
 * compare the files.
 */

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmSystem.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input1.dcm input2.dcm" « std::endl;
    return 1;
    }
  const char *filename1 = argv[1];
  const char *filename2 = argv[2];

  gdcm::ImageReader reader1;
  reader1.SetFileName( filename1 );
  if( !reader1.Read() )
    {
    std::cerr « "Could not read: " « filename1 « std::endl;
    return 1;
    }

  gdcm::ImageReader reader2;
  reader2.SetFileName( filename2 );
  if( !reader2.Read() )
    {
    std::cerr « "Could not read: " « filename2 « std::endl;
    return 1;
    }

  // TODO: need a DataSet== operator implementation

  std::cout « "Both files can be read and looks like DICOM" « std::endl;

  size_t s1 = gdcm::System::FileSize(filename1);
  size_t s2 = gdcm::System::FileSize(filename2);

  if( s1 != s2 )
    {
    std::cout « "Size mismatch: " « s1 « " != " « s2 « std::endl;
    return 1;
```

```
    }
  else
    {
    std::cout « "Size match: " « s1 « " = " « s2 « std::endl;
    }

  std::ifstream is1( filename1, std::ios::binary );
  char *buffer1 = new char[s1];
  is1.read(buffer1, s1);

  std::ifstream is2( filename2, std::ios::binary );
  char *buffer2 = new char[s2];
  is2.read(buffer2, s2);

  assert( s1 == s2 );
  if( memcmp(buffer1, buffer2, s1 ) == 0 )
    {
    std::cout « "memcmp succeed ! File are bit identical" « std::endl;
    }
  else
    {
    std::cout « "memcmp failed!" « std::endl;
    }

  // Hum...memcmp failed, for big endian/ little endian inversion the histogram of bytes
  // should still be the same. So let's compute it
  // buffer2[0] = 1; // let's make the test fail
  std::multiset<char> set1( buffer1, buffer1 + s1 );
  std::multiset<char> set2( buffer2, buffer2 + s2 );


  if( set1 == set2 )
    {
    std::cout « "set1 == set2. Byte histogram seems valid" « std::endl;
    }
  else
    {
    std::cout « "set1 != set2" « std::endl;
    }
  delete[] buffer1;
  delete[] buffer2;


  return 0;
}
```

14.40 ClinicalTrialAnnotate.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Dummy implementation of C.7.1.3 Clinical Trial Subject Module
 *
 * Usage:
 *  ClinicalTrialAnnotate gdcmData/012345.002.050.dcm out.dcm
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAnonymizer.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
```

```
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « filename « std::endl;
    return 1;
    }

  // The output of gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();

  // the dataset is the the set of element we are interested in:
  //gdcm::DataSet &ds = file.GetDataSet();

  gdcm::Anonymizer ano;
  ano.SetFile( reader.GetFile() );
  ano.RemoveGroupLength();
  ano.RemovePrivateTags();

  // PS 3.3 - 2008
  // C.7.1.3 Clinical Trial Subject Module
  // <entry group="0012" element="0010" vr="LO" vm="1" name="Clinical Trial Sponsor Name"/>
  ano.Replace( gdcm::Tag(0x12,0x10), "BigCompany name" );
  // <entry group="0012" element="0020" vr="LO" vm="1" name="Clinical Trial Protocol ID"/>
  ano.Replace( gdcm::Tag(0x12,0x20), "My Clinical Trial Protocol ID" );
  // <entry group="0012" element="0021" vr="LO" vm="1" name="Clinical Trial Protocol Name"/>
  ano.Replace( gdcm::Tag(0x12,0x21), "My Clinical Trial Protocol Name" );
  // <entry group="0012" element="0030" vr="LO" vm="1" name="Clinical Trial Site ID"/>
  ano.Replace( gdcm::Tag(0x12,0x30), "My Clinical Trial Site ID" );
  // <entry group="0012" element="0031" vr="LO" vm="1" name="Clinical Trial Site Name"/>
  ano.Replace( gdcm::Tag(0x12,0x31), "My Clinical Trial Site Name" );
  // <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial Subject ID"/>
  ano.Replace( gdcm::Tag(0x12,0x40), "My Clinical Trial Subject ID" );
  // <entry group="0012" element="0042" vr="LO" vm="1" name="Clinical Trial Subject Reading ID"/>
  ano.Replace( gdcm::Tag(0x12,0x42), "My Clinical Trial Subject Reading ID" );


  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.41 CompressImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 *
 */

#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmWriter.h"
```

```
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"
#include "gdcmImageChangeTransferSyntax.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « filename « std::endl;
    return 1;
    }

  // The output of gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();

  // the dataset is the the set of element we are interested in:
  //gdcm::DataSet &ds = file.GetDataSet();

  gdcm::Image &image = reader.GetImage();
  // image.SetSpacing(0, 0.1);
  // image.SetSpacing(1, 0.2);
  image.Print( std::cout );

  gdcm::ImageChangeTransferSyntax change;
  change.SetTransferSyntax( gdcm::TransferSyntax::JPEG2000Lossless );
  change.SetTransferSyntax( gdcm::TransferSyntax::JPEGLosslessProcess14_1 );
  //change.SetTransferSyntax( gdcm::TransferSyntax::JPEGBaselineProcess1 );
  //change.SetTransferSyntax( image.GetTransferSyntax() );
  change.SetInput( image );
  bool b = change.Change();
  if( !b )
    {
    std::cerr « "Could not change the Transfer Syntax" « std::endl;
    return 1;
    }

  //std::ofstream out( outfilename, std::ios::binary );
  //image.GetBuffer2(out);
  //out.close();
  gdcm::ImageWriter writer;
  writer.SetImage( change.GetOutput() );
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.42 ConvertToQImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to setup the pipeline from a gdcm::ImageReader into a
 * Qt QImage data structure.
 * It only handles 2D image.
 *
 * Ref:
 * http://doc.trolltech.com/4.5/qimage.html
 *
 * Usage:
 *  ConvertToQImage gdcmData/012345.002.050.dcm output.png

 * Thanks:
 *   Sylvain ADAM (sylvain51 hotmail com) for contributing this example
 */

#include "gdcmImageReader.h"
#include <QImage>
#include <QImageWriter>

bool ConvertToFormat_RGB888(gdcm::Image const & gimage, char *buffer, QImage* &imageQt)
{
  const unsigned int* dimension = gimage.GetDimensions();

  unsigned int dimX = dimension[0];
  unsigned int dimY = dimension[1];

  gimage.GetBuffer(buffer);

  // Let's start with the easy case:
  if( gimage.GetPhotometricInterpretation() == gdcm::PhotometricInterpretation::RGB )
    {
    if( gimage.GetPixelFormat() != gdcm::PixelFormat::UINT8 )
      {
      return false;
      }
    unsigned char *ubuffer = (unsigned char*)buffer;
    // QImage::Format_RGB888  13  The image is stored using a 24-bit RGB format (8-8-8).
    imageQt = new QImage((unsigned char *)ubuffer, dimX, dimY, 3*dimX, QImage::Format_RGB888);
    }
  else if( gimage.GetPhotometricInterpretation() == gdcm::PhotometricInterpretation::MONOCHROME2 )
    {
    if( gimage.GetPixelFormat() == gdcm::PixelFormat::UINT8 )
      {
      // We need to copy each individual 8bits into R / G and B:
      unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
      unsigned char *pubuffer = ubuffer;
      for(unsigned int i = 0; i < dimX*dimY; i++)
        {
        *pubuffer++ = *buffer;
        *pubuffer++ = *buffer;
        *pubuffer++ = *buffer++;
        }

      imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
      }
    else if( gimage.GetPixelFormat() == gdcm::PixelFormat::INT16 )
      {
      // We need to copy each individual 16bits into R / G and B (truncate value)
      short *buffer16 = (short*)buffer;
      unsigned char *ubuffer = new unsigned char[dimX*dimY*3];
      unsigned char *pubuffer = ubuffer;
      for(unsigned int i = 0; i < dimX*dimY; i++)
        {
        // Scalar Range of gdcmData/012345.002.050.dcm is [0,192], we could simply do:
        // *pubuffer++ = *buffer16;
        // *pubuffer++ = *buffer16;
        // *pubuffer++ = *buffer16;
        // instead do it right:
        *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
        buffer16++;
        }

      imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
      }
    else
      {
```

```
      std::cerr « "Pixel Format is: " « gimage.GetPixelFormat() « std::endl;
      return false;
      }
    }
  else
    {
    std::cerr « "Unhandled PhotometricInterpretation: " « gimage.GetPhotometricInterpretation() « std::endl;
    return false;
    }

  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::ImageReader ir;
  ir.SetFileName( filename );
  if(!ir.Read())
    {
    //Read failed
    return 1;
    }

  std::cout«"Getting image from ImageReader..."«std::endl;

  const gdcm::Image &gimage = ir.GetImage();
  std::vector<char> vbuffer;
  vbuffer.resize( gimage.GetBufferLength() );
  char *buffer = &vbuffer[0];

  QImage *imageQt = NULL;
  if( !ConvertToFormat_RGB888( gimage, buffer, imageQt ) )
    {
    return 1;
    }

  QImageWriter writer;
  writer.setFormat("png");
  writer.setFileName( outfilename );
  if( !writer.write( *imageQt ) )
    {
    return 1;
    }

  return 0;
}
```

14.43 CreateARGBImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * http://www.w3.org/Graphics/PNG/inline-alpha.html
 * alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
 *
 * $ convert alphatest.png alphatest.rgba
 */

#include "gdcmImageReader.h"
```

```
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.rgba output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  size_t len = gdcm::System::FileSize(filename);
  std::ifstream is(filename, std::ios::binary);

  char * buf = new char[len];
  is.read(buf, len);

  gdcm::ImageWriter writer;
  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 2 );
  unsigned int dims[3] = {};
  dims[0] = 380;
  dims[1] = 287;
  image.SetDimensions( dims );
  gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
  pf.SetSamplesPerPixel( 4 );
  image.SetPixelFormat( pf );
  gdcm::PhotometricInterpretation pi = gdcm::PhotometricInterpretation::ARGB;
  image.SetPhotometricInterpretation( pi );
  image.SetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetByteValue( buf, (uint32_t)len );
  image.SetDataElement( pixeldata );

  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }
  delete[] buf;


  return 0;
}
```

14.44 CreateCMYKImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * http://www.w3.org/Graphics/PNG/inline-alpha.html
 * alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
 *
 * $ convert alphatest.png alphatest.cmyk
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
```

```
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.cmyk output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  size_t len = gdcm::System::FileSize(filename);
  std::ifstream is(filename, std::ios::binary);

  char * buf = new char[len];
  is.read(buf, len);

  gdcm::ImageWriter writer;
  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 2 );
  unsigned int dims[3] = {};
  dims[0] = 380;
  dims[1] = 287;
  image.SetDimensions( dims );
  gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
  pf.SetSamplesPerPixel( 4 );
  image.SetPixelFormat( pf );
  gdcm::PhotometricInterpretation pi = gdcm::PhotometricInterpretation::CMYK;
  image.SetPhotometricInterpretation( pi );
  image.SetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetByteValue( buf, (uint32_t)len );
  image.SetDataElement( pixeldata );

  writer.SetFileName( outfilename );
  if( !writer.Write() )
    {
    return 1;
    }
  delete[] buf;

  return 0;
}
```

14.45 CreateJPIPDataSet.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example was created during the GSOC 2011 project for
 * JPIP
 */
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"
```

```
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " output.dcm" « std::endl;
    return 1;
    }
  const char *outfilename = argv[1];

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );

  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::JPIPReferenced );

  gdcm::Anonymizer anon;
  anon.SetFile( file );

  gdcm::MediaStorage ms = gdcm::MediaStorage::SecondaryCaptureImageStorage;

  gdcm::UIDGenerator gen;
  anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString() );
  std::cout « ms.GetString() « std::endl;
  anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate() );
  //
  anon.Replace( gdcm::Tag(0x0010,0x10), "JPIP^EXAMPLE" );
  anon.Replace( gdcm::Tag(0x0010,0x20), "012345" );
  anon.Empty( gdcm::Tag(0x0010,0x30) );
  anon.Empty( gdcm::Tag(0x0010,0x40) );
  anon.Empty( gdcm::Tag(0x0008,0x20) );
  anon.Empty( gdcm::Tag(0x0008,0x30) );
  anon.Empty( gdcm::Tag(0x0008,0x90) );
  anon.Empty( gdcm::Tag(0x0020,0x10) );
  anon.Empty( gdcm::Tag(0x0020,0x11) );
  anon.Empty( gdcm::Tag(0x0008,0x50) );
  anon.Empty( gdcm::Tag(0x0020,0x0013) );
  anon.Replace( gdcm::Tag(0x0020,0xd), gen.Generate() );
  anon.Replace( gdcm::Tag(0x0020,0xe), gen.Generate() );
  anon.Replace( gdcm::Tag(0x0008,0x64), "WSD " );
  anon.Replace( gdcm::Tag(0x0008,0x60), "OT" );

  gdcm::Attribute<0x0028,0x7FE0> at;
  at.SetValue( "http://dicom.example.com/jpipserver.cgi?target=img.jp2" );
  ds.Insert( at.GetAsDataElement() );

  // Need to retrieve the PixelFormat information from the given file

  if (!w.Write() )
    {
    std::cerr « "Could not write: " « outfilename « std::endl;
    return 1;
    }

  return 0;
}
```

14.46 DeriveSeries.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
```

```
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFileDerivation.h"
#include "gdcmUIDGenerator.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
  {
    return 1;
  }
  const char * ref = argv[1];
  const char * in  = argv[2];

  gdcm::Reader r1;
  r1.SetFileName( ref );
  if( !r1.Read() ) return 1;

  gdcm::Reader r2;
  r2.SetFileName( in );
  if( !r2.Read() ) return 1;


  // Fix Spatial info:
  gdcm::DataSet & ds1 = r1.GetFile().GetDataSet();
  gdcm::File & file2 = r2.GetFile();
  gdcm::DataSet & ds2 = file2.GetDataSet();
  //gdcm::Attribute<0x8,0x8> img_type = { "ORIGINAL", "PRIMARY" };
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0008,0x0008) ));
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0032) ));
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0037) ));
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0088) )); // Spacing between slices
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x0013) )); // Instance Number
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x5100) )); // Patient Position
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0050) )); // Slice Thickness
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0008,0x0070) )); // Manufacturer
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0018,0x0081) )); // Echo Time
  ds2.Replace( ds1.GetDataElement( gdcm::Tag(0x0020,0x1041) )); // Slice Location

  gdcm::Attribute<0x8,0x16> sopclassuid;
  sopclassuid.SetFromDataSet( ds1 );
  gdcm::Attribute<0x8,0x18> sopinstanceuid;
  sopinstanceuid.SetFromDataSet( ds1 );

  // Step 2: DERIVED object
  gdcm::FileDerivation fd;
  fd.AddReference( sopclassuid.GetValue(), sopinstanceuid.GetValue() );

  // http://dicom.nema.org/MEDICAL/dicom/current/output/chtml/part16/chapter_D.html#DCM_121321
  // CID 7202 "Source Image Purposes of Reference"
  // DCM 121321 "Mask image for image processing operation"
  fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121321 );
  // CID 7203 "Image Derivation"
  // DCM 113047 "Pixel by pixel mask"
  fd.SetDerivationCodeSequenceCodeValue( 113047 );
  fd.SetFile( file2 );
  // If all Code Value are ok the filter will execute properly
  if( !fd.Derive() )
    {
    std::cerr « "Sorry could not derive using input info" « std::endl;
    return 1;
    }

  gdcm::Writer w;
  w.SetFile( r2.GetFile() );
  w.SetFileName( "derived.dcm" );
  if( !w.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.47 DiffFile.cxx

```
/*=========================================================================
```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input1.dcm input2.dcm" « std::endl;
    return 1;
    }
  const char *filename1 = argv[1];
  const char *filename2 = argv[2];

  gdcm::Reader reader1;
  reader1.SetFileName( filename1 );
  if( !reader1.Read() )
    {
    return 1;
    }

  gdcm::Reader reader2;
  reader2.SetFileName( filename2 );
  if( !reader2.Read() )
    {
    return 1;
    }

  const gdcm::File &file1 = reader1.GetFile();
  const gdcm::File &file2 = reader2.GetFile();

  const gdcm::DataSet &ds1 = file1.GetDataSet();
  const gdcm::DataSet &ds2 = file2.GetDataSet();

  gdcm::DataSet::ConstIterator it1 = ds1.Begin();
  gdcm::DataSet::ConstIterator it2 = ds2.Begin();

  const gdcm::DataElement &de1 = *it1;
  const gdcm::DataElement &de2 = *it2;
  if( de1 == de2 )
    {
    }
  while( it1 != ds1.End() && it2 != ds2.End() && *it1 == *it2 )
    {
    ++it1;
    ++it2;
    }

  if( it1 != ds1.End() || it2 != ds2.End() )
    {
    std::cerr « "Problem with:" « std::endl;
    if( it1 != ds1.End() )
      {
      std::cerr « "ds1: " « *it1 « std::endl;
      }
    if( it2 != ds2.End() )
      {
      std::cerr « "ds2: " « *it2 « std::endl;
      }
    return 1;
    }

  return 0;
}
```

14.48 DiscriminateVolume.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmScanner.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"
#include "gdcmDirectionCosines.h"
#include <cmath>

/*
 * The following example is a basic sorted which should work in generic cases.
 * It sort files based on:
 * Study Instance UID
 *   Series Instance UID
 *     Frame of Reference UID
 *       Image Orientation (Patient)
 *         Image Position (Patient) (Sorting based on IPP + IOP)
 */

namespace gdcm {
  const Tag t1(0x0020,0x000d); // Study Instance UID
  const Tag t2(0x0020,0x000e); // Series Instance UID
  const Tag t3(0x0020,0x0052); // Frame of Reference UID
  const Tag t4(0x0020,0x0037); // Image Orientation (Patient)

class DiscriminateVolume
{
private:
  std::vector< Directory::FilenamesType > SortedFiles;
  std::vector< Directory::FilenamesType > UnsortedFiles;

  Directory::FilenamesType GetAllFilenamesFromTagToValue(
    Scanner const & s, Directory::FilenamesType const &filesubset, Tag const &t, const char *valueref)
{
  Directory::FilenamesType theReturn;
  if( valueref )
    {
    size_t len = strlen( valueref );
    Directory::FilenamesType::const_iterator file = filesubset.begin();
    for(; file != filesubset.end(); ++file)
      {
      const char *filename = file->c_str();
      const char * value = s.GetValue(filename, t);
      if( value && strncmp(value, valueref, len ) == 0 )
        {
        theReturn.push_back( filename );
        }
      }
    }
  return theReturn;
}

void ProcessAIOP(Scanner const & , Directory::FilenamesType const & subset, const char *iopval)
{
  std::cout « "IOP: " « iopval « std::endl;
  IPPSorter ipp;
  ipp.SetComputeZSpacing( true );
  ipp.SetZSpacingTolerance( 1e-3 ); // ??
  bool b = ipp.Sort( subset );
  if( !b )
    {
    // If you reach here this means you need one more parameter to discriminiat this
    // series. Eg. T1 / T2 intertwinted. Multiple Echo (0018,0081)
    std::cerr « "Failed to sort: " « subset.begin()->c_str() « std::endl;
    for(
      Directory::FilenamesType::const_iterator file = subset.begin();
      file != subset.end(); ++file)
      {
```

```
      std::cerr « *file « std::endl;
      }
    UnsortedFiles.push_back( subset );
    return ;
    }
  ipp.Print( std::cout );
  SortedFiles.push_back( ipp.GetFilenames() );
}

void ProcessAFrameOfRef(Scanner const & s, Directory::FilenamesType const & subset, const char * frameuid)
{
  // In this subset of files (belonging to same series), let's find those
  // belonging to the same Frame ref UID:
  Directory::FilenamesType files = GetAllFilenamesFromTagToValue(
    s, subset, t3, frameuid);

  std::set< std::string > iopset;

  for(
    Directory::FilenamesType::const_iterator file = files.begin();
    file != files.end(); ++file)
    {
    //std::cout « *file « std::endl;
    const char * value = s.GetValue(file->c_str(), gdcm::t4 );
    assert( value );
    iopset.insert( value );
    }
  size_t n = iopset.size();
  if ( n == 0 )
    {
    assert( files.empty() );
    return;
    }

  std::cout « "Frame of Ref: " « frameuid « std::endl;
  if ( n == 1 )
    {
    ProcessAIOP(s, files, iopset.begin()->c_str() );
    }
  else
    {
    const char *f = files.begin()->c_str();
    std::cerr « "More than one IOP: " « f « std::endl;
    // Make sure that there is actually 'n' different IOP
    gdcm::DirectionCosines ref;
    gdcm::DirectionCosines dc;
    for(
      std::set< std::string >::const_iterator it = iopset.begin();
      it != iopset.end(); ++it )
      {
      ref.SetFromString( it->c_str() );
      for(
        Directory::FilenamesType::const_iterator file = files.begin();
        file != files.end(); ++file)
        {
        std::string value = s.GetValue(file->c_str(), gdcm::t4 );
        if( value != it->c_str() )
          {
          dc.SetFromString( value.c_str() );
          const double crossdot = ref.CrossDot(dc);
          const double eps = std::fabs( 1. - crossdot );
          if( eps < 1e-6 )
            {
            std::cerr « "Problem with IOP discrimination: " « file->c_str()
              « " " « it->c_str() « std::endl;
            return;
            }
          }
        }
      }
      // If we reach here this means there is actually 'n' different IOP
    for(
      std::set< std::string >::const_iterator it = iopset.begin();
      it != iopset.end(); ++it )
      {
      const char *iopvalue = it->c_str();
      Directory::FilenamesType iopfiles = GetAllFilenamesFromTagToValue(
        s, files, t4, iopvalue );
      ProcessAIOP(s, iopfiles, iopvalue );
      }
    }
```

```
}

void ProcessASeries(Scanner const & s, const char * seriesuid)
{
  std::cout « "Series: " « seriesuid « std::endl;
  // let's find all files belonging to this series:
  Directory::FilenamesType seriesfiles = GetAllFilenamesFromTagToValue(
    s, s.GetFilenames(), t2, seriesuid);

  gdcm::Scanner::ValuesType vt3 = s.GetValues(t3);
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt3.begin()
    ; it != vt3.end(); ++it )
    {
    ProcessAFrameOfRef(s, seriesfiles, it->c_str());
    }
}

void ProcessAStudy(Scanner const & s, const char * studyuid)
{
  std::cout « "Study: " « studyuid « std::endl;
  gdcm::Scanner::ValuesType vt2 = s.GetValues(t2);
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt2.begin()
    ; it != vt2.end(); ++it )
    {
    ProcessASeries(s, it->c_str());
    }
}
public:

void Print( std::ostream & os )
{
  os « "Sorted Files: " « std::endl;
  for(
    std::vector< Directory::FilenamesType >::const_iterator it = SortedFiles.begin();
    it != SortedFiles.end(); ++it )
    {
    os « "Group: " « std::endl;
    for(
      Directory::FilenamesType::const_iterator file = it->begin();
      file != it->end(); ++file)
      {
      os « *file « std::endl;
      }
    }
  os « "Unsorted Files: " « std::endl;
  for(
    std::vector< Directory::FilenamesType >::const_iterator it = UnsortedFiles.begin();
    it != UnsortedFiles.end(); ++it )
    {
    os « "Group: " « std::endl;
    for(
      Directory::FilenamesType::const_iterator file = it->begin();
      file != it->end(); ++file)
      {
      os « *file « std::endl;
      }
    }

}

  std::vector< Directory::FilenamesType > const & GetSortedFiles() const { return SortedFiles; }
  std::vector< Directory::FilenamesType > const & GetUnsortedFiles() const { return UnsortedFiles; }

void ProcessIntoVolume( Scanner const & s )
{
  gdcm::Scanner::ValuesType vt1 = s.GetValues( gdcm::t1 );
  for(
    gdcm::Scanner::ValuesType::const_iterator it = vt1.begin()
    ; it != vt1.end(); ++it )
    {
    ProcessAStudy( s, it->c_str() );
```

```
    }

}

};

} // namespace gdcm

int main(int argc, char *argv[])
{
  std::string dir1;
  if( argc < 2 )
    {
    const char *extradataroot = nullptr;
#ifdef GDCM_BUILD_TESTING
    extradataroot = gdcm::Testing::GetDataExtraRoot();
#endif
    if( !extradataroot )
      {
      return 1;
      }
    dir1 = extradataroot;
    dir1 += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
  else
    {
    dir1 = argv[1];
    }

  gdcm::Directory d;
  d.Load( dir1, true ); // recursive !

  gdcm::Scanner s;
  s.AddTag( gdcm::t1 );
  s.AddTag( gdcm::t2 );
  s.AddTag( gdcm::t3 );
  s.AddTag( gdcm::t4 );
  bool b = s.Scan( d.GetFilenames() );
  if( !b )
    {
    std::cerr « "Scanner failed" « std::endl;
    return 1;
    }

  gdcm::DiscriminateVolume dv;
  dv.ProcessIntoVolume( s );
  dv.Print( std::cout );

  return 0;
}
```

14.49 DumpADAC.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * the goal of this example is to mimic the behavior of disp_img_header
 * see http://www.gmecorp-usa.com/IM/NM/GC/ADAC/SV/adactechtips/Released_01Q3.pdf
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

#include <iostream>
```

```
#include <fstream>
#include <vector>

#include <string.h>
#include <assert.h>
#include <stdint.h>

struct dict
{
  uint16_t key;
  const char *name;
};

dict Array[] = {
  { 0x01, "Patient name" },
  { 0x02, "Patient ID" },
  { 0x03, "Patient sex" },
  { 0x04, "Patient age" },
  { 0x05, "Patient height" },
  { 0x06, "Patient weight" },
  { 0x07, "Exam date" },
  { 0x08, "Dose admin. time" },
  { 0x09, "Unique exam key" },
  { 0x0a, "Exam procedure" },
  { 0x0b, "Referring physician" },
  { 0x0c, "Attending physician" },
  { 0x0d, "Imaging modality" },
  { 0x0e, "Hospital ID" },
  { 0x0f, "Histogram crv file" },
  { 0x10, "Acq. start time" },
  { 0x11, "Object data type" },
  { 0x12, "Image viewid" },
  { 0x13, "Imaging device name" },
  { 0x14, "Device serial number" },
  { 0x15, "Collimator" },
  { 0x16, "Software version" },
  { 0x17, "Radiopharmaceutical #1" },
  { 0x18, "Energy window #1 center" },
  { 0x19, "Radiopharmaceutical #2" },
  { 0x1a, "Energy window #1 width" },
  { 0x1b, "Isotope imaging mode" },
  { 0x1c, "Energy window #2 center" },
  { 0x1d, "Energy window #2 width" },
  { 0x1e, "Energy window #3 center" },
  { 0x1f, "Energy window #3 width" },
  { 0x20, "Energy window #4 center" },
  { 0x21, "Energy window #4 width" },
  { 0x22, "??Energy window #5 center" },
  { 0x23, "??Energy window #5 width" },
  { 0x24, "Patient orientation" },
  { 0x25, "Spatial resolution" },
  { 0x26, "Slice thickness" },
  { 0x27, "Image X dimension" },
  { 0x28, "Image Y dimension" },
  { 0x29, "Image Z dimension" },
  { 0x2a, "Image pixel width" },
  { 0x2b, "Uniformity corr. file" },
  { 0x2c, "Acquisition zoom factor" },
  { 0x2d, "Total counts in set" },
  { 0x2e, "Time / frame" },
  { 0x2f, "Total acq. time" },
  { 0x30, "Maximum pixel value" },
  { 0x31, "Minimum pixel value" },
  { 0x32, "R-R interval time" },
  { 0x33, "Percent of cycle imaged" },
  { 0x34, "# of cycles accepted" },
  { 0x35, "# of cycles rejected" },
  { 0x36, "Approximate ED frame" },
  { 0x37, "Approximate ES frame" },
  { 0x38, "Approximate EF" },
  { 0x39, "Starting angle" },
  { 0x3a, "Degrees of rotation" },
  { 0x3b, "Direction of rotation" },
  { 0x3c, "Cont. or step/shoot" },
  { 0x3d, "Lim recon start frame" },
  { 0x3e, "Upper window grey shade" },
  { 0x3f, "Lower lvl grey shade" },
  { 0x40, "Associated color map" },
  { 0x41, "Custom color map file" },
  { 0x42, "Manipulated image" },
  { 0x43, "Axis of rotation corr." },
```

```
  { 0x44, "Reorientation azimuth" },
  { 0x45, "Reorientation elevation" },
  { 0x46, "Filter type" },
  { 0x47, "Filter order" },
  { 0x48, "Filter cutoff frequency" },
  { 0x49, "Reconstruction type" },
  { 0x4a, "Attenuation coefficient" },
  { 0x4b, "Associated parent file" },
  { 0x4c, "Unique patient key" },
  { 0x52, "Normalization crv file" },
  { 0x53, "Unique object key" },
  { 0x54, "This phase of VFR is" },
  { 0x55, "True color value" },
  { 0x56, "# of sets of x,y,z grps" },
  { 0x57, "Scale factor of set" },
  { 0x6d, "Date of birth" },
  { 0x6e, "Directional orientation" },
  { 0x6f, "Number of VFR studies" },
  { 0x70, "R-R low tolerance" },
  { 0x71, "R-R high tolerance" },
  { 0x72, "Prog specific results:" },

  { 0x99, nullptr }
};

void printname( int , int , uint16_t v )
{
  if( v == 0x1 )
    {
    std::cout « "DATABASE PARAMETERS" « std::endl;
    std::cout « "____________________" « std::endl;
    }
  else if( v == 0x27 )
    {
    std::cout « "IMAGE PARAMETERS" « std::endl;
    std::cout « "_________________" « std::endl;
    }
  else if( v == 0x13 )
    {
    std::cout « "EXTRA PARAMETERS" « std::endl;
    std::cout « "_________________" « std::endl;
    }
  else if( v == 0x2e )
    {
    std::cout « "*** NOT CURRENTLY USED :" « std::endl;
    }
  static const unsigned int n = sizeof( Array ) / sizeof( *Array ) - 1;
  for( unsigned int i = 0; i < n; ++i )
    {
    if( v == Array[i].key )
      {
      std::cout « /*"" « std::dec « len « "," « mult « " " « */ Array[i].name;
      std::cout « " : ";
      return;
      }
    }
  std::cout « /*"\t# " « std::dec « len « "," « mult « */ std::hex « v « "\t: ";
}

uint16_t readint16(std::istream &is )
{
  uint16_t val;
  is.read( (char*)&val, sizeof( val ));
  return (uint16_t)((val»8) | (val«8));
}

uint32_t readint32(std::istream &is )
{
  uint32_t val;
  is.read( (char*)&val, sizeof( val ));
  val= ((val«8)&0xFF00FF00) | ((val»8)&0x00FF00FF);
  return (val»16) | (val«16);
}

float readfloat32(std::istream &is )
{
  union { uint32_t val; float f;} dual;
  dual.val = readint32(is);
  return dual.f;
```

```
}

struct el
{
  uint16_t v1;
  uint16_t v2;
  uint16_t v3;
  void read( std::istream & is )
    {
    v1 = readint16(is);
    v2 = readint16(is);
    v3 = readint16(is);
    }
  void print( std::ostream & os )
    {
    os « std::hex « v1 « "\t" « v2 « "\t" « v3 « std::endl;
    }
};

std::vector<el> Vel;

void readelement( std::istream & is )
{
  el e;
  e.read( is );
  Vel.push_back( e );
}

void printascii( uint16_t tag, const char *buffer, size_t len )
{
  std::ostream & os = std::cout;
  if( tag == 0x72 )
    {
    os « "\n  ";
    for(size_t i = 0; i < len; ++i)
      {
      const char &c = buffer[i];
      if( c == 0x0 ) os « "!";
      else if( c == 0x0f ) os « " ";
      else if( c == 0x17 ) os « ":";
      else if( c == 0x14 ) os « ":";
      else if( c == 0x10 ) os « ":";
      else if( c == 0x16 ) os « ":";
      else if( c == 0x08 ) os « ":";
      else if( c == 0x0b ) os « ":";
      else if( c == 0x0e ) os « ":";
      else if( c == 0x07 ) os « ":";
      else os « c;
      }
    os « "";
    }
  else
    {
    (void)len;
    os « "" « buffer « "";
    }
}

bool DumpADAC( std::istream & is )
{
  std::ostream &os = std::cout;

  char magic[6 + 1];
  magic[6] = 0;
  is.read( magic, 6);
//  std::cout « magic « " ";
  assert( strcmp( magic, "adac01" ) == 0 );
  int c = is.get();
  assert( c == 0 ); (void)c;
  c = is.get();
  assert( c == 'X' );

  uint16_t v;
  v = readint16(is);
//  std::cout « v « std::endl;
  assert( v == 512 ); (void)v; // ??

  int nel = 87;
  for (int i = 0; i <= nel; ++i )
    {
    readelement( is );
```

```
    }

  char buffer[512];
  for( int i = 0; i <= nel; ++i )
    {
    const el &e = Vel[i];
    int diff;
    if( i == nel )
      {
      diff = 2048 - e.v3;
      if( diff > 512 ) diff = 512;
      }
    else
      {
      const el &enext = Vel[i+1];
      diff = enext.v3 - e.v3;
      }
    is.seekg( e.v3, std::ios::beg );
    //std::cout « "(" « std::hex « std::setw( 2 ) « std::setfill( '0' ) « e.v1 « ") " « std::hex « std::setw( 3
      ) « std::setfill( '0' ) « e.v2 « " ";
    printname( diff, 0, e.v1 );
    int mult = 1;
    if( e.v2 == 0 )
      {
      is.read( buffer, diff);
      buffer[ diff ] = 0;
      printascii( e.v1, buffer, diff);
      }
    else if( e.v2 == 0x100 )
      {
      mult = diff / 2;
      assert( diff == 2 * mult );
      for ( int ii = 0; ii < mult; ++ii )
        {
        if ( ii ) os « "\\";
        uint16_t val = readint16(is);
        os « "" « std::dec « val « "";
        }
      }
    else if( e.v2 == 0x200 )
      {
      assert( diff == 4 );
      uint32_t val = readint32(is);
      os « "" « std::dec « val « "";
      }
    else if( e.v2 == 0x300 )
      {
      assert( diff == 4 );
      float val = readfloat32(is);
      os « "" « std::dec « val « "";
      }
    else
      {
      assert( 0 );
      }
    os « std::endl;
    }
  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  // (0019,1061) UN (OB) 61\64\61\63\30         # 2048,1 Ver200 ADAC Pegasys Headers
  const gdcm::PrivateTag tver200adacpegasysheaders(0x0019,0x61,"ADAC_IMG");
  if( !ds.FindDataElement( tver200adacpegasysheaders ) ) return 1;
  const gdcm::DataElement& ver200adacpegasysheaders = ds.GetDataElement( tver200adacpegasysheaders );
  if ( ver200adacpegasysheaders.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = ver200adacpegasysheaders.GetByteValue();

  // (0019,1021) US 1               # 2,1 Ver200 Number of ADAC Headers
```

```
  // TODO

  // (0019,1041) IS [2048\221184 ]  # 12,1-n Ver200 ADAC Header/Image Size
  if( bv->GetLength() != 2048 ) return 1;

  gdcm::Element<gdcm::VR::IS,gdcm::VM::VM2> el;
  const gdcm::PrivateTag tver200adacheaderimagesize(0x0019,0x41,"ADAC_IMG");
  if( !ds.FindDataElement( tver200adacheaderimagesize ) ) return 1;
  const gdcm::DataElement& ver200adacheaderimagesize = ds.GetDataElement( tver200adacheaderimagesize );
  el.SetFromDataElement( ver200adacheaderimagesize );
  if( el.GetValue(0) != 2048 ) return 1;

  std::istringstream is;
  std::string dup( bv->GetPointer(), bv->GetLength() );
  is.str( dup );
  bool b = DumpADAC( is );
  if( !b ) return 1;



  return 0;
}
```

14.50 DumpExamCard.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*

  Try to extract contents of Philips RAW storage class:

(0002,0002) UI [1.2.840.10008.5.1.4.1.1.66]                        # 26,1 Media Storage SOP Class UID
(0002,0003) UI [1.3.46.670589.11.17240.5.23.4.1.3012.2010032409482568018]         # 56,1 Media Storage SOP
      Instance UID
(0002,0010) UI [1.2.840.10008.1.2.1]                               # 20,1 Transfer Syntax UID
(0002,0012) UI [1.3.46.670589.11.0.0.51.4.4.1]                     # 30,1 Implementation Class UID
(0002,0013) SH [MR DICOM 4.1]                                      # 12,1 Implementation Version Name

 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Triplett,William T for bringing to your attention on this ExamCard stuff
 */
#include "gdcmReader.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmBase64.h"

#include <iomanip>

static bool compfn(const char *s1, const char *s2)
{
  return strcmp(s1,s2) < 0 ? true : false;
}

static const char *PDFStrings[] = { // Keep me ordered please
  "COILSTATE", // series of string ?
  "HARDWARE_CONFIG", // series of number ?
```

```
  "PDF_CONTROL_GEN_PARS",
  "PDF_CONTROL_PREP_PARS",
  "PDF_CONTROL_RECON_PARS",
  "PDF_CONTROL_SCAN_PARS",
  "PDF_EXAM_PARS",
  "PDF_HARDWARE_PARS",
  "PDF_PREP_PARS",
  "PDF_PRESCAN_COIL_PARS",
  "PDF_SPT_PARS",
};

static bool isvalidpdfstring( const char *pdfstring )
{
  assert( pdfstring );
  static const size_t n = sizeof( PDFStrings ) / sizeof( *PDFStrings );
  static const char **begin = PDFStrings;
  static const char **end = begin + n;
  return std::binary_search(begin, end, pdfstring, compfn);
}

typedef enum
{
  param_float = 0,
  param_integer = 1, // 1 « 0
  param_string = 2, // 1 « 1
  param_3, // ??
  param_enum = 4  // 1 « 2
} param_type;

static const char *gettypenamefromtype( int i)
{
  const char *ret = nullptr;
  param_type e = (param_type)i;
  switch( e )
    {
  case param_float:
    ret = "float";
    break;
  case param_integer:
    ret = "int";
    break;
  case param_string:
    ret = "string";
    break;
  case param_3:
    ret = "??";
    break;
  case param_enum:
    ret = "enum";
    break;
    }
  assert( ret );
  return ret;
}

struct header
{
/*
 * TODO:
 * Looks as if we could read all int*, float* and string* at once...
 */
  int32_t v1; // offset to int pointer array ?
  uint16_t nints; // number of ints (max number?)
  uint16_t v3; // always 0 ?
  int32_t v4; // offset to float pointer array ?
  uint32_t nfloats;
  int32_t v6; // offset to string pointer array ?
  uint32_t nstrings;
  int32_t v8; // always 8 ??
  uint32_t numparams;
  uint32_t getnints() const { return nints; }
  uint32_t getnfloats() const { return nfloats; }
  uint32_t getnstrings() const { return nstrings; }
  uint32_t getnparams() const { return numparams; }
  void read( std::istream & is )
    {
    is.read( (char*)&v1,sizeof(v1));
    if( v1 == 0x01 ) {
      // direct (FIXME how should we detect this, much like TIFF ???)
      nints = 0;
      v3 = 0;
```

```
      v4 = 0;
      nfloats = 0;
      v6 = 0;
      nstrings = 0;
      v8 = 0;
      numparams = 0;
      uint32_t bla;
      is.read( (char*)&bla, sizeof(bla) );
      assert( bla == 0x2 || bla == 0x3 );
      nstrings = 1;
      numparams = 1;
    } else {
      // indirect
      is.read( (char*)&nints,sizeof(nints));
      is.read( (char*)&v3,sizeof(v3));
      assert( v3 == 0 ); // looks like this is always 0
      is.read( (char*)&v4,sizeof(v4));
      is.read( (char*)&nfloats,sizeof(nfloats));
      is.read( (char*)&v6,sizeof(v6));
      is.read( (char*)&nstrings,sizeof(nstrings));
      is.read( (char*)&v8,sizeof(v8));
      assert( v8 == 8 );
      is.read( (char*)&numparams,sizeof(numparams));
    }
    }
  void print( std::ostream & os )
    {
    os « v1 « ",";
    os « nints « ",";
    os « v3 « ",";
    os « v4 « ",";
    os « nfloats « ",";
    os « v6 « ",";
    os « nstrings « ",";
    os « v8 « ",";
    os « numparams « std::endl;
    }
};

struct param
{
  char name[32+1];
  uint8_t boolean;
  int32_t type;
  uint32_t dim;
  union {
  uint32_t val;
  char * ptr; } v4;
  int32_t /*std::streamoff*/ offset;
  param_type gettype() const { return (param_type)type; }
  uint32_t getdim() const { return dim; }
  void read_direct_int( std::istream & is ) {
    uint32_t bla;
    int max = 9;
    std::vector<uint32_t> v;
    for( int i = 0; i < max; ++i ) {
      is.read( (char*)&bla, sizeof(bla) );
      v.push_back( bla );
    }
    is.read( (char*)&bla, sizeof(bla) );
    char name0[32];
    memset(name0,0,sizeof(name0));
    assert( bla < sizeof(name0) );
    is.read( name0, bla);
    size_t l = strlen(name0);
    assert( l == bla ); (void)l;
    char * ptr = strdup( name0 );
    v4.ptr = ptr;
    type = param_string;
    dim = 1;
    offset = 0; // important !
  }
  void read_direct_string( std::istream & is ) {
    uint32_t bla;
    is.read( (char*)&bla, sizeof(bla) );
    char name0[32];
    memset(name0,0,sizeof(name0));
    assert( bla < sizeof(name0) );
    is.read( name0, bla);
    size_t l = strlen(name0);
    assert( l == bla ); (void)l;
```

```
  memcpy( this->name, name0, bla );
  is.read( (char*)&bla, sizeof(bla) );
  assert( bla == 0x1 );
  is.read( (char*)&bla, sizeof(bla) );
  char value[32];
  memset(value,0,sizeof(value));
  assert( bla < sizeof(value) );
  is.read( value, bla);
  is.read( (char*)&bla, sizeof(bla) );
  assert( bla == 0 ); // trailing stuff ?
  is.read( (char*)&bla, sizeof(bla) );
  assert( bla == 0 ); // trailing stuff ?
  const uint32_t cur = (uint32_t)is.tellg();
  std::cerr « "offset:" « cur « std::endl;
  if( cur == 65 )
    is.read( (char*)&bla, 1 );
  else if( cur == 66 )
    is.read( (char*)&bla, 1 );
  else if( cur == 122 )
    is.read( (char*)&bla, 2 );
  else
    assert(0);
  type = param_string;
  dim = 1;
  // FIXME: store the value in v4 for now:
  char * ptr = strdup( value );
  v4.ptr = ptr;
  offset = 0; // important !
}
void read( std::istream & is )
  {
  is.read( name, 32 + 1);
  // This is always the same issue the string can contains garbage from previous run,
  // we need to print only until the first \0 character:
  assert( strlen( name ) <= 32 );
  is.read( (char*)&boolean,1);
  assert( boolean == 0 || boolean == 1 || boolean == 0x69 ); // some kind of bool, or digital trash ?
  is.read( (char*)&type, sizeof( type ) );
  assert( gettypenamefromtype( type ) );
  is.read( (char*)&dim, sizeof( dim ) ); // number of elements
  is.read( (char*)&v4.val, sizeof( v4.val ) );
  //assert( v4.val == 0 ); // always 0 ? sometimes not...
  const uint32_t cur = (uint32_t)is.tellg();
  is.read( (char*)&offset, sizeof( offset ) );
  assert( offset != 0 );
  offset += cur;
  }

void print( std::ostream & os ) const
  {
  os « name « ",";
  os « (int)boolean « ",";
  os « type « ",";
  os « dim « ",";
  os « v4.val « ",";
  os « offset « std::endl;
  }
void printvalue( std::ostream & os, std::istream & is ) const
  {
  if( offset ) {
  is.seekg( offset );
  switch( type )
    {
  case param_float:
      {
      os.precision(2);
      os « std::fixed;
      for( uint32_t idx = 0; idx < dim; ++idx )
        {
        if( idx ) os « ",";
        float v;
        is.read( (char*)&v, sizeof(v) );
        os « v; // what if the string contains \0 ?
        }
      }
    break;
  case param_integer:
      {
      int32_t v;
      for( uint32_t idx = 0; idx < dim; ++idx )
        {
```

```
          if( idx ) os « ",";
          is.read( (char*)&v, sizeof(v) );
          os « v;
          }
        }
      break;
    case param_string:
        {
        int size = 81;
        std::string v;
        v.resize( size );
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os « ";";
          is.read( &v[0], size );
          os « v.c_str();
          }
        }
      break;
    case param_enum:
        {
        int32_t v;
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os « ",";
          is.read( (char*)&v, sizeof(v) );
          os « v;
          }
        }
      break;
      }
    } else {
#if 1
      // direct
      assert ( type == param_string );
      char * ptr = v4.ptr;
        //std::string v;
        //v.resize( dim );
        //is.read( &v[0], dim );
        os « ptr;
#endif
    }

    }
  void printxml( std::ostream & os, std::istream & is ) const
    {
    // <Attribute Name="CGEN_force_par_mode" Type="enum">0</Attribute>
    os « "  <Attribute";
    os « " Name=\"" « name « "\"";
    os « " Type=\"" « gettypenamefromtype(type) « "\"";
    if( dim != 1 )
      {
      os « " ArraySize=\"" « dim « "\"";
      }
    os « ">";
    printvalue( os, is );
    os « "</Attribute>\n";
    }
  void printcsv( std::ostream & os, std::istream & is ) const
    {
    os « std::setw(32) « std::left « name « ",";
    os « std::setw(7) « std::right « gettypenamefromtype(type) « ",";
    os « std::setw(4) « dim « ",";
    os « " ";
    printvalue( os, is );
    os « ",\n";
    }
};

static bool ProcessNested( gdcm::DataSet & ds )
{
  /*
  TODO:
   Looks like the real length of the blob is stored here:
(2005,1132) SQ                                                  # u/l,1 ?
  (fffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ]                 # 26,1 Private Creator
    (2005,1143) SL 3103                                         # 4,1 ?

Wotsit ?
(2005,1132) SQ                                                  # u/l,1 ?
```

```
  (fffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ]                    # 26,1 Private Creator
    (2005,1147) CS [Y ]                                            # 2,1 ?
  */
  bool ret = false;

  //  (2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS]                    # 20,1 Protocol Data Name
  const gdcm::PrivateTag pt0(0x2005,0x37,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt0 ) ) return false;
  const gdcm::DataElement &de0 = ds.GetDataElement( pt0 );
  if( de0.IsEmpty() ) return false;
  const gdcm::ByteValue * bv0 = de0.GetByteValue();
  std::string s0( bv0->GetPointer() , bv0->GetLength() );

  // (2005,1139) LO [IEEE_PDF]                                     # 8,1 Protocol Data Type
  const gdcm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt1 ) ) return false;
  const gdcm::DataElement &de1 = ds.GetDataElement( pt1 );

  // (2005,1143) SL 53                                             # 4,1 Protocol Data Block Length (non-padded)
  const gdcm::PrivateTag pt2(0x2005,0x43,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt2 ) ) return false;
  const gdcm::DataElement &de2 = ds.GetDataElement( pt2 );

  // (2005,1147) CS [Y ]                                           # 2,1 Protocol Data Boolean
  const gdcm::PrivateTag pt3(0x2005,0x47,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt3 ) ) return false;
  const gdcm::DataElement &de3 = ds.GetDataElement( pt3 );
  (void)de3;

  // (2005,1144) OW 00\00\00\00\05\00\00\00\35\2e\31\2e\37\00         # 54,1 Protocol Data Block
  const gdcm::PrivateTag pt(0x2005,0x44,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt ) ) return false;
  const gdcm::DataElement &de = ds.GetDataElement( pt );
  if( de.IsEmpty() ) return false;
  const gdcm::ByteValue * bv = de.GetByteValue();

  if( s0 == "ExamCardBlob" )
    {
    assert( de1.IsEmpty() );

    std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
    fn += ".xml";
    std::ofstream out( fn.c_str() );

    // remove trailing \0
    size_t len = strlen( bv->GetPointer() );
    out.write( bv->GetPointer() , len );
    out.close();

    // Extract binary64 thingy (this is a ugly hack, better use an XML parser)
    std::string dup( bv->GetPointer(), len );
    std::string::size_type pos1 = dup.find( "<ExamCardBlob>" );
    std::string::size_type pos2 = dup.find( "</ExamCardBlob>" );

    std::string b64( bv->GetPointer() + pos1 + 14, pos2 - (pos1 + 14) );

    // ulgy hack to remove \r\n from input base64:
    std::string::iterator r_pos = std::remove(b64.begin(), b64.end(), '\r');
    b64.erase(r_pos, b64.end());
    std::string::iterator n_pos = std::remove(b64.begin(), b64.end(), '\n');
    b64.erase(n_pos, b64.end());
#if 0
    std::ofstream out2( "debug" );
    out2.write( b64.c_str(), b64.size() );
    out2.close();
#endif

    const size_t dlen = gdcm::Base64::GetDecodeLength(b64.c_str(), b64.size() );

    std::string decoded;
    decoded.resize( dlen );
    gdcm::Base64::Decode( &decoded[0], decoded.size(), b64.c_str(), b64.size() );

    std::ofstream f64( "soap.xml" );
    f64.write( decoded.c_str(), decoded.size() );
    f64.close();

    ret = true;
    }
  else
```

```
    {
    if( de1.IsEmpty() ) return false;
    const gdcm::ByteValue * bv1 = de1.GetByteValue();
    gdcm::Element<gdcm::VR::SL,gdcm::VM::VM1> dlen = {{0L}};
    dlen.SetFromDataElement( de2 );
    std::string s1( bv1->GetPointer() , bv1->GetLength() );

    if( s1 == "IEEE_PDF" )
      {

      std::istringstream is;
      assert( bv->GetLength() == (size_t)dlen.GetValue() || bv->GetLength() == (size_t)(dlen.GetValue() + 1) );
      std::string dup( bv->GetPointer(), dlen.GetValue() /*bv->GetLength()*/ );
      is.str( dup );

      header h;
      h.read( is );
      //assert( is.peek() && is.eof() );
#if 1
      static int c = 0;
      std::string fn0 = gdcm::LOComp::Trim( s1.c_str() ); // remove trailing space
      std::stringstream ss;
      ss « fn0 « "_" « c++;
      if( h.v1 == 0x01 )
        ss « ".direct";
      else
        ss « ".indirect";
      std::cout « "fn0=" « ss.str() « " Len= " « bv->GetLength() « std::endl;
      std::ofstream out( ss.str().c_str() );
      out.write( bv->GetPointer(), bv->GetLength() );
      out.close();
#endif
#if 1
      std::cout « dup.c_str() « std::endl;
      h.print( std::cout );
#endif

      std::vector< param > params;
      if( h.v1 == 0x01 ) {
        for( uint32_t i = 0; i < 1 /* h.getnparams()*/; ++i ) {
          param p;
          if( s0 == "HARDWARE_CONFIG " )
            {
            p.read_direct_int( is );
            }
          else if( s0 == "COILSTATE " )
            {
            p.read_direct_string( is );
            }
          else
            {
            assert(0);
            }
          params.push_back( p );
        }
      } else {
        assert( is.tellg() == std::streampos(0x20) );
        is.seekg( 0x20 );

        param p;
        for( uint32_t i = 0; i < h.getnparams(); ++i )
          {
          p.read( is );
          //p.print( std::cout );
          params.push_back( p );
          }
      }

      std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
      bool b1 = isvalidpdfstring( fn.c_str() );
      assert( b1 ); (void)b1;
      fn += ".csv";
      //fn += ".xml";
      std::ofstream csv( fn.c_str() );

      // let's do some bookkeeping:
      uint32_t nfloats = 0;
      uint32_t nints = 0;
      uint32_t nstrings = 0;
      for( std::vector<param>::const_iterator it = params.begin();
        it != params.end(); ++it )
```

```
        {
        param_type type = it->gettype();
        switch( type )
          {
        case param_float:
          nfloats += it->getdim();
          break;
        case param_integer:
          nints += it->getdim();
          break;
        case param_string:
          nstrings += it->getdim();
          break;
        default:
          ;
          }
        }
#if 0
      std::cout « "Stats:" « std::endl;
      std::cout « "nfloats:" « nfloats « std::endl;
      std::cout « "nints:" « nints « std::endl;
      std::cout « "nstrings:" « nstrings « std::endl;
#endif
      assert( h.getnints() >= nints );
      assert( h.getnfloats() >= nfloats );
      assert( h.getnstrings() >= nstrings);

      for( uint32_t i = 0; i < h.getnparams(); ++i )
        {
        params[i].printcsv( csv, is );
        //params[i].printxml( csv, is );
        }
      csv.close();
      ret = true;
      }
    else if( s1 == "ASCII " )
      {
#if 0
      std::cerr « "ASCII is not handled" « std::endl;
      std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
      fn += ".asc";
      std::ofstream out( fn.c_str() );
      out.write( bv->GetPointer() , bv->GetLength() );
      out.close();
#endif
      std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
      fn += ".sin";
      std::ofstream sin( fn.c_str() );

      const char *beg = bv->GetPointer();
      const char *end = beg + bv->GetLength();
      assert( *beg == 0 );
      const char *p = beg + 1; // skip first \0
      size_t prev = 0;
      for( ; p != end; ++p )
        {
        if( *p == 0 )
          {
          const char *s = beg + prev + 1;
          if( *s )
            {
            sin « s « std::endl;
            }
          else
            {
            sin « std::endl;
            }
          prev = p - beg;
          }
        }
      sin.close();

      ret = true;
      }
    else if( s1 == "BINARY" )
      {
      std::cerr « "BINARY is not handled" « std::endl;
      std::string fn = gdcm::LOComp::Trim( s0.c_str() ); // remove trailing space
      fn += ".bin";
      std::ofstream out( fn.c_str() );
      //out.write( bv->GetPointer() + 512, bv->GetLength() - 512);
```

```
      out.write( bv->GetPointer() , bv->GetLength() );
      out.close();

#if 0
      int array[ 128 ];
      memcpy( array, bv->GetPointer(), 512 );
      for( int i = 0; i < 14; ++i )
        {
        std::cout « array[i] « std::endl;
        }
#endif

      ret = true;
      }
    }
  // else -> ret == false
  assert( ret );

  return ret;
}

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
/*
(2005,1132) SQ                                          # u/l,1 ?
  (fffe,e000) na (Item with undefined length)
    (2005,0011) LO [Philips MR Imaging DD 002 ]         # 26,1 Private Creator
    (2005,1137) PN (LO) [PDF_CONTROL_GEN_PARS]          # 20,1 ?
    (2005,1138) PN (LO) (no value)                      # 0,1 ?
    (2005,1139) PN (LO) [IEEE_PDF]                      # 8,1 ?
    (2005,1140) PN (LO) (no value)                      # 0,1 ?
    (2005,1141) PN (LO) (no value)                      # 0,1 ?
    (2005,1143) SL 3103                                 # 4,1 ?
    (2005,1144) OW
      66\05\00\00\3b\01\00\00\4a\0a\00\00\0e\00\00\00\7a\0a\00\00\95\01\00\00\08\00\00\00\1b\00\00\00\43\47\45\4e\5f\75\73\65\72\5
      # 3104,1 ?
    (2005,1147) CS [Y ]                                 # 2,1 ?
  (fffe,e00d)
*/
  const gdcm::PrivateTag pt(0x2005,0x32,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt ) ) return 1;
  const gdcm::DataElement &de = ds.GetDataElement( pt );
  if( de.IsEmpty() ) return 1;

  gdcm::SequenceOfItems *sqi = de.GetValueAsSQ();
  if ( !sqi ) return 1;
  gdcm::SequenceOfItems::SizeType s = sqi->GetNumberOfItems();
  for( gdcm::SequenceOfItems::SizeType i = 1; i <= s; ++i )
    {
    gdcm::Item &item = sqi->GetItem(i);

    gdcm::DataSet &nestedds = item.GetNestedDataSet();

    if( !ProcessNested( nestedds ) ) {
      std::cerr « "Error processing Item #" « i « std::endl;
    }
    }

  return 0;
}
```

14.51 DumpGEMSMovieGroup.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```
  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

bool PrintNameValueMapping( gdcm::SequenceOfItems *sqi_values,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
  using namespace gdcm;
  // prepare names mapping:
  typedef VRToType<VR::UL>::Type UL;
  std::map< UL, std::string > names;
  assert( sqi_names );
  assert( sqi_values );
  SequenceOfItems::SizeType s = sqi_names->GetNumberOfItems();
  PrivateTag tindex(0x7fe1,0x71,"GEMS_Ultrasound_MovieGroup_001");
  PrivateTag tname (0x7fe1,0x72,"GEMS_Ultrasound_MovieGroup_001");
  // First sequence contains all possible names (this is a dict)
  for( SequenceOfItems::SizeType i = 1; i <= s; ++i )
    {
    const Item & item = sqi_names->GetItem( i );
    const DataSet & ds = item.GetNestedDataSet();
    if( !ds.FindDataElement( tindex )
      || !ds.FindDataElement( tname ) )
      {
      assert( 0 );
      return false;
      }
    const DataElement & index = ds.GetDataElement( tindex );
    const DataElement & name = ds.GetDataElement( tname );
    if( index.IsEmpty() || name.IsEmpty() )
      {
      assert( 0 );
      return false;
      }
    gdcm::Element<VR::UL, VM::VM1> el1;
    el1.SetFromDataElement( index );

    gdcm::Element<VR::LO, VM::VM1> el2;
    el2.SetFromDataElement( name );
//    std::cout « el1.GetValue() « " " « el2.GetValue() « std::endl;
    names.insert( std::make_pair( el1.GetValue(), el2.GetValue() ) );
    }

  SequenceOfItems::SizeType s2 = sqi_values->GetNumberOfItems();
  assert( s2 <= s );
  PrivateTag tindex2(0x7fe1,0x48,"GEMS_Ultrasound_MovieGroup_001");
  for( SequenceOfItems::SizeType i = 1; i <= s2; ++i )
    {
    const Item & item = sqi_values->GetItem( i );
    const DataSet & ds = item.GetNestedDataSet();
    if( !ds.FindDataElement( tindex2 ) )
      {
      assert( 0 );
      return false;
      }
    const DataElement & index2 = ds.GetDataElement( tindex2 );
    if( index2.IsEmpty() )
      {
      assert( 0 );
      return false;
      }
    gdcm::Element<VR::FD, VM::VM1_2> el1;
```

```
    el1.SetFromDataElement( index2 );

    UL copy = (UL)el1.GetValue();
#if 1
    std::cout « indent;
    std::cout « "( " « names[ copy ];
#endif
    // (7fe1,1052) FD 1560                                      # 8,1 ?
    // (7fe1,1057) LT [MscSkelSup]                              # 10,1 ?
    //PrivateTag tvalue(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001");
    PrivateTag tvalueint(0x7fe1,0x49,"GEMS_Ultrasound_MovieGroup_001"); // UL
    PrivateTag tvaluefloat1(0x7fe1,0x51,"GEMS_Ultrasound_MovieGroup_001"); // FL
    PrivateTag tvaluefloat(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001"); // FD
    PrivateTag tvalueul(0x7fe1,0x53,"GEMS_Ultrasound_MovieGroup_001"); // UL
    PrivateTag tvaluesl(0x7fe1,0x54,"GEMS_Ultrasound_MovieGroup_001"); // SL
    PrivateTag tvalueob(0x7fe1,0x55,"GEMS_Ultrasound_MovieGroup_001"); // OB
    PrivateTag tvaluetext(0x7fe1,0x57,"GEMS_Ultrasound_MovieGroup_001");  // LT
    PrivateTag tvaluefd(0x7fe1,0x77,"GEMS_Ultrasound_MovieGroup_001");  // FD / 1-N
    PrivateTag tvaluesl3(0x7fe1,0x79,"GEMS_Ultrasound_MovieGroup_001");  // SL / 1-N
    PrivateTag tvaluesl2(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001");  // SL ??
    PrivateTag tvaluefd1(0x7fe1,0x87,"GEMS_Ultrasound_MovieGroup_001");  // FD / 1-N
    PrivateTag tvaluefloat2(0x7fe1,0x88,"GEMS_Ultrasound_MovieGroup_001");  // FD ??
#if 1
    std::cout « " ) = ";
#endif
    if( ds.FindDataElement( tvalueint ) )
      {
      const DataElement & value = ds.GetDataElement( tvalueint );
      gdcm::Element<VR::UL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat1 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefloat1 );
      gdcm::Element<VR::FL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefloat );
      gdcm::Element<VR::FD,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvaluesl ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluesl );
      gdcm::Element<VR::SL,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvalueul ) )
      {
      const DataElement & value = ds.GetDataElement( tvalueul );
      gdcm::Element<VR::UL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      assert( el2.GetLength() == 1 );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvalueob ) )
      {
      const DataElement & value = ds.GetDataElement( tvalueob );
//      gdcm::Element<VR::SL,VM::VM1> el2;
//      el2.SetFromDataElement( value );
//      std::cout « el2.GetValue() « std::endl;
      std::cout « value « std::endl;
      }
    else if( ds.FindDataElement( tvaluetext ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluetext );
      gdcm::Element<VR::LT,VM::VM1> el2;
      el2.SetFromDataElement( value );
      std::cout « el2.GetValue() « std::endl;
      }
    else if( ds.FindDataElement( tvaluesl2 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluesl2 );
      gdcm::Element<VR::SL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
```

```
      el2.Print( std::cout );
      assert( el2.GetLength() == 4 );
      std::cout « std::endl;
      }
    else if( ds.FindDataElement( tvalues13 ) )
      {
      const DataElement & value = ds.GetDataElement( tvalues13 );
      gdcm::Element<VR::SL,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
//      assert( el2.GetLength() == 4 );
      std::cout « std::endl;
      }
    else if( ds.FindDataElement( tvaluefd ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefd );
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
//      assert( el2.GetLength() == 4 || el2.GetLength() == 3 || el2.GetLength() == 8 );
      std::cout « std::endl;
      }
    else if( ds.FindDataElement( tvaluefloat2 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefloat2 );
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
      assert( el2.GetLength() == 2 );
      std::cout « std::endl;
      }
    else if( ds.FindDataElement( tvaluefd1 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluefd1 );
      gdcm::Element<VR::FD,VM::VM1_n> el2;
      el2.SetFromDataElement( value );
      el2.Print( std::cout );
      assert( el2.GetLength() == 4 );
      std::cout « std::endl;
      }
    else
      {
      std::cout « "(no value)" « std::endl;
//      std::cout « ds « std::endl;
      assert( ds.Size() == 2 );
      }
    }
  return true;
}

bool PrintNameValueMapping2( gdcm::PrivateTag const & privtag, const gdcm::DataSet & ds ,
  gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
  if( !ds.FindDataElement( privtag ) ) return false;
  const gdcm::DataElement& seq_values = ds.GetDataElement( privtag );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = seq_values.GetValueAsSQ();

  return PrintNameValueMapping( sqi, sqi_names, indent);
}

bool PrintNameValueMapping3( gdcm::PrivateTag const & privtag1, gdcm::PrivateTag const & privtag2, const
      gdcm::DataSet & ds ,
gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
  if( !ds.FindDataElement( privtag1 ) )
    {
    assert( 0 );
    return false;
    }
  const gdcm::DataElement& values10name = ds.GetDataElement( privtag1 );
  gdcm::Element<gdcm::VR::LO,gdcm::VM::VM1> el;
  el.SetFromDataElement( values10name );
  std::cout « std::endl;
  std::cout « " <" « el.GetValue().c_str() « ">" « std::endl;

  return PrintNameValueMapping2( privtag2, ds, sqi_names, indent);
}

bool print73( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
  const gdcm::PrivateTag tseq_values73(0x7fe1,0x73,"GEMS_Ultrasound_MovieGroup_001");
```

```
  if( !ds10.FindDataElement( tseq_values73 ) )
    {
    std::cout « indent « "No group 73" « std::endl;
    return false;
    }
  const gdcm::DataElement& seq_values73 = ds10.GetDataElement( tseq_values73 );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values73 = seq_values73.GetValueAsSQ();

  size_t ni3 = sqi_values73->GetNumberOfItems();
  for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
    gdcm::Item &item_73 = sqi_values73->GetItem(i3);
    gdcm::DataSet &ds73 = item_73.GetNestedDataSet();
    assert( ds73.Size() == 3 );

    const gdcm::PrivateTag tseq_values74name(0x7fe1,0x74,"GEMS_Ultrasound_MovieGroup_001");
    const gdcm::PrivateTag tseq_values75(0x7fe1,0x75,"GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values74name, tseq_values75, ds73, sqi_dict, indent);
    std::cout « std::endl;
    }
  return true;
}

bool print36( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
  (void)sqi_dict;
  const gdcm::PrivateTag tseq_values36(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001");
  if( !ds10.FindDataElement( tseq_values36 ) )
    {
    std::cout « indent « "No group 36" « std::endl;
    return false;
    }
  const gdcm::DataElement& seq_values36 = ds10.GetDataElement( tseq_values36 );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values36 = seq_values36.GetValueAsSQ();

  size_t ni3 = sqi_values36->GetNumberOfItems();
  assert( ni3 >= 1 );
  for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
    gdcm::Item &item_36 = sqi_values36->GetItem(i3);
    gdcm::DataSet &ds36 = item_36.GetNestedDataSet();
    assert( ds36.Size() == 4 );

    // (7fe1,1037) UL 47  # 4,1 US MovieGroup Number of Frames
    // (7fe1,1043) OB 40\00\1c\c4\67\2f\0b\11\40         # 376,1 ?
    // (7fe1,1060) OB 4e\4e\49\4f\4e\47\46\43\2a         # 4562714,1 US MovieGroup Image Data
    //
    const gdcm::PrivateTag timagedata(0x7fe1,0x60,"GEMS_Ultrasound_MovieGroup_001");
    assert( ds36.FindDataElement( timagedata ) );
    gdcm::DataElement const & imagedata = ds36.GetDataElement( timagedata );

      const gdcm::ByteValue * bv = imagedata.GetByteValue();
  assert( bv );
      static int c = 0;
      std::stringstream ss;
      ss « "/tmp/debug";
      ss « c++;
      std::ofstream os( ss.str().c_str(), std::ios::binary );
      os.write( bv->GetPointer(), bv->GetLength() );
      os.close();

    //const gdcm::PrivateTag tseq_values85(0x7fe1,0x85,"GEMS_Ultrasound_MovieGroup_001");
    //PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict, indent);
    //std::cout « std::endl;
    }
  return true;
}
bool print83( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
  const gdcm::PrivateTag tseq_values83(0x7fe1,0x83,"GEMS_Ultrasound_MovieGroup_001");
  if( !ds10.FindDataElement( tseq_values83 ) )
    {
    std::cout « indent « "No group 83" « std::endl;
    return false;
    }
  const gdcm::DataElement& seq_values83 = ds10.GetDataElement( tseq_values83 );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values83 = seq_values83.GetValueAsSQ();

  size_t ni3 = sqi_values83->GetNumberOfItems();
  for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
```

```
    gdcm::Item &item_83 = sqi_values83->GetItem(i3);
    gdcm::DataSet &ds83 = item_83.GetNestedDataSet();
    assert( ds83.Size() == 3 );

    const gdcm::PrivateTag tseq_values84name(0x7fe1,0x84,"GEMS_Ultrasound_MovieGroup_001");
    const gdcm::PrivateTag tseq_values85(0x7fe1,0x85,"GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values84name, tseq_values85, ds83, sqi_dict, indent);
    std::cout « std::endl;
    }
  return true;
}

bool PrintNameValueMapping4( gdcm::PrivateTag const & privtag0, const gdcm::DataSet & subds, gdcm::PrivateTag
      const & privtag1, gdcm::PrivateTag const & privtag2,
gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
  (void)indent;
  if( !subds.FindDataElement( privtag0 ) )
    {
    assert( 0 );
    return false;
    }
  const gdcm::DataElement& seq_values10 = subds.GetDataElement( privtag0 );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values10 = seq_values10.GetValueAsSQ();

  size_t ni1 = sqi_values10->GetNumberOfItems();
//  assert( ni1 == 1 );
  for( size_t i1 = 1; i1 <= ni1; ++i1 )
    {
    gdcm::Item &item_10 = sqi_values10->GetItem(i1);
    gdcm::DataSet &ds10 = item_10.GetNestedDataSet();
    assert( ds10.Size() == 2 + 3 );
    // (7fe1,0010)
    // (7fe1,1012)
    // (7fe1,1018)
    // (7fe1,1020)
    // (7fe1,1083)

    PrintNameValueMapping3( privtag1, privtag2, ds10, sqi_dict, "  " );
    std::cout « std::endl;

    const gdcm::PrivateTag tseq_values20(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values20 ) )
      {
      assert( 0 );
      return false;
      }
    const gdcm::DataElement& seq_values20 = ds10.GetDataElement( tseq_values20 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values20 = seq_values20.GetValueAsSQ();

    size_t ni2 = sqi_values20->GetNumberOfItems();
    //assert( ni == 1 );
    for( size_t i2 = 1; i2 <= ni2; ++i2 )
      {
      gdcm::Item &item_20 = sqi_values20->GetItem(i2);
      gdcm::DataSet &ds20 = item_20.GetNestedDataSet();
      size_t count = ds20.Size(); (void)count;
      assert( ds20.Size() == 2 + 3 || ds20.Size() == 2 + 2 );
      // (7fe1,0010)
      // (7fe1,1024)
      // (7fe1,1026)
      // (7fe1,1036)
      // (7fe1,103a)
      // (7fe1,1083) (*)

      const gdcm::PrivateTag tseq_values20name(0x7fe1,0x24,"GEMS_Ultrasound_MovieGroup_001");
      const gdcm::PrivateTag tseq_values26(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
      PrintNameValueMapping3( tseq_values20name, tseq_values26, ds20, sqi_dict, "   ");
      std::cout « std::endl;

      print36(ds20, sqi_dict, "    ");
      print83(ds20, sqi_dict, "    ");
      }

    print83(ds10, sqi_dict, "   ");
    }
  return true;
}

int main(int argc, char *argv[])
{
```

```
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() ) return 1;

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

  if( !ds.FindDataElement( tseq ) ) return 1;
  const DataElement& seq = ds.GetDataElement( tseq );

  SmartPointer<SequenceOfItems> sqi = seq.GetValueAsSQ();
  assert( sqi->GetNumberOfItems() == 1 );

  Item &item = sqi->GetItem(1);
  DataSet &subds = item.GetNestedDataSet();

  const PrivateTag tseq_dict(0x7fe1,0x70,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_dict ) ) return 1;
  const DataElement& seq_dict = subds.GetDataElement( tseq_dict );
  SmartPointer<SequenceOfItems> sqi_dict = seq_dict.GetValueAsSQ();

  const PrivateTag tseq_values8(0x7fe1,0x8,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_values8 ) ) return 1;
  const DataElement& seq_values8 = subds.GetDataElement( tseq_values8 );
  SmartPointer<SequenceOfItems> sqi_values8 = seq_values8.GetValueAsSQ();

  const PrivateTag tseq_values8name(0x7fe1,0x2,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds.FindDataElement( tseq_values8name ) ) return 1;
  const DataElement& values8name = subds.GetDataElement( tseq_values8name );
{
  Element<VR::LO,VM::VM1> el;
  el.SetFromDataElement( values8name );
  std::cout « el.GetValue() « std::endl;
}
  size_t count = subds.Size(); (void)count;
  assert( subds.Size() == 3 + 2 + 1 || subds.Size() == 3 + 2 + 2);

//  (7fe1,0010) # 30,1 Private Creator
//  (7fe1,1002) # 8,1 US MovieGroup Value 0008 Name
//  (7fe1,1003) # 4,1 ?
//  (7fe1,1008) # 8140,1 US MovieGroup Value 0008 Sequence
//  (7fe1,1010) # 1372196,1 ?
//  (7fe1,1070) # 33684,1 US MovieGroup Dict
//  (7fe1,1073) (*)
  PrintNameValueMapping( sqi_values8, sqi_dict, " ");

  const PrivateTag tseq_values10(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq_values10name(0x7fe1,0x12,"GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq_values18(0x7fe1,0x18,"GEMS_Ultrasound_MovieGroup_001");
  PrintNameValueMapping4( tseq_values10, subds, tseq_values10name, tseq_values18, sqi_dict,"  ");

  print73( subds, sqi_dict, "  " );

#if 0
  gdcm::DataSet::ConstIterator it = subds.Begin();
  for( ; it != subds.End(); ++it )
    {
  const gdcm::DataElement &de = *it;
    std::cout « de.GetTag() « std::endl;
    }
#endif

  return 0;
}
```

14.52 DumpImageHeaderInfo.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Dump TOSHIBA MDW HEADER / Image Header Info
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <string.h>
#include <assert.h>
#include <stdint.h>

struct element
{
  std::istream & read( std::istream & is );
};

std::istream & element::read( std::istream & is )
{
  static const uint32_t ref = 0xe000fffe;
  std::ostream &os = std::cout;
  if( is.eof() )
    {
    return is;
    }
  uint32_t magic;
  if( !is.read( (char*)&magic, sizeof(magic) ) )
    {
    return is;
    }
  //os « magic « std::endl;
  assert( magic == ref ); (void)ref;

  uint32_t l;
  is.read( (char*)&l, sizeof(l) );
  //os « l « std::endl;

  char str[17];
  str[16] = 0;
  is.read( str, 16 );
  os « str « " (" « l « ")" « std::endl;
  std::vector<char> bytes;
  bytes.resize( l - 16 );
  if( !bytes.empty() )
    {
    is.read( bytes.data(), l - 16 );
    }
  //os « "pos:" « is.tellg() « std::endl;

  if( strcmp(str, "TUSREMEASUREMENT" ) == 0 )
    {
    const char *p = bytes.data();
    uint32_t val;
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    memcpy( (char*)&val, p, sizeof(val) );
```

```
    os « " " « val « std::endl;
    p += sizeof(val);
#if 0
    float f;
    memcpy( (char*)&f, p, sizeof(f) );
    os « " " « f « std::endl;
    p += sizeof(f);
#else
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
#endif
    memcpy( (char*)&val, p, sizeof(val) );
    os « " " « val « std::endl;
    p += sizeof(val);
    char str2[17];
    memcpy( str2, p, 16 );
    str2[16] = 0;
    os « " " « str2 « std::endl;
    }

#if 0
  std::ofstream out( str, std::ios::binary );
  out.write( (char*)&magic, sizeof( magic ) );
  out.write( (char*)&l, sizeof( l ) );
  out.write( str, 16 );
  out.write( &bytes[0], bytes.size() );
#endif
  return is;
}

static bool DumpImageHeaderInfo( std::istream & is, size_t reflen )
{
  // TUSNONIMAGESTAM (5176)
  // TUSREMEASUREMEN (1352)
  // TUSBSINGLELAYOU (16)
  // TUSCLIPPARAMETE (104)

  element el;
  while( el.read( is ) )
    {
    }
  //size_t pos = is.tellg();
  //assert( pos == reflen );
  (void)reflen;

  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  const gdcm::PrivateTag timageheaderinfo(0x0029,0x10,"TOSHIBA MDW HEADER");
  if( !ds.FindDataElement( timageheaderinfo) ) return 1;
  const gdcm::DataElement& imageheaderinfo = ds.GetDataElement( timageheaderinfo );
  if ( imageheaderinfo.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = imageheaderinfo.GetByteValue();

  std::istringstream is;
  std::string dup( bv->GetPointer(), bv->GetLength() );
  is.str( dup );
  bool b = DumpImageHeaderInfo( is, bv->GetLength() );
  if( !b ) return 1;

#if 0
  const float d1 = 0.0041666668839752674; // 89 88 88 3B // 0x44c
  //const float d1 = 0.053231674455417881;
  const float d2 = 0.10828025639057159; // 0A C2 DD 3D // 0x1ac
  //const float d1 = 0.17869562069272813;
  //const unsigned int d2 = 4294967280;
  const float d3 = 0.10828025639057159; // 0A C2 DD 3D // 0x15c
```

```
  const int32_t d4 = 134;
  const uint32_t d5 = 1153476;
  std::ofstream t("/tmp/debug", std::ios::binary );
  //t.write( (char*)&d0, sizeof( d0 ) );
  t.write( (char*)&d1, sizeof( d1 ) );
  t.write( (char*)&d2, sizeof( d2 ) );
  t.write( (char*)&d3, sizeof( d3 ) );
  t.write( (char*)&d4, sizeof( d4 ) );
  t.write( (char*)&d5, sizeof( d5 ) );
  t.close();
#endif

  return 0;
}
```

14.53 DumpPhilipsECHO.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmDeflateStream.h"
#include "gdcm_zlib.h"

/*
 * This example extract the ZLIB compressed US image from a Philips private tag
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Usage:
 *
 * $ DumpPhilipsECHO private_us.dcm raw_us_img.raw
 * $ gdcmimg --sop-class-uid 1.2.840.10008.5.1.4.1.1.3.1 --size 608,427,88 raw_us_img.raw raw_us_img.dcm
 */

// header:
struct hframe
{
  uint32_t val0; // 800 increment ?
  uint16_t val1[2];
  uint16_t val2[2];
  uint32_t imgsize;

  bool operator==(const hframe &h) const
    {
    return val0 == h.val0 &&
      val1[0] == h.val1[0] &&
      val1[1] == h.val1[1] &&
      val2[0] == h.val2[0] &&
      val2[1] == h.val2[1] &&
      imgsize == h.imgsize;
    }

};

static bool ProcessDeflate( const char *outfilename, const int nslices, const
  int buf_size, const char *buf, const std::streampos len,
  const char *crcbuf, const size_t crclen )
{
```

```
  std::vector< hframe > crcheaders;
  crcheaders.reserve( nslices );
    {
    std::istringstream is;
    is.str( std::string( crcbuf, crclen ) );
    hframe header;
    for( int r = 0; r < nslices; ++r )
      {
      is.read( (char*)&header, sizeof( header ));
#if 0
      std::cout « header.val0
         « " " « header.val1[0]
         « " " « header.val1[1]
         « " " « header.val2[0]
         « " " « header.val2[1]
         « " " « header.imgsize « std::endl;
#endif
      crcheaders.push_back( header );
      }
    }

  std::istringstream is;
  is.str( std::string( buf, (size_t)len ) );

  std::streamoff totalsize;
  is.read( (char*)&totalsize, sizeof( totalsize ));
  assert( totalsize == len );

  uint32_t nframes;
  is.read( (char*)&nframes, sizeof( nframes ));
  assert( nframes == (uint32_t)nslices );

  std::vector< std::streamoff > offsets;
  offsets.reserve( nframes );
  for( uint32_t frame = 0; frame < nframes ; ++frame )
    {
    uint32_t offset;
    is.read( (char*)&offset, sizeof( offset ));
    offsets.push_back( offset );
    }

  std::vector<char> outbuf;

  const int size[2] = { 608, 427 }; // FIXME: where does it comes from ?
  std::stringstream ss;
  ss « outfilename;
  ss « '_';
  //ss « crcheaders[0].imgsize; // FIXME: Assume all header are identical !
  ss « size[0];
  ss « '_';
  ss « size[1];
  ss « '_';
  ss « nframes;
  ss « ".raw";
  std::ofstream os( ss.str().c_str(), std::ios::binary );

  assert( buf_size >= size[0] * size[1] );
  outbuf.resize( buf_size );

  hframe header;
  //uint32_t prev = 0;
  for( unsigned int r = 0; r < nframes; ++r )
    {
    is.read( (char*)&header, sizeof( header ));

    assert( header == crcheaders[r] );
    assert( header.val1[0] == 2000 );
    assert( header.val1[1] == 3 );
    assert( header.val2[0] == 1 );
    assert( header.val2[1] == 1280 );

    uLongf destLen = buf_size; // >= 608,427
    Bytef *dest = (Bytef*)outbuf.data();
    assert( is.tellg() == offsets[r] + 16 );
    const Bytef *source = (const Bytef*)buf + offsets[r] + 16;
    uLong sourceLen;
    if( r + 1 == nframes )
      sourceLen = (uLong)totalsize - (uLong)offsets[r] - 16;
    else
      sourceLen = (uLong)offsets[r+1] - (uLong)offsets[r] - 16;
    // FIXME: in-memory decompression:
```

```
    int ret = uncompress (dest, &destLen, source, sourceLen);
    assert( ret == Z_OK ); (void)ret;
    assert( destLen >= (uLongf)size[0] * size[1] ); // 16bytes padding ?
    assert( header.imgsize == (uint32_t)size[0] * size[1] );
    //os.write( &outbuf[0], outbuf.size() );
    os.write( outbuf.data(), size[0] * size[1] );

    // skip data:
    is.seekg( sourceLen, std::ios::cur );
    }
  os.close();
  assert( is.tellg() == totalsize );

  return true;
}

static bool ProcessNone( const char *outfilename, const int nslices, const
  int buf_size, const char *buf, const std::streampos len,
  const char *crcbuf, const size_t crclen )
{
  std::vector< hframe > crcheaders;
  crcheaders.reserve( nslices );
    {
    std::istringstream is;
    is.str( std::string( crcbuf, crclen ) );
    hframe header;
    for( int r = 0; r < nslices; ++r )
      {
      is.read( (char*)&header, sizeof( header ));
#if 0
      std::cout « header.val0
         « " " « header.val1[0]
         « " " « header.val1[1]
         « " " « header.val2[0]
         « " " « header.val2[1]
         « " " « header.imgsize « std::endl;
#endif
      crcheaders.push_back( header );
      }
    }

  std::istringstream is;
  is.str( std::string( buf, (size_t)len ) );

  std::streampos totalsize;
  is.read( (char*)&totalsize, sizeof( totalsize ));
  assert( totalsize == len );

  uint32_t nframes;
  is.read( (char*)&nframes, sizeof( nframes ));
  assert( nframes == (uint32_t)nslices );

  std::vector< uint32_t > offsets;
  offsets.reserve( nframes );
  for( uint32_t frame = 0; frame < nframes ; ++frame )
    {
    uint32_t offset;
    is.read( (char*)&offset, sizeof( offset ));
    offsets.push_back( offset );
    //std::cout « offset « std::endl;
    }

  std::vector<char> outbuf;
  // No idea how to present the data, I'll just append everything, and present it as 2D
  std::stringstream ss;
  ss « outfilename;
  ss « '_';
  ss « crcheaders[0].imgsize; // FIXME: Assume all header are identical !
  ss « '_';
  ss « nframes;
  ss « ".raw";
  std::ofstream os( ss.str().c_str(), std::ios::binary );
  outbuf.resize( buf_size ); // overallocated + 16
  char *buffer = outbuf.data();

  hframe header;
  for( unsigned int r = 0; r < nframes; ++r )
    {
    is.read( (char*)&header, sizeof( header ));
#if 0
      std::cout « header.val0
```

```
        « " " « header.val1[0]
        « " " « header.val1[1]
        « " " « header.val2[0]
        « " " « header.val2[1]
        « " " « header.imgsize « std::endl;
#endif
    assert( header == crcheaders[r] );

    is.read( buffer, buf_size - 16 );
    os.write( buffer, header.imgsize );
    }
  assert( is.tellg() == totalsize );
  os.close();

  return true;
}

#ifndef NDEBUG
static const char * const UDM_USD_DATATYPE_STRINGS[] = {
  "UDM_USD_DATATYPE_DIN_2D_ECHO",
  "UDM_USD_DATATYPE_DIN_2D_ECHO_CONTRAST",
  "UDM_USD_DATATYPE_DIN_DOPPLER_CW",
  "UDM_USD_DATATYPE_DIN_DOPPLER_PW",
  "UDM_USD_DATATYPE_DIN_DOPPLER_PW_TDI",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_FLOW",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_PMI",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_CPA",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_TDI",
  "UDM_USD_DATATYPE_DIN_MMODE_ECHO",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR_TDI",
  "UDM_USD_DATATYPE_DIN_PARAM_BLOCK",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_VELOCITY",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_POWER",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_VARIANCE",
  "UDM_USD_DATATYPE_DIN_DOPPLER_AUDIO",
  "UDM_USD_DATATYPE_DIN_DOPPLER_HIGHQ",
  "UDM_USD_DATATYPE_DIN_PHYSIO",
  "UDM_USD_DATATYPE_DIN_2D_COLOR_STRAIN",
  "UDM_USD_DATATYPE_DIN_COMPOSITE_RGB",
  "UDM_USD_DATATYPE_DIN_XFOV_REALTIME_GRAPHICS",
  "UDM_USD_DATATYPE_DIN_XFOV_MOSAIC",
  "UDM_USD_DATATYPE_DIN_COMPOSITE_R",
  "UDM_USD_DATATYPE_DIN_COMPOSITE_G",
  "UDM_USD_DATATYPE_DIN_COMPOSITE_B",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VELOCITY",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR_POWER",
  "UDM_USD_DATATYPE_DIN_MMODE_COLOR_VARIANCE",
  "UDM_USD_DATATYPE_DIN_2D_ELASTO",
};

static inline bool is_valid( const char * datatype_str )
{
  static const int n = sizeof( UDM_USD_DATATYPE_STRINGS ) / sizeof( *UDM_USD_DATATYPE_STRINGS );
  bool found = false;
  if( datatype_str )
    {
    for( int i = 0; !found && i < n; ++i )
      {
      found = strcmp( datatype_str, UDM_USD_DATATYPE_STRINGS[i] ) == 0;
      }
    }
  return found;
}
#endif

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() ) return 1;

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds1 = file.GetDataSet();

  const PrivateTag tseq1(0x200d,0x3cf8,"Philips US Imaging DD 045");
  if( !ds1.FindDataElement( tseq1 ) ) return 1;
```

```
const DataElement& seq1 = ds1.GetDataElement( tseq1 );

SmartPointer<SequenceOfItems> sqi1 = seq1.GetValueAsSQ();
assert( sqi1->GetNumberOfItems() >= 1 );

const size_t nitems = sqi1->GetNumberOfItems();
for( size_t item = 1; item < nitems; ++item )
  {
  Item &item1 = sqi1->GetItem(item);
  DataSet &ds2 = item1.GetNestedDataSet();

  // (200d,300d)  LO  28  UDM_USD_DATATYPE_DIN_2D_ECHO
  const PrivateTag tdatatype(0x200d,0x300d,"Philips US Imaging DD 033");
  if( !ds2.FindDataElement( tdatatype ) ) return 1;
  const DataElement& datatype = ds2.GetDataElement( tdatatype );
  const ByteValue *bvdatatype = datatype.GetByteValue();
  if( !bvdatatype ) return 1;

  const PrivateTag tseq2(0x200d,0x3cf1,"Philips US Imaging DD 045");
  if( !ds2.FindDataElement( tseq2 ) ) return 1;
  const DataElement& seq2 = ds2.GetDataElement( tseq2 );

  SmartPointer<SequenceOfItems> sqi2 = seq2.GetValueAsSQ();
  assert( sqi2->GetNumberOfItems() >= 1 );

  // FIXME: what if not in first Item ?
  assert( sqi2->GetNumberOfItems() == 1 );
  Item &item2 = sqi2->GetItem(1);
  DataSet &ds3 = item2.GetNestedDataSet();

  const PrivateTag tzlib(0x200d,0x3cfa,"Philips US Imaging DD 045");
  if( !ds3.FindDataElement( tzlib ) ) return 1;
  const DataElement& zlib = ds3.GetDataElement( tzlib );

  const ByteValue *bv = zlib.GetByteValue();
  if( !bv ) return 1;
  if( bv->GetLength() != 4 ) return 1;

  // (200d,3010)  IS  2  88
  const PrivateTag tnslices(0x200d,0x3010,"Philips US Imaging DD 033");
  if( !ds3.FindDataElement( tnslices ) ) return 1;
  const DataElement& nslices = ds3.GetDataElement( tnslices );
  Element<VR::IS,VM::VM1> elnslices;
  elnslices.SetFromDataElement( nslices );
  const int nslicesref = elnslices.GetValue();
  assert( nslicesref >= 0 );
  // (200d,3011)  IS  6  259648
  const PrivateTag tzalloc(0x200d,0x3011,"Philips US Imaging DD 033");
  if( !ds3.FindDataElement( tzalloc ) ) return 1;
  const DataElement& zalloc = ds3.GetDataElement( tzalloc );
  Element<VR::IS,VM::VM1> elzalloc;
  elzalloc.SetFromDataElement( zalloc );
  const int zallocref = elzalloc.GetValue();
  assert( zallocref >= 0 );
  // (200d,3021)  IS  2  0
  const PrivateTag tzero(0x200d,0x3021,"Philips US Imaging DD 033");
  if( !ds3.FindDataElement( tzero ) ) return 1;
  const DataElement& zero = ds3.GetDataElement( tzero );
  Element<VR::IS,VM::VM1> elzero;
  elzero.SetFromDataElement( zero );
  const int zerocref = elzero.GetValue();
  assert( zerocref == 0 ); (void)zerocref;

  // (200d,3cf3) OB
  const PrivateTag tdeflate(0x200d,0x3cf3,"Philips US Imaging DD 045");
  if( !ds3.FindDataElement( tdeflate) ) return 1;
  const DataElement& deflate = ds3.GetDataElement( tdeflate );
  const ByteValue *bv2 = deflate.GetByteValue();

  // (200d,3cfb) OB
  const PrivateTag tcrc(0x200d,0x3cfb,"Philips US Imaging DD 045");
  if( !ds3.FindDataElement( tcrc ) ) return 1;
  const DataElement& crc = ds3.GetDataElement( tcrc );
  const ByteValue *bv3 = crc.GetByteValue();

  std::string outfile = std::string( bvdatatype->GetPointer(), bvdatatype->GetLength() );
  outfile = LOComp::Trim( outfile.c_str() );
  const char *outfilename = outfile.c_str();
  assert( is_valid(outfilename) );
  if( bv2 )
    {
```

```
      assert( bv3 );
      assert( zallocref > 0 );
      assert( nslicesref > 0 );
      std::cout « ds2 « std::endl;

      if( strncmp(bv->GetPointer(), "ZLib", 4) == 0 )
        {
        if( !ProcessDeflate( outfilename, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
          {
          return 1;
          }
        }
      else if( strncmp(bv->GetPointer(), "None", 4) == 0 )
        {
        if( !ProcessNone( outfilename, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->GetLength() ) )
          {
          return 1;
          }
        }
      else
        {
        std::string str( bv->GetPointer(), bv->GetLength() );
        std::cerr « "Unhandled: " « str « std::endl;
        return 1;
        }
      }
    }

  return 0;
}
```

14.54 DumpSiemensBase64.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * https://groups.google.com/forum/#!msg/comp.protocols.dicom/2kZ2lLP8EcM/WzjFrtjnAgAJ
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"
#include "gdcmCSAHeader.h"
#include "gdcmBase64.h"
#include "gdcmExplicitDataElement.h"
#include "gdcmSwapper.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
  {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
```

```
  }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  gdcm::CSAHeader csa;
  const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
  if( !ds.FindDataElement( t1 ) ) return 1;
  csa.LoadFromDataElement( ds.GetDataElement( t1 ) );

  //const char name[] = "MRDiffusion";
  const char name[] = "MR_ASL";
  if( csa.FindCSAElementByName(name) )
  {
    const gdcm::CSAElement & el = csa.GetCSAElementByName(name);
    const gdcm::ByteValue* bv = el.GetByteValue();
    std::string str( bv->GetPointer(), bv->GetLength() );
    str.erase(std::remove(str.begin(), str.end(), '\n'), str.end());
    size_t dl = gdcm::Base64::GetDecodeLength( str.c_str(), str.size() );
    std::vector<char> buf;
    buf.resize( dl );
    size_t dl2 = gdcm::Base64::Decode( buf.data(), buf.size(), str.c_str(), str.size() );
    (void)dl2;
    std::stringstream ss;
    ss.str( std::string(buf.data(), buf.size()) );
    gdcm::File file;
    gdcm::DataSet &ds2 = file.GetDataSet();
    gdcm::DataElement xde;
    try
    {
      while( xde.Read<gdcm::ExplicitDataElement,gdcm::SwapperNoOp>( ss ) )
      {
        ds2.Insert( xde );
      }
      assert( ss.eof() );
    }
    catch(std::exception &)
    {
      return 1;
    }
    gdcm::Printer p;
    p.SetFile( file );
    p.Print(std::cout);

  }

  return 0;
}
```

14.55 DumpToSQLITE3.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Ref:
 * http://massmail.spl.harvard.edu/public-archives/slicer-devel/2010/004408.html
 *
 * Implementation details:
 * http://www.sqlite.org/c3ref/bind_blob.html
 * http://www.adp-gmbh.ch/sqlite/bind_insert.html
 */
#include "gdcmScanner.h"
#include "gdcmDirectory.h"
#include "gdcmTag.h"
```

```
#include "gdcmTrace.h"

#include "sqlite3.h"

#include <stdio.h>
#include <time.h>

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  time_t time_start = time(nullptr);

  gdcm::Trace::SetDebug( false );
  gdcm::Trace::SetWarning( false );
  const char *inputdirectory = argv[1];

  gdcm::Directory d;
  unsigned int nfiles = d.Load( inputdirectory, true);

  gdcm::Scanner s;
  using gdcm::Tag;
  s.AddTag( Tag(0x20,0xd) ); // Study Instance UID
  s.AddTag( Tag(0x20,0xe) ); // Series Instance UID

  bool b0 = s.Scan( d.GetFilenames() );
  if( !b0 ) return 1;
  time_t time_scanner = time(nullptr);

  std::cout « "Finished loading data from : " « nfiles « " files" « std::endl;

//  MappingType const &mappings = s.GetMappings();


  sqlite3* db;
  sqlite3_open("./dicom.db", &db);

  if(db == nullptr)
    {
    std::cerr « "Could not open database." « std::endl;
    return 1;
    }

  const char sql_stmt[] = "create table browser (seriesuid, studyuid)";
  int   ret;

  char *errmsg;
  ret = sqlite3_exec(db, sql_stmt, nullptr, nullptr, &errmsg);

  if(ret != SQLITE_OK)
    {
    printf("Error in statement: %s [%s].\n", sql_stmt, errmsg);
    return 1;
    }
  using gdcm::Directory;
  using gdcm::Scanner;
  const Directory::FilenamesType& files = d.GetFilenames();
  Directory::FilenamesType::const_iterator file = files.begin();

  sqlite3_stmt *stmt;
  if ( sqlite3_prepare(
      db,
      "insert into browser values (?,?)",  // stmt
      -1, // If than zero, then stmt is read up to the first nul terminator
      &stmt,
      nullptr  // Pointer to unused portion of stmt
  )
    != SQLITE_OK)
    {
    printf("\nCould not prepare statement.");
    return 1;
    }
  //printf("\nThe statement has %d wildcards\n", sqlite3_bind_parameter_count(stmt));
  for(; file != files.end(); ++file)
    {
    const char *filename = file->c_str();
    bool b = s.IsKey(filename);
    if( b )
      {
```

```
      const Scanner::TagToValue &mapping = s.GetMapping(filename);
      Scanner::TagToValue::const_iterator it = mapping.begin();

      sqlite3_reset(stmt);

      for( int index = 1; it != mapping.end(); ++it, ++index)
        {
        //const Tag & tag = it->first;
        const char *value = it->second;

        if (sqlite3_bind_text (
            stmt,
            index,  // Index of wildcard
            value,
            (int)strlen(value),  // length of text
            SQLITE_STATIC // SQLite assumes that the information is in static
        )
          != SQLITE_OK)
          {
          printf("\nCould not bind int.\n");
          return 1;
          }
        }
      if (sqlite3_step(stmt) != SQLITE_DONE)
        {
        printf("\nCould not step (execute) stmt.\n");
        return 1;
        }
      }
    }

  sqlite3_close(db);

  time_t time_sqlite = time(nullptr);

  std::cout « "Time to scan DICOM files: " « (time_scanner - time_start) « std::endl;
  std::cout « "Time to build SQLITE3: " « (time_sqlite - time_scanner) « std::endl;

  return 0;
}
```

14.56 DumpToshibaDTI.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * https://groups.google.com/d/msg/comp.protocols.dicom/7IaIkT0ZG5U/k7LPu81VvAMJ
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

static bool DumpToshibaDTI( const char * input, size_t len )
{
  static int i = 0;
  ++i;
  if( len % 2 ) return false;

  std::vector<char> copy( input, input + len );
```

```
  std::reverse( copy.begin(), copy.end() );

#if 0
  std::ostringstream f;
  f « "debug" « i;
  std::ofstream of( f.str().c_str(), std::ios::binary );
  of.write( &copy[0], copy.size() );
  of.close();
#else

  std::istringstream is;
  std::string dup( copy.data(), copy.size() );
  is.str( dup );

  gdcm::File file;
  gdcm::FileMetaInformation & fmi = file.GetHeader();
  fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::DataSet & ds = file.GetDataSet();
  ds.Read<gdcm::ExplicitDataElement,gdcm::SwapperNoOp>( is );

  //gdcm::DictPrinter p;
  gdcm::Printer p;
  p.SetFile( file );
  p.SetColor( true );
  p.Print( std::cout );
#endif

  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  // (0029,0010) ?? (LO) [PMTF INFORMATION DATA ]                      # 22,1 Private Creator
  // (0029,1001) ?? (SQ) (Sequence with undefined length)              # u/l,1 ?

  const gdcm::PrivateTag tpmtf(0x0029,0x1,"PMTF INFORMATION DATA");
  if( !ds.FindDataElement( tpmtf) ) return 1;
  const gdcm::DataElement& pmtf = ds.GetDataElement( tpmtf );
  if ( pmtf.IsEmpty() ) return 1;
  gdcm::SmartPointer<gdcm::SequenceOfItems> seq = pmtf.GetValueAsSQ();
  if ( !seq || !seq->GetNumberOfItems() ) return 1;

  size_t n = seq->GetNumberOfItems();
  for( size_t i = 1; i <= n; ++i )
    {
    gdcm::Item &item = seq->GetItem(i);
    gdcm::DataSet &subds = item.GetNestedDataSet();
    // (0029,0010) ?? (LO) [PMTF INFORMATION DATA ]                  # 22,1 Private Creator
    // (0029,1090) ?? (OB) 00\05\00\13\00\12\00\22\                  # 202,1 ?
    const gdcm::PrivateTag tseq(0x0029,0x90,"PMTF INFORMATION DATA");

    if( subds.FindDataElement( tseq ) )
      {
      const gdcm::DataElement & de = subds.GetDataElement( tseq );
      const gdcm::ByteValue * bv = de.GetByteValue();
      if( !bv ) return 1;

      bool b = DumpToshibaDTI( bv->GetPointer(), bv->GetLength() );
      if( !b ) return 1;
      }

    }

  return 0;
}
```

14.57 DumpToshibaDTI2.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 *
      https://gazelle.ihe.net/EVSClient/dicomResult.seam;jsessionid=x+Rf9Zs+ip49P+jC3L8SLZb8?&oid=1.3.6.1.4.1.12559.11.1.2.1.4.162
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrinter.h"
#include "gdcmDictPrinter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <assert.h>

static bool DumpToshibaDTI2( const char * input, size_t len )
{
  static int i = 0;
  ++i;
  if( len % 2 ) return false;

  std::vector<char> copy( input, input + len );
  std::reverse( copy.begin(), copy.end() );

#if 0
  std::ostringstream f;
  f « "debug" « i;
  std::ofstream of( f.str().c_str(), std::ios::binary );
  of.write( &copy[0], copy.size() );
  of.close();
#else

  std::istringstream is;
  std::string dup( copy.data(), copy.size() );
  is.str( dup );

  gdcm::File file;
  gdcm::FileMetaInformation & fmi = file.GetHeader();
  fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  gdcm::DataSet & ds = file.GetDataSet();
  ds.Read<gdcm::ExplicitDataElement,gdcm::SwapperNoOp>( is );

  //gdcm::DictPrinter p;
  gdcm::Printer p;
  p.SetFile( file );
  p.SetColor( true );
  p.Print( std::cout );
#endif

  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }
```

```
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  /*
  (0029,1001) SQ (Sequence with explicit length #=6)      # 18746, 1 Unknown Tag & Data
    (fffe,e000) na (Item with explicit length #=2)          # 206, 1 Item
      (0029,0010) LO [TOSHIBA_MEC_MR3]                        #  16, 1 PrivateCreator
      (0029,1090) OB 00\07\00\06\00\05\00\04\00\03\00\02\00\0c\00\01\00\00\00\00\00\12... # 170, 1 Unknown Tag &
      Data
    (fffe,e00d) na (ItemDelimitationItem for re-encoding)   #   0, 0 ItemDelimitationItem
    (fffe,e000) na (Item with explicit length #=2)          # 866, 1 Item
      (0029,0010) LO [TOSHIBA_MEC_MR3]                        #  16, 1 PrivateCreator
      (0029,1090) OB 45\4e\49\50\53\4c\20\52\41\5c\45\4e\49\50\53\4c\54\5c\52\45\53\55... # 830, 1 Unknown Tag &
      Data
  [...]
  (0029,1002) SQ (Sequence with explicit length #=1)      # 120, 1 Unknown Tag & Data
    (fffe,e000) na (Item with explicit length #=2)          # 112, 1 Item
      (0029,0010) LO [TOSHIBA_MEC_MR3]                        #  16, 1 PrivateCreator
      (0029,1090) OB 00\10\00\02\53\55\10\80\70\0d\30\31\5e\33\52\4d\5f\43\45\4d\5f\41... #  76, 1 Unknown Tag &
      Data
    (fffe,e00d) na (ItemDelimitationItem for re-encoding)   #   0, 0 ItemDelimitationItem
  */

  const gdcm::PrivateTag tmecmr3(0x0029,0x1,"TOSHIBA_MEC_MR3");
  if( !ds.FindDataElement( tmecmr3) ) return 1;
  const gdcm::DataElement& mecmr3 = ds.GetDataElement( tmecmr3 );
  if ( mecmr3.IsEmpty() ) return 1;
  gdcm::SmartPointer<gdcm::SequenceOfItems> seq = mecmr3.GetValueAsSQ();
  if ( !seq || !seq->GetNumberOfItems() ) return 1;

  size_t n = seq->GetNumberOfItems();
  for( size_t i = 1; i <= n; ++i )
    {
    gdcm::Item &item = seq->GetItem(i);
    gdcm::DataSet &subds = item.GetNestedDataSet();
    const gdcm::PrivateTag tseq(0x0029,0x90,"TOSHIBA_MEC_MR3");

    if( subds.FindDataElement( tseq ) )
      {
      const gdcm::DataElement & de = subds.GetDataElement( tseq );
      const gdcm::ByteValue * bv = de.GetByteValue();
      if( !bv ) return 1;

      bool b = DumpToshibaDTI2( bv->GetPointer(), bv->GetLength() );
      if( !b ) return 1;
      }

    }

  return 0;
}
```

14.58 DumpVisusChange.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmDirectory.h"
#include "gdcmStringFilter.h"

#include <vector>
#include <algorithm>

/*
 */
static bool process( std::vector<gdcm::DataElement> & ms, const char * filename)
```

```
{
  using namespace gdcm;
  Tag pd(0x7fe0,0x0000);
  std::set<gdcm::Tag> skiptags;
  skiptags.insert( pd );

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.ReadUpToTag( pd, skiptags ) )
  {
    std::cerr « "Failure to read: " « filename « std::endl;
    return false;
  }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds1 = file.GetDataSet();

  const gdcm::PrivateTag tseq1(0x5533,0x33,"Visus Change");
  if( !ds1.FindDataElement( tseq1 ) ) return true;
  const gdcm::DataElement& seq1 = ds1.GetDataElement( tseq1 );

  SmartPointer<SequenceOfItems> sqi1 = seq1.GetValueAsSQ();

  const size_t nitems = sqi1->GetNumberOfItems();
  for( size_t item = 1; item < nitems; ++item )
    {
    Item &item1 = sqi1->GetItem(item);
    DataSet &ds2 = item1.GetNestedDataSet();
    for(DataSet::ConstIterator it = ds2.Begin(); it != ds2.End(); ++it )
      {
      DataElement const & de = *it;
      // cannot simply use std::set here, see there is a discrepancy in between
      // operator== and operator<.
      // So only use operator== here:
      std::vector<DataElement>::iterator vit = std::find(ms.begin(), ms.end(), de);
      if( vit == ms.end() )
        ms.push_back(de);
      }
    }
  return true;
}

int main(int argc, char *argv[])
{
  bool usefastpath = true;

  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Directory::FilenamesType filenames;
  if( !gdcm::System::FileExists(filename) )
    {
    std::cerr « "Could not find file: " « filename « std::endl;
    return 1;
    }

  gdcm::Directory dir;
  if( gdcm::System::FileIsDirectory(filename) )
  {
    unsigned int nfiles = dir.Load(filename, false);
    if( nfiles == 0 )
      {
      std::cerr « "Could not find files: " « filename « std::endl;
      return 1;
      }
    filenames = dir.GetFilenames();
  }
  else
  {
    filenames.push_back( filename );
  }
  gdcm::StringFilter sf;

  Tag pd(0x7fe0,0x0000);
  std::set<gdcm::Tag> skiptags;
  skiptags.insert( pd );

  gdcm::Reader reader;
  reader.SetFileName( filenames[0].c_str() );
  if( !reader.ReadUpToTag( pd, skiptags ) )
  {
```

```
    std::cerr « "Could not read file: " « filename « std::endl;
    return 1;
  }
  gdcm::File &file = reader.GetFile();
  sf.SetFile(file);

  if( usefastpath ) {
    // Heuristic, assume if private tag cannot be found in first file, skip the directory
    gdcm::DataSet &ds1 = file.GetDataSet();

    const gdcm::PrivateTag tseq1(0x5533,0x33,"Visus Change");
    if( !ds1.FindDataElement( tseq1 ) ){
      std::cerr « "Could not find private tag in first file skipping whole directory: " « filename « std::endl;
      return 0;
    }
  }
  std::vector<DataElement> ms;
  for(gdcm::Directory::FilenamesType::const_iterator cit = filenames.begin(); cit != filenames.end(); ++cit )
  {
    if( !process(ms, cit->c_str()) ) {
      return 1;
    }
  }

  if( !ms.empty() ) {
    std::sort(ms.begin(), ms.end());
    std::cout « filename « ",\"";
    for(std::vector<DataElement>::const_iterator it = ms.begin(); it != ms.end(); ++it )
    {
      DataElement const & de = *it;
      std::string const & s = sf.ToString( de );
      std::cout « de.GetTag() « " " « s « std::endl;
    }
    std::cout « "\"" « std::endl;
  }

  return 0;
}
```

14.59 DuplicatePCDE.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
/*
  Usage:
  DuplicatePCDE gdcmData/D_CLUNIE_CT1_J2KI.dcm out.dcm

aka:
medical.nema.org/medical/dicom/DataSets/WG04/IMAGES/J2KI/CT1_J2KI

See:
  gdcmConformanceTests/CT1_J2KI_DuplicatePCDE.dcm

Original thread can be found at:

http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/82f28c4db28963af
```

```
Question:
1.
  There is no restriction for a specific Private Creator Data Element
(PCDE) to be unique within the same group, right ?
  Decoders of Private Data would have to handle the case where a PCDE
would be repeated and should NOT stop on the first instance of a
particular PCDE, right ?

  Eg. when searching for the tag associated with
(0x0029,0x0010,"SIEMENS CSA HEADER") in the following (pseudo)
dataset:

(0029,0010) LO [SIEMENS CSA HEADER]                     #  18, 1
PrivateCreator
(0029,0011) LO [SIEMENS MEDCOM HEADER]                  #  22, 1
PrivateCreator
(0029,0012) LO [SIEMENS MEDCOM HEADER2]                 #  22, 1
PrivateCreator
(0029,0013) LO [SIEMENS CSA HEADER]                     #  18, 1
PrivateCreator
(0029,1008) CS [IMAGE NUM 4]                            #  12, 1
CSAImageHeaderType
(0029,1009) LO [20050723]                               #   8, 1
CSAImageHeaderVersion
(0029,1010) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
(0029,1018) CS [MR]                                     #   2, 1
CSASeriesHeaderType
(0029,1019) LO [20050723]                               #   8, 1
CSASeriesHeaderVersion
(0029,1020) OB 53\56\31\30\04\03\02\01\2c\00\00\00\4d
\00\00\00\55\73\65\64\50\61... # 51520, 1 CSASeriesHeaderInfo
(0029,1131) LO [4.0.163088300]                          #  14, 1
PMTFInformation1
(0029,1132) UL 32768                                    #   4, 1
PMTFInformation2
(0029,1133) UL 0                                        #   4, 1
PMTFInformation3
(0029,1134) CS [DB TO DICOM]                            #  12, 1
PMTFInformation4
(0029,1260) ?? 63\6f\6d\20                              #   4, 1
Unknown Tag & Data
(0029,1310) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo

  one should return two instances, correct ?

Answer:
I would say that this is covered in principle by the PS 3.5 7.1
"The Data Elements ... shall occur at most once in a Data Set"
rule, since the data element is defined by the tuple
(private creator,gggg,ee) where xxee is the element
number and xx is arbitrary and has no inherent meaning and
does not serve to disambiguate the data element.

E.g.:

(0019,0030) Private Creator ID = "Smith"
...
(0019,0032) Private Creator ID = "Smith"
...
(0019,3015) Fractal Index = "32"
...
(0019,3215) Fractal Index = "32"

would be illegal because even though they are assigned different
(completely arbitrary) blocks, with the same group, element
number and private creator, (0019,3015) and (0019,3215) are the
"same" data element.

*/

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
```

```
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  // Let's get all private element from group 0x9:
/*
(0009,0010) LO [GEMS_IDEN_01]                                    # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ]                                  # 14,1 Full fidelity
(0009,1002) SH [CT01]                                            # 4,1 Suite id
(0009,1004) SH [HiSpeed CT/i]                                    # 12,1 Product id
(0009,1027) SL 862399669                                         # 4,1 Image actual date
(0009,1030) SH (no value)                                        # 0,1 Service id
(0009,1031) SH (no value)                                        # 0,1 Mobile location number
(0009,10e6) SH [05]                                              # 2,1 Genesis Version - now
(0009,10e7) UL 973283917                                         # 4,1 Exam Record checksum
(0009,10e9) SL 862399669                                         # 4,1 Actual series data time stamp
*/
  gdcm::Tag start(0x0009,0x0);
  // Create a temporary duplicate dataset, since we cannot insert data element as we go over them (std::set
  // would reorganize itself as we go over it ...)
  gdcm::DataSet dup;
  gdcm::Tag new_private(0x0009,0x0);
  while (start.GetGroup() == 0x9 )
    {
    const gdcm::DataElement& de = ds.FindNextDataElement(start);
    const gdcm::Tag &t = de.GetTag();
    if( t.IsPrivateCreator() )
      {
      std::cout « t « std::endl;
      // Ok let's duplicate into the next available attribute:
      gdcm::DataElement duplicate = de;
      duplicate.GetTag().SetElement( (uint16_t)(t.GetElement() + 1) );
      dup.Insert( duplicate );
      new_private = duplicate.GetTag();
      }
    else if( t.IsPrivate() && !t.IsPrivateCreator() )
      {
      //std::cout « de « std::endl;
      std::string owner = ds.GetPrivateCreator( de.GetTag() );
      //std::cout « owner « std::endl;
      gdcm::DataElement duplicate = de;
      duplicate.GetTag().SetPrivateCreator( new_private );
      if( const gdcm::ByteValue *bv = duplicate.GetByteValue() )
        {
        // Warning: when doing : duplicate = de, only the pointer to the ByteValue is passed
        // (to avoid large memory duplicate). We need to explicitly duplicate the bytevalue ourselves:
        gdcm::ByteValue *dupbv = new gdcm::ByteValue( bv->GetPointer(),
          bv->GetLength() );
        // Let's recognize the duplicated ASCII-type elements:
        if( duplicate.GetVR() & gdcm::VR::VRASCII )
          dupbv->Fill( 'X' );
        duplicate.SetValue( *dupbv );
        }
      dup.Insert( duplicate );
      }
    start = t;
    // move to next possible 'public' element
    start.SetElement( (uint16_t)(start.GetElement() + 1) );
    }

  gdcm::DataSet::ConstIterator it = dup.Begin();
  for( ; it != dup.End(); ++it )
    {
    ds.Insert( *it );
    }

  gdcm::Writer w;
  w.SetFile( file );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
```

```
}
```

14.60 ELSCINT1WaveToText.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"

/*
 * This example shows how to read a Wave Information tag from ELSCINT1
 * The wave information is stored in Tag (01e1,18,ELSCINT1) hidden in a
 * Secondary Capture Image Storage (usually a 'N' Symbol is shown)
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Gauthier Bouilhol
 */

template <typename T>
bool dumpargs(std::ostream & os, T c1, T c2, T c3, T c4, T c5, T c6, T c7, T c8)
{
  static const char sep = '\t';
  os « c1 « sep « c2 « sep « c3 « sep « c4 « sep « c5 « sep « c6 « sep « c7 « sep « c8;
  os « std::endl;
  return true;
}

bool wave2stream( std::ostream &text_file, const char *in, size_t len )
{
  const short * buffer = (const short*)in;
  size_t length = len / sizeof( short );
  text_file « "COMPLETE_WAVE" « '\t' « "MASK"       « '\t' « "AQUISITION_PROFIL" « '\t' « "END-INHALE" « '\t' «
      "END-EXHALE" « '\t' « "AQUISITION_WAVE" « '\t' « "WAVE_STATISTICS" « '\t' « "MASK"        « std::endl;
  for (size_t i=0;i<length-76;i+=2)
    {
    if ( i < 74 )
      {
      if (buffer[i+75] == 0)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                    « '\t' « "  "         «
      '\t' « "  "         « '\t' « "  "              « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == 16384)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                    « '\t' « buffer[i+74] «
      '\t' « "  "         « '\t' « "  "              « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == 256)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                    « '\t' « "  "         «
      '\t' « buffer[i+74] « '\t' « "  "              « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == -32768)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                    « '\t' « "  "         «
      '\t' « "  "         « '\t' « buffer[i+74]      « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
      if (buffer[i+75] == -16384)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                    « '\t' « buffer[i+74] «
      '\t' « "  "         « '\t' « buffer[i+74]      « '\t' « buffer[i]         « '\t' « buffer[i+1] «
      std::endl;
```

```
      if (buffer[i+75] == -32512)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                    « '\t' « "  "         «
      '\t' « buffer[i+74] « '\t' « buffer[i+74]      « '\t' « buffer[i]          « '\t' « buffer[i+1] «
      std::endl;
      }
    else
      {
      if (buffer[i+75] == 0)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                    « '\t' « "  "         «
      '\t' « "  "         « '\t' « "  "              « '\t' « "  "              « '\t' « "  "        «
      std::endl;
      if (buffer[i+75] == 16384)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                    « '\t' « buffer[i+74] «
      '\t' « "  "         « '\t' « "  "              « '\t' « "  "              « '\t' « "  "        «
      std::endl;
      if (buffer[i+75] == 256)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 0                    « '\t' « "  "         «
      '\t' « buffer[i+74] « '\t' « "  "              « '\t' « "  "              « '\t' « "  "        «
      std::endl;
      if (buffer[i+75] == -32768)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                    « '\t' « "  "         «
      '\t' « "  "         « '\t' « buffer[i+74]      « '\t' « "  "              « '\t' « "  "        «
      std::endl;
      if (buffer[i+75] == -16384)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                    « '\t' « buffer[i+74] «
      '\t' « "  "         « '\t' « buffer[i+74]      « '\t' « "  "              « '\t' « "  "        «
      std::endl;
      if (buffer[i+75] == -32512)
        text_file « buffer[i+74]    « '\t' « buffer[i+75] « '\t' « 1                    « '\t' « "  "         «
      '\t' « buffer[i+74] « '\t' « buffer[i+74]      « '\t' « "  "              « '\t' « "  "        «
      std::endl;
      }
    }

  return true;
}

int main(int argc, char *argv [])
{
  if( argc < 3 ) return 1;
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  const gdcm::PrivateTag twave(0x01e1,0x18,"ELSCINT1");
  if( !ds.FindDataElement( twave ) ) return 1;
  const gdcm::DataElement& wave = ds.GetDataElement( twave );
  if ( wave.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = wave.GetByteValue();
  assert( bv );

  std::ofstream os( outfilename, std::ios::binary );
  // Dump that to a CSV file:
  wave2stream( os, bv->GetPointer(), bv->GetLength() );
  os.close();

  return 0;
}
```

14.61 EmptyMask.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmEmptyMaskGenerator.h"

#include <string>
#include <cstring>

int main( int argc, char *argv[] )
{
  std::string inputdir;
  std::string outputdir;
  bool input_sopclassuid = true;
  bool grayscale_secondary_sopclassuid = false;
  if( argc < 3 ) return 1;
  inputdir = argv[1];
  outputdir = argv[2];
  // input_sopclassuid -> Use original SOP Class UID from input DICOM (Default).
  // grayscale_secondary_sopclassuid -> Use Grayscale Secondary Image Storage SOP Class UID.
  if( argc >= 3 )
  {
    input_sopclassuid = false;
    if( strcmp("input_sopclassuid", argv[3]) == 0 )
      input_sopclassuid = true;
    else if (strcmp("grayscale_secondary_sopclassuid", argv[3]) == 0 ) {
      grayscale_secondary_sopclassuid = true;
    }
  }

  //
  gdcm::EmptyMaskGenerator emg;
  if( input_sopclassuid )
    emg.SetSOPClassUIDMode( gdcm::EmptyMaskGenerator::UseOriginalSOPClassUID );
  else if( grayscale_secondary_sopclassuid )
    emg.SetSOPClassUIDMode( gdcm::EmptyMaskGenerator::UseGrayscaleSecondaryImageStorage );
  emg.SetInputDirectory( inputdir.c_str() );
  emg.SetOutputDirectory( outputdir.c_str() );
  if( !emg.Execute() )
  {
    return 1;
  }

  return 0;
}
```

14.62 EncapsulateFileInRawData.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"

#include "magic.h" // libmagic, API to file command line tool

/*
 * Let say you want to encapsulate a file type that is not defined in DICOM (exe, zip, png)
 * PNG is a bad example, unless it contains transparency (which has been deprecated).
 * It will take care of dispatching each chunk to an appropriate data item (pretty much like
 * WaveformData)
```

```
 *
 * Usage:
 * ./EncapsulateFileInRawData large_input_file.exe large_input_file.dcm
 */

// TODO:
// $ file -bi /tmp/gdcm-2.1.0.pdf
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " inputfile output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  if( !gdcm::System::FileExists( filename ) ) return 1;

  size_t s = gdcm::System::FileSize(filename);
  if( !s ) return 1;

  magic_t cookie = magic_open(MAGIC_NONE);
  const char * file_type = magic_file(cookie, filename);
  if( !file_type ) return 1;
  magic_close(cookie);

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  //gdcm::DataSet &ds = file.GetDataSet();
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );

  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );

  gdcm::Anonymizer anon;
  anon.SetFile( file );

  gdcm::MediaStorage ms = gdcm::MediaStorage::RawDataStorage;

  gdcm::UIDGenerator gen;
  anon.Replace( gdcm::Tag(0x0008,0x16), ms.GetString() );
  std::cout « ms.GetString() « std::endl;
  anon.Replace( gdcm::Tag(0x0008,0x18), gen.Generate() );


  if (!w.Write() )
    {
    std::cerr « "Could not write: " « outfilename « std::endl;
    return 1;
    }

  return 0;
}
```

14.63 ExtractEncryptedContent.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"

#include <fstream>

/*
```

```
openssl smime -encrypt -binary -aes256 -in outputfile.dcm -inform DER -out outputfile.der -outform DER
      ../trunk/Testing/Source/Data/certificate.pem

openssl smime -decrypt -binary -in out.der -inform DER -out outputfile.dcm -outform DER -inkey
      ../trunk/Testing/Source/Data/privatekey.pem ../trunk/Testing/Source/Data/certificate.pem

 */

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.der" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  const gdcm::DataElement &EncryptedAttributesSequence = ds.GetDataElement( gdcm::Tag( 0x0400,0x0500 ) );

  gdcm::SequenceOfItems *sqi = EncryptedAttributesSequence.GetValueAsSQ();

  if ( !sqi || sqi->GetNumberOfItems() != 1 ) return 1;

  gdcm::Item &item = sqi->GetItem(1);

  gdcm::DataSet &nestedds = item.GetNestedDataSet();

  if( ! nestedds.FindDataElement( gdcm::Tag( 0x0400,0x0520) ) ) return 1;

  const gdcm::DataElement &EncryptedContent = nestedds.GetDataElement( gdcm::Tag( 0x0400,0x0520) );

  const gdcm::ByteValue *bv = EncryptedContent.GetByteValue();

  std::ofstream of( outfilename, std::ios::binary );
  of.write( bv->GetPointer(), bv->GetLength() );
  of.close();

  return 0;
}
```

14.64 ExtractIconFromFile.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to either retrieve an Icon if present somewhere
 * in the file, or else generate one.
 */
#include "gdcmImageReader.h"
#include "gdcmPNMCodec.h"
#include "gdcmIconImageFilter.h"
#include "gdcmIconImageGenerator.h"

bool WriteIconAsPNM(const char* filename, const gdcm::IconImage& icon)
```

```
{
  gdcm::PNMCodec pnm;
  pnm.SetDimensions( icon.GetDimensions() );
  pnm.SetPixelFormat( icon.GetPixelFormat() );
  pnm.SetPhotometricInterpretation( icon.GetPhotometricInterpretation() );
  pnm.SetLUT( icon.GetLUT() );
  const gdcm::DataElement& in = icon.GetDataElement();
  bool b = pnm.Write( filename, in );
  assert( b );
  return b;
}

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read (or not image): " « filename « std::endl;
    return 1;
    }

  gdcm::IconImageFilter iif;
  iif.SetFile( reader.GetFile() );
  bool b = iif.Extract();

  if( b )
    {
    const gdcm::IconImage &icon = iif.GetIconImage(0);
    icon.Print( std::cout );

    if( !icon.GetTransferSyntax().IsEncapsulated() )
      {
      // Let's write out this icon as PNM file
      WriteIconAsPNM("icon.ppm", icon);
      }
    else if( icon.GetTransferSyntax() == gdcm::TransferSyntax::JPEGBaselineProcess1
      || icon.GetTransferSyntax() == gdcm::TransferSyntax::JPEGExtendedProcess2_4
    )
      {
      const gdcm::DataElement& in = icon.GetDataElement();
      const gdcm::ByteValue *bv = in.GetByteValue();
      assert( bv );
      std::ofstream out( "icon.jpg", std::ios::binary );
      out.write( bv->GetPointer(), bv->GetLength() );
      out.close();
      }
    }
  else
    {
    assert( iif.GetNumberOfIconImages() == 0 );
    std::cerr « "No Icon Found anywhere in file" « std::endl;

    const gdcm::Image &img = reader.GetImage();
    gdcm::IconImageGenerator iig;
    iig.AutoPixelMinMax(true);
    iig.SetPixmap( img );
    const unsigned int idims[2] = { 64, 64 };
    iig.SetOutputDimensions( idims );
    //iig.SetPixelMinMax(60, 868);
    if( !iig.Generate() ) return 1;
    const gdcm::IconImage & icon = iig.GetIconImage();
    WriteIconAsPNM("icon.ppm", icon);
    }

  return 0;
}
```

14.65 Extracting_All_Resolution.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```
  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include <fstream>
#include <stdint.h>
#include <string.h>
#include <assert.h>
#include <gdcm_j2k.h>
#include <gdcm_jp2.h>
#include<iostream>
#include <cstring>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmSystem.h"
#include <fstream>



#include "gdcm_openjpeg.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

void error_callback(const char *msg, void *) {
  (void)msg;
}
void warning_callback(const char *msg, void *) {
  (void)msg;
}
void info_callback(const char *msg, void *) {
  (void)msg;
}


bool Write_Resolution(gdcm::StreamImageWriter & theStreamWriter, const char *filename, int res, std::ostream&
      of, int flag,  gdcm::SequenceOfItems *sq, int No_Of_Resolutions)
{
  std::ifstream is;
  is.open( filename, std::ios::binary );
  opj_dparameters_t parameters;  /* decompression parameters */
  opj_event_mgr_t event_mgr;    /* event manager */
  opj_dinfo_t* dinfo;  /* handle to a decompressor */
  opj_cio_t *cio;
  opj_image_t *image = NULL;
  // FIXME: Do some stupid work:
  is.seekg( 0, std::ios::end);
  std::streampos buf_size = is.tellg();
  char *dummy_buffer = new char[(unsigned int)buf_size];
  is.seekg(0, std::ios::beg);
  is.read( dummy_buffer, buf_size);
  unsigned char *src = (unsigned char*)dummy_buffer;
  uint32_t file_length = (uint32_t)buf_size; // 32bits truncation should be ok since DICOM cannot have larger
      than 2Gb image


  /* configure the event callbacks (not required) */
  memset(&event_mgr, 0, sizeof(opj_event_mgr_t));
  event_mgr.error_handler = error_callback;
  event_mgr.warning_handler = warning_callback;
```

```
  event_mgr.info_handler = info_callback;

  /* set decoding parameters to default values */
  opj_set_default_decoder_parameters(&parameters);

  // default blindly copied
  parameters.cp_layer=0;
  parameters.cp_reduce= res;
  //   parameters.decod_format=-1;
  //   parameters.cod_format=-1;

  const char jp2magic[] = "\x00\x00\x00\x0C\x6A\x50\x20\x20\x0D\x0A\x87\x0A";
  if( memcmp( src, jp2magic, sizeof(jp2magic) ) == 0 )
    {
    /* JPEG-2000 compressed image data ... sigh */
    // gdcmData/ELSCINT1_JP2vsJ2K.dcm
    // gdcmData/MAROTECH_CT_JP2Lossy.dcm
    //gdcmWarningMacro( "J2K start like JPEG-2000 compressed image data instead of codestream" );
    parameters.decod_format = 1; //JP2_CFMT;
    //assert(parameters.decod_format == JP2_CFMT);
    }
  else
    {
    /* JPEG-2000 codestream */
    //parameters.decod_format = J2K_CFMT;
    //assert(parameters.decod_format == J2K_CFMT);
    assert( 0 );
    }
  parameters.cod_format = 11; // PGX_DFMT;
  //assert(parameters.cod_format == PGX_DFMT);

  /* get a decoder handle */
    dinfo = opj_create_decompress(CODEC_JP2);

  /* catch events using our callbacks and give a local context */
  opj_set_event_mgr((opj_common_ptr)dinfo, &event_mgr, NULL);

  /* setup the decoder decoding parameters using user parameters */
  opj_setup_decoder(dinfo, &parameters);

  /* open a byte stream */
  cio = opj_cio_open((opj_common_ptr)dinfo, src, file_length);

  /* decode the stream and fill the image structure */
  image = opj_decode(dinfo, cio);
  if(!image) {
    opj_destroy_decompress(dinfo);
    opj_cio_close(cio);
    //gdcmErrorMacro( "opj_decode failed" );
    return 1;
  }

      opj_cp_t * cp = ((opj_jp2_t*)dinfo->jp2_handle)->j2k->cp;
      opj_tcp_t *tcp = &cp->tcps[0];
      opj_tccp_t *tccp = &tcp->tccps[0];
    /*   std::cout « "\n No of Cols In Image" « image->x1;
      std::cout « "\n No of Rows In Image" « image->y1;
      std::cout « "\n No of Components in Image" « image->numcomps;
      std::cout « "\n No of Resolutions"« tccp->numresolutions « "\n";
*/
       opj_j2k_t* j2k = NULL;
       opj_jp2_t* jp2 = NULL;
       jp2 = (opj_jp2_t*)dinfo->jp2_handle;
       int reversible = jp2->j2k->cp->tcps->tccps->qmfbid;
       //std:: cout « reversible;
      int compno = 0;
      opj_image_comp_t *comp = &image->comps[compno];
      int Dimensions[2];
      Dimensions[0]= comp->w;
      Dimensions[1] = comp->h;
      opj_cio_close(cio);
      unsigned long len = Dimensions[0]*Dimensions[1] * image->numcomps;
      //std::cout « "\nTest" «image->comps[0].factor;
      char *raw = new char[len];
  for (unsigned int compno = 0; compno < (unsigned int)image->numcomps; compno++)
    {
    opj_image_comp_t *comp = &image->comps[compno];

    int w = image->comps[compno].w;
    int h = image->comps[compno].h;
      uint8_t *data8 = (uint8_t*)raw + compno;
```

```
      for (int i = 0; i < w * h ; i++)
        {
        int v = image->comps[compno].data[i];
        *data8 = (uint8_t)v;
        data8 += image->numcomps;
        }
    }

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( gdcm::VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );

  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::CTImageStorage );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString()));
  ds.Insert( de1 );

  const char mystr[] = "MONOCHROME2 ";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );

   gdcm::Attribute<0x0028,0x0010> row = {image->comps[0].w};
   //row.SetValue(512);
   ds.Insert( row.GetAsDataElement() );
//  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col = {image->comps[0].h};
   ds.Insert( col.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0002> at1 = {image->numcomps};
   ds.Insert( at1.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0102> at3 = {7};
  ds.Insert( at3.GetAsDataElement() );



  if (flag == 1)
  {

    for (int i=0; i < No_Of_Resolutions; i++)
    {

     int a = 1;
     int b =1;

     while(a!=((No_Of_Resolutions)-i))
        {
          b = b*2;
          a = a+1;
         }
     uint16_t row = (image->y1)/b;
     uint16_t col = (image->x1)/b;
     //std::cout « row;
     gdcm::Element<gdcm::VR::IS,gdcm::VM::VM1> el2;
     el2.SetValue(i+1);
     gdcm::DataElement rfn = el2.GetAsDataElement();     //ulr --> upper left row
     rfn.SetTag( gdcm::Tag(0x0008,0x1160) );

     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el;
     el.SetValue(1,0);
```

```
    el.SetValue(1,1);
    gdcm::DataElement ulr = el.GetAsDataElement();     //ulr --> upper left col/row
    ulr.SetTag( gdcm::Tag(0x0048,0x0201) );

    gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el1;
    el1.SetValue(col,0);
    el1.SetValue(row,1);
    gdcm::DataElement brr = el1.GetAsDataElement();
    brr.SetTag( gdcm::Tag(0x0048,0x0202) );            //brr --> bottom right col/row
   gdcm::Item it;
   gdcm::DataSet &nds = it.GetNestedDataSet();
   nds.Insert( rfn );
   nds.Insert(ulr);
   nds.Insert(brr);

   sq->AddItem(it);
  }

  gdcm::Writer w1;
  gdcm::File &file1 = w1.GetFile();
  gdcm::DataSet &ds1 = file1.GetDataSet();
  file1.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

  gdcm::UIDGenerator uid1;
  gdcm::DataElement dea( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
  dea.SetVR( gdcm::VR::UI );
  const char *u1 = uid1.Generate();
  dea.SetByteValue( u1, strlen(u1) );
  ds1.Insert( dea );

  gdcm::DataElement deb( gdcm::Tag(0x8,0x16) );
  deb.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms1( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
  deb.SetByteValue( ms1.GetString(), strlen(ms1.GetString()));
  ds1.Insert( deb );

  const char mystr1[] = "MONOCHROME2 ";
  gdcm::DataElement dec( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  dec.SetVR( gdcm::VR::CS );
  dec.SetByteValue(mystr, strlen(mystr1));
  ds1.Insert( dec );

   gdcm::Attribute<0x0028,0x0010> row1 = {image->y1};
   //row.SetValue(512);
   ds1.Insert( row1.GetAsDataElement() );
//  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col1 = {image->x1};
   ds1.Insert( col1.GetAsDataElement() );
   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames1 = {tccp->numresolutions};
   ds1.Insert( Number_Of_Frames1.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0100> ata = {8};
   ds1.Insert( ata.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0002> atb = {image->numcomps};
   ds1.Insert( atb.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0101> atc = {8};
   ds1.Insert( atc.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0102> atd = {7};
  ds1.Insert( atd.GetAsDataElement() );

   theStreamWriter.SetFile(file1);

  gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
  des.SetVR(gdcm::VR::SQ);
  //des.SetVR(gdcm::VM::VM1);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds1.Insert(des);


  if (!theStreamWriter.WriteImageInformation()){
      std::cerr « "unable to write image information" « std::endl;
      return 1; //the CanWrite function should prevent getting here, else,
      //that's a test failure
```

```
    }

 }

   theStreamWriter.SetFile(file);

  if (!theStreamWriter.CanWriteFile()){
      delete [] raw;
      std::cout « "Not able to write";
      return 0;//this means that the file was unwritable, period.
      //very similar to a ReadImageInformation failure
    }
else
   std::cout«"\nabletoread";

 // Important to write here
 std::vector<unsigned int> extent = gdcm::ImageHelper::GetDimensionsValue(file);

    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 4;
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = extent[2];
    std::cout « "\n"«xmax « "\n" « ymax«"\n"«zmax«"\n" « image->numcomps«"\n";


    if (xmax == 0 || ymax == 0)
      {
      std::cerr « "Image has no size, unable to write zero-sized image." « std::endl;
      return 0;
      }


    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially.  So, store how far you got in the buffer with each iteration.
    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout « "\n" «len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(raw[prevLen]), len);
        std::cout « "\nable to write";
        if (!theStreamWriter.Write(finalBuffer, len)){
          std::cerr « "writing failure:" « "output.dcm" « " at y = " « y « " and z= " « z « std::endl;
          delete [] raw;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
      }
    }
    delete raw;

  delete[] src;  //FIXME

if(dinfo) {
    opj_destroy_decompress(dinfo);
  }

 opj_image_destroy(image);

 return true;

}

bool Different_Resolution( gdcm::StreamImageWriter & theStreamWriter, const char *filename, int res,
      std::ostream& of)
{
  //std::vector<std::string>::const_iterator it = filenames.begin();
  bool b = true;
  int flag = 1;

 gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
```

```
 sq->SetLengthToUndefined();


  for(int i = res-1 ; i>=0; --i)
  {
    b = b && Write_Resolution( theStreamWriter, filename, i, of ,flag,sq,res);
   // b = b && Get_Resolution( theStreamWriter, filename, i, of ,0);
    flag = 0;
  }
  //b = b && Get_Lowest_Resolution( writer, sq, filename, res-1 );
  //b = b && PopulateSingeFile( writer, sq, jpeg, filename2 );
  //image.SetDimension(2,  res )
  return b;
}


int main(int argc, char *argv[])
{


   if( argc < 4 )
    {
    std::cerr « argv[0] « " input.jp2 output.dcm No. Of Resolutions " « std::endl;
    return 1;
    }
   const char *filename = argv[1];
  const char *outfilename = argv[2];
  char *resolutions = argv[3];
   int res = int((*resolutions)-48);
    //std:: cout « "\nres"« res;
   gdcm::StreamImageWriter theStreamWriter;

   std::ofstream of;
   of.open( outfilename, std::ios::out | std::ios::binary );
   theStreamWriter.SetStream(of);

   if( !Different_Resolution( theStreamWriter, filename,res,of ) ) return 1;

 uint16_t firstTag1 =  0xfffe;
 uint16_t secondTag1 = 0xe0dd;
 uint32_t thirdTag1 =  0x00000000;
 //uint16_t fourthTag1 = 0xffff;
 const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
 char* tmpBuffer2 = new char[theBufferSize1];
 memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
 //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
 assert( of && !of.eof() && of.good() );
 of.write(tmpBuffer2, theBufferSize1);
 of.flush();
 assert( of );



return 0;
}
```

14.66 Fake_Image_Using_Stream_Image_Writer.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmReader.h"
```

```
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

int main(int, char *[])
{

  char * buffer = new char[ 256 * 256 *3 ];
  // *p = (uint8_t*)buffer;
  char * p = buffer;

  gdcm::Trace::DebugOn();
  gdcm::Trace::WarningOn();

  for(int row = 0; row < 256; ++row)
  {
    for(int col = 0; col < 256; ++col)
      //for(int b = 0; b < 256; ++b)
      {
        *p++ = 255;
        *p++ = 0;
        *p++ = 0;
       }
  }

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( gdcm::VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );

  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString()));
  ds.Insert( de1 );

  const char mystr[] = "RGB";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );

   gdcm::Attribute<0x0028,0x0010> row = {256};
   //row.SetValue(512);
   ds.Insert( row.GetAsDataElement() );
 //  w.SetCheckFileMetaInformation( true );
   gdcm::Attribute<0x0028,0x0011> col = {256};
   ds.Insert( col.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0002> at1 = {3}; //bits per pixel
   ds.Insert( at1.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );

  gdcm::Attribute<0x0028,0x0102> at3 = {7};
```

```
   ds.Insert( at3.GetAsDataElement() );

 gdcm::Attribute<0x0028,0x006> at4 = {0};
 ds.Insert( at4.GetAsDataElement() );

 gdcm::Attribute<0x0028,0x0103> at5 = {0};
 ds.Insert( at5.GetAsDataElement() );

 //de.SetTag(gdcm::Tag(0x7fe0,0x0010));
 //ds.Insert(de);

  gdcm::StreamImageWriter theStreamWriter;
gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
 sq->SetLengthToUndefined();

     uint16_t row1 = 256;
     uint16_t col1 = 256;
     //std::cout « row;

     gdcm::Element<gdcm::VR::IS,gdcm::VM::VM1> el2;
     el2.SetValue(1);
     gdcm::DataElement rfn = el2.GetAsDataElement();     //rfn ---> reference frame number
     rfn.SetTag( gdcm::Tag(0x0008,0x1160) );

     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el;
     el.SetValue(1,0);
     el.SetValue(1,1);
     gdcm::DataElement ulr = el.GetAsDataElement();     //ulr --> upper left col/row
     ulr.SetTag( gdcm::Tag(0x0048,0x0201) );

     gdcm::Element<gdcm::VR::US,gdcm::VM::VM2> el1;
     el1.SetValue(col1,0);
     el1.SetValue(row1,1);
     gdcm::DataElement brr = el1.GetAsDataElement();
     brr.SetTag( gdcm::Tag(0x0048,0x0202) );            //brr --> bottom right col/row

    gdcm::Item it;
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert( rfn );
    nds.Insert(ulr);
    nds.Insert(brr);

    sq->AddItem(it);

  gdcm::DataElement des( gdcm::Tag(0x0048,0x0200) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds.Insert(des);


 theStreamWriter.SetFile(file);

  std::ofstream of;
  of.open( "output.dcm", std::ios::out | std::ios::binary );
  theStreamWriter.SetStream(of);


if (!theStreamWriter.CanWriteFile()){
      delete [] buffer;
      std::cout « "Not able to write";
      return 0;//this means that the file was unwritable, period.
      //very similar to a ReadImageInformation failure
    }
else
   std::cout«"\nabletoread";

if (!theStreamWriter.WriteImageInformation()){
      std::cerr « "unable to write image information" « std::endl;
      delete [] buffer;
      return 1; //the CanWrite function should prevent getting here, else,
      //that's a test failure
    }

 std::vector<unsigned int> extent =
      gdcm::ImageHelper::GetDimensionsValue(file);

    unsigned short xmax = extent[0];
    unsigned short ymax = extent[1];
    unsigned short theChunkSize = 1;
```

```
    unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = extent[2];

    std::cout « xmax « ymax « zmax;

    if (xmax == 0 || ymax == 0)
      {
      std::cerr « "Image has no size, unable to write zero-sized image." « std::endl;
      return 0;
      }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially.  So, store how far you got in the buffer with each iteration.
    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout « "\n" «len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(buffer[prevLen]), len);
        std::cout « "\nable to write";
        if (!theStreamWriter.Write(finalBuffer, len)){
          std::cerr « "writing failure:" « "output.dcm" « " at y = " « y « " and z= " « z « std::endl;
          delete [] buffer;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
      }
    }
    delete buffer;

 uint16_t firstTag1 =  0xfffe;
 uint16_t secondTag1 = 0xe0dd;
 uint32_t thirdTag1 =  0x00000000;
 //uint16_t fourthTag1 = 0xffff;
 const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
 char* tmpBuffer2 = new char[theBufferSize1];
 memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
 memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
 //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
 assert( of && !of.eof() && of.good() );
 of.write(tmpBuffer2, theBufferSize1);
 of.flush();
 assert( of );


  return 0;
}
```

14.67 FixBrokenJ2K.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmFile.h"

// http://www.lost.in.ua/dicom/c.dcm
```

```
//
// -> BuggyJ2Kvvvua-fixed2-j2k.dcm

/*
 * This program attempts to fix a broken J2K/DICOM:
 * It contains 2 bugs:
 * 1. The first 8 bytes seems to be random bytes: remove them
 * 2. YCC is set to 1, while image is grayscale need to set it back to 0
 *
 * Ref:
 * It's a software from http://rentgenprom.ru/ , shipped with universal digital radiographic units
 * "ProScan-2000". The Ukrainian manufacturer developed own digital radiographic unit and it is
 * compatible with software from "ProScan-2000".
 * Information found in DICOM file is:
 *
 * (0008,0070) LO [ZAO "Renthenprom" (JSC Rentgenprom) ]             # 36,1 Manufacturer
 * (0018,1020) LO [2.13.1.7]                                         # 8,1-n Software Version(s)
 *
 */
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  const gdcm::DataElement &pixeldata0 = file.GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
  const gdcm::SequenceOfFragments *sqf = pixeldata0.GetSequenceOfFragments();
  if( !sqf )
    {
    return 1;
    }
  const gdcm::Fragment &frag0 = sqf->GetFragment(0);

  gdcm::ByteValue *bv = const_cast<gdcm::ByteValue*>(frag0.GetByteValue());
  char *ptr = (char*)bv->GetVoidPointer();
  size_t len = bv->GetLength();

  static const unsigned char sig[] = {0,0,0,0,0x6A,0x70,0x32,0x63};
  if( memcmp(ptr, sig, sizeof(sig)) != 0 )
    {
    std::cerr « "magic random signature not found" « std::endl;
    return 1;
    }

  // Apparently the flag to enable a color transform on 3 color components is set in
  // the COD marker. (YCC is byte[6] in the COD marker)
  // we need to disable this flag;
  char *cod_marker = ptr + 0x35; /* 0x2d + 0x8 */ // FIXME
  if( cod_marker[0] == (char)0xff && cod_marker[1] == 0x52 )
    {
    // found start of COD
    if( cod_marker[6+2] == 1 )
      {
      // Change in place:
      *((char*)cod_marker + 6+2) = 0;
      // Prepare a new DataElement:
      gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
      pixeldata.SetVR( gdcm::VR::OB );
      gdcm::SmartPointer<gdcm::SequenceOfFragments> sq = new gdcm::SequenceOfFragments;

      gdcm::Fragment frag;
      // remove 8 first bytes:
      frag.SetByteValue( ptr + 8, (uint32_t)(len - 8) );
      sq->AddFragment( frag );
      pixeldata.SetValue( *sq );
      file.GetDataSet().Replace( pixeldata );
      }
    else
      {
      return 1;
```

```
      }
    }
  else
    {
    std::cerr « "COD not found" « (int)cod_marker[0] « std::endl;
    return 1;
    }

  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename );
  writer.CheckFileMetaInformationOff();
  if( !writer.Write() )
    {
    std::cerr « "Could not write" « std::endl;
    }

  // paranoid check:
  gdcm::ImageReader ireader;
  ireader.SetFileName( outfilename );
  if( !ireader.Read() )
    {
    std::cerr « "file written is still not valid, please report" « std::endl;
    return 1;
    }


  return 0;
}
```

14.68 FixJAIBugJPEGLS.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"

#include <fstream>

#include "gdcm_charls.h"

/*
 * This small example should show how one can handle the famous JAI-JPEGLS bug
 * It will take in as invalid DICOM/JAI-JPEG-LS and write out as Explicit Little
 * Endian. One can use `gdcmconv --jpegls` to recompress properly
 *
 * References:
 * http://charls.codeplex.com/discussions/230307?ProjectName=charls
 * http://charls.codeplex.com/workitem/7297
 * http://www.dcm4che.org/jira/browse/DCM-442
 * http://www.dcm4che.org/jira/browse/DCMEE-1144
 * http://java.net/jira/browse/JAI_IMAGEIO_CORE-183
 *
 * Explanation of the issue:
 *
 * Seems, the error is in the calculation of the default values for thresholds T1,
 * T2, T3, in particular min(MAXVAL, 4095) is not applied in
 *
 * FACTOR = (min(MAXVAL, 4095) + 128)/256
 *
 * as specified in http://www.itu.int/rec/T-REC-T.87-199806-I/en .
 *
 */
int main(int argc, char *argv[])
```

```
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::FileMetaInformation::SetSourceApplicationEntityTitle( "FixJAIBugJPEGLS" );

  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::Image &image = reader.GetImage();
  //unsigned long len = image.GetBufferLength();
  const gdcm::DataElement & in =
    reader.GetFile().GetDataSet().GetDataElement( gdcm::Tag(0x7fe0,0x0010) );
  const gdcm::SequenceOfFragments *sf = in.GetSequenceOfFragments();
  if( !sf )
    {
    std::cerr « "No pixel data (or not encapsulated)" « std::endl;
    return 1;
    }
  const unsigned int *dims = image.GetDimensions();
  if ( sf->GetNumberOfFragments() != dims[2] )
    {
    std::cerr « "Unsupported" « std::endl;
    return 1;
    }

//  unsigned long totalLen = sf->ComputeByteLength();
  std::vector<unsigned char> rgbyteOutall;
  for(unsigned int i = 0; i < sf->GetNumberOfFragments(); ++i)
    {
    const gdcm::Fragment &frag = sf->GetFragment(i);
    if( frag.IsEmpty() ) return 1;
    const gdcm::ByteValue *bv = frag.GetByteValue();
    if( !bv ) return 1;
    unsigned long totalLen = bv->GetLength();

    std::vector<char> vbuffer;
    vbuffer.resize( totalLen );
    char *buffer = vbuffer.data();
    bv->GetBuffer(buffer, totalLen);
    const unsigned char* pbyteCompressed0 = (const unsigned char*)buffer;
    while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
      {
      totalLen--;
      }

    JlsParameters metadata;
    char errorMsg[256+1]={'\0'};
    if (JpegLsReadHeader(buffer, totalLen, &metadata, errorMsg) != charls::ApiResult::OK)
      {
      std::cerr « "Can't parse jpegls: " « errorMsg « std::endl;
      return 1;
      }

    std::cout « metadata.width « std::endl;
    std::cout « metadata.height « std::endl;
    std::cout « metadata.bitsPerSample « std::endl;

    gdcm::PixelFormat const & pf = image.GetPixelFormat();
    std::cout « pf « std::endl;

    // http://charls.codeplex.com/discussions/230307?ProjectName=charls
    unsigned char marker_lse_13[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0x1F, 0xFF,
      0x00, 0x22,  // T1 = 34
      0x00, 0x83,  // T2 = 131
      0x02, 0x24,  // T3 = 548
      0x00, 0x40
    };

    unsigned char marker_lse_14[] = {
```

```
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0x3F, 0xFF,
      0x00, 0x42, // T1 = 66
      0x01, 0x03, // T2 = 259
      0x04, 0x44, // T3 = 1092
      0x00, 0x40
    };

    unsigned char marker_lse_15[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0x7F, 0xFF,
      0x00, 0x82, // T1 = 130
      0x02, 0x03, // T2 = 515
      0x08, 0x84, // T3 = 2180
      0x00, 0x40
    };

    unsigned char marker_lse_16[] = {
      0xFF, 0xF8, 0x00, 0x0D,
      0x01,
      0xFF, 0xFF,
      0x01, 0x02, // T1 = 258
      0x04, 0x03, // T2 = 1027
      0x11, 0x04, // T3 = 4356
      0x00, 0x40
    };

    const unsigned char *marker_lse = nullptr;
    switch( metadata.bitsPerSample )
      {
    case 13:
      marker_lse = marker_lse_13;
      break;
    case 14:
      marker_lse = marker_lse_14;
      break;
    case 15:
      marker_lse = marker_lse_15;
      break;
    case 16:
      marker_lse = marker_lse_16;
      break;
      }
    if( !marker_lse )
      {
      std::cerr « "Can’t handle: " « metadata.bitsPerSample « std::endl;
      return 1;
      }

    // FIXME: One should recompute the value for 0x0F
    vbuffer.insert (vbuffer.begin() + 0x0F, marker_lse, marker_lse+15);

#if 0
    std::ofstream of( "/tmp/d.jls", std::ios::binary );
    of.write( &vbuffer[0], vbuffer.size() );
    of.close();
#endif

    const char *pbyteCompressed = vbuffer.data();
    size_t cbyteCompressed = vbuffer.size(); // updated length

    JlsParameters params;
    JpegLsReadHeader(pbyteCompressed, cbyteCompressed, &params, nullptr);

    std::vector<unsigned char> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize(params.height *params.width * ((params.bitsPerSample + 7)
        / 8) * params.components);

    CharlsApiResultType result =
      JpegLsDecode(rgbyteOut.data(), rgbyteOut.size(), pbyteCompressed, cbyteCompressed, &params, errorMsg );
    if (result != charls::ApiResult::OK)
      {
      std::cerr « "Could not patch JAI-JPEGLS: " « errorMsg « std::endl;
      return 1;
      }
    rgbyteOutall.insert( rgbyteOutall.end(), rgbyteOut.begin(), rgbyteOut.end() );
```

```
    }

  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetVR( gdcm::VR::OW );
  pixeldata.SetByteValue( (char*)rgbyteOutall.data(), (uint32_t)rgbyteOutall.size() );


  // Add the pixel data element
  reader.GetFile().GetDataSet().Replace( pixeldata );
  reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);

  gdcm::Writer writer;
  writer.SetFileName( outfilename );
  writer.SetFile( reader.GetFile() );
  writer.Write();

  std::cout « "Success !" « std::endl;

  return 0;
}
```

14.69 FixOrientation.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmFile.h"
#include "gdcmOrientation.h"
#include "gdcmAttribute.h"

// Very simple orientation changer, fix invalid dataset
int main(int argc, char* argv[] )
{
  // assume AXIAL input for now
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if (! reader.Read() )
    {
    return 1;
    }

  const double axial[] = { 1,0,0, 0,1,0 };
  (void)axial;
  const double coronal[] = { 0,0,1, 1,0,0 };
  (void)coronal;
  const double sagittal[] = { 0,1,0, 0,0,1 };
  (void)sagittal;
  gdcm::Attribute<0x0020,0x0032> at1; // IPP
  (void)at1;
  gdcm::Attribute<0x0020,0x0037> at2; // IOP
  (void)at2;

  gdcm::File & f = reader.GetFile();
  gdcm::DataSet & ds = f.GetDataSet();
  at1.SetFromDataSet( ds );
```

```
#if 0
  at2.SetFromDataSet( ds );
  const double * iop = at2.GetValues();
  if( !std::equal(iop, iop + 6, axial ) )
  {
    gdcm::Orientation::OrientationType type = gdcm::Orientation::GetType ( iop );
    std::cerr « "Wrong orientation: " « gdcm::Orientation::GetLabel( type ) « std::endl;
    return 1;
  }
  at2.SetValues( sagittal );
  ds.Replace( at2.GetAsDataElement() );
#endif

  // for sagittal: swap element 0 & 2
  const double tmp0 = at1.GetValue(0);
  const double tmp2 = at1.GetValue(2);
  (void)tmp2;
  //at1.SetValue(tmp2, 0);
  //at1.SetValue(tmp0, 2);
  at1.SetValue( - tmp0 );
  ds.Replace( at1.GetAsDataElement() );

  gdcm::Writer writer;
  writer.SetFile( f );
  writer.SetFileName( outfilename );
  if ( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.70 GenAllVR.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFileExplicitFilter.h"

#include <cstdlib>
#include <cstring>


gdcm::Tag FindTagFromVR(gdcm::Dict const &dict, gdcm::VR const &vr)
{
  using gdcm::Dict;
  Dict::ConstIterator beg = dict.Begin();
  Dict::ConstIterator end = dict.End();
  Dict::ConstIterator it;
  for( it = beg; it != end; ++it)
```

```
    {
    const gdcm::Tag &t = it->first;
    const gdcm::DictEntry &de = it->second;
    const gdcm::VR &vr_de = de.GetVR();
    if( vr == vr_de && !de.GetRetired() && t.GetGroup() >= 0x8 )
      {
      return t;
      }
    }
  return gdcm::Tag(0xffff,0xffff);
}

struct rnd_gen {
  rnd_gen(char const* r = "abcdefghijklmnopqrstuvwxyz0123456789")
    : range(r), len(std::strlen(r)) { }

  char operator ()() const {
    return range[static_cast<std::size_t>(std::rand() * (1.0 / ((double)RAND_MAX + 1.0 )) * (double)len)];
  }
private:
  char const* range;
  std::size_t len;
};

/*
 */
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " output.dcm" « std::endl;
    return 1;
    }
  const char *outfilename = argv[1];
  static const gdcm::Global &g = gdcm::Global::GetInstance();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict();
  using gdcm::VR;
  using gdcm::Tag;

  gdcm::Writer w;

  gdcm::File &f = w.GetFile();
  gdcm::DataSet &ds = f.GetDataSet();

  gdcm::FileExplicitFilter fef;
  //fef.SetChangePrivateTags( true );
  fef.SetFile( w.GetFile() );
  if( !fef.Change() )
    {
    std::cerr « "Failed to change" « std::endl;
    return 1;
    }

  gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();

//  gdcm::DummyValueGenerator dvg;

  const std::size_t len = 10;
  char ss[len+1];
  ss[len] = '\0';

  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
  owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );

  // Create an item
  gdcm::Item it;
  it.SetVLToUndefined();
  gdcm::DataSet &nds = it.GetNestedDataSet();
  //    nds.Insert(owner);
  //    nds.Insert(de);

  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
```

```
  des.SetVLToUndefined();

  ds.Insert(owner);
  ds.Insert(des);

  // avoid INVALID = 0
  for(int i = 1; i < 27; ++i)
    {
    VR vr = (VR::VRType)(1LL « i);
    Tag t = FindTagFromVR( pubdict, vr );
    if( vr != VR::UN && vr != VR::SQ )
      {
      assert( t != Tag(0xffff,0xffff) );
      gdcm::DataElement de( t );
      std::generate_n(ss, len, rnd_gen());
      de.SetVR( vr );
      de.SetByteValue( ss, (uint32_t)std::strlen( ss ) );
      nds.Insert( de );
      }
    }
  sq->AddItem(it);

  // Make sure to override any UID stuff
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, (uint32_t)strlen(u) );
  ds.Insert( de );

  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
  ds.Insert( de );

  gdcm::FileMetaInformation &fmi = f.GetHeader();
  //fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
  fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

  w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.71 GenFakeIdentifyFile.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDummyValueGenerator.h"
#include "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
```

```
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmDicts.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"

#include <cstdlib>
#include <cstring>

gdcm::DataElement CreateFakeElement(gdcm::Tag const &tag, bool toremove)
{
  static const gdcm::Global &g = gdcm::Global::GetInstance();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict();
  static size_t countglobal = 0;
  static std::vector<gdcm::Tag> balcptags =
    gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
  size_t count = countglobal % balcptags.size();

  const gdcm::DictEntry &dictentry = pubdict.GetDictEntry(tag);

  gdcm::DataElement de;
  de.SetTag( tag );
  using gdcm::VR;
  const VR &vr = dictentry.GetVR();
  //if( vr != VR::INVALID )
  if( vr.IsDual() )
    {
    if( vr == VR::US_SS )
      {
      de.SetVR( VR::US );
      }
    else if( vr == VR::US_SS_OW )
      {
      de.SetVR( VR::OW );
      }
    else if( vr == VR::OB_OW )
      {
      de.SetVR( VR::OB );
      }
    }
  else
    {
    de.SetVR( vr );
    }
  const char str[] = "BasicApplicationLevelConfidentialityProfileAttributes";
  const char safe[] = "This is safe to keep";
  if( de.GetVR() != VR::SQ )
    {
    if( toremove )
      de.SetByteValue( str, (uint32_t)strlen(str) );
    else
      de.SetByteValue( safe, (uint32_t)strlen(safe) );
    }
  else
    {
    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    // Insert sequence into data set
    assert(de.GetVR() == gdcm::VR::SQ );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();
    de.SetValue(*sq);
    de.SetVLToUndefined();
    //ds.Insert(de);

    if( !toremove )
      {
      nds.Insert( CreateFakeElement( balcptags[count], true ) );
      countglobal++;
      }
    else
      {
      gdcm::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no reason to be 'anonymized'...
      nds.Insert( at1.GetAsDataElement() );
      gdcm::Attribute<0x000a,0x0000> at2 = { 0 };
      nds.Insert( at2.GetAsDataElement() );
      }
```

```
    sq->AddItem(it);
    }
  return de;
}

/*
 */
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " output.dcm" « std::endl;
    return 1;
    }
  using gdcm::Tag;
  using gdcm::VR;
  const char *outfilename = argv[1];

  std::vector<gdcm::Tag> balcptags =
    gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();

  gdcm::Writer w;
  gdcm::File &f = w.GetFile();
  gdcm::DataSet &ds = f.GetDataSet();

  // Add attribute that need to be anonymized:
  std::vector<gdcm::Tag>::const_iterator it = balcptags.begin();
  for(; it != balcptags.end(); ++it)
    {
    ds.Insert( CreateFakeElement( *it, true ) );
    }

  // Add attribute that do NOT need to be anonymized:
  static const gdcm::Global &g = gdcm::Global::GetInstance();
  static const gdcm::Dicts &dicts = g.GetDicts();
  static const gdcm::Dict &pubdict = dicts.GetPublicDict();

  using gdcm::Dict;
  Dict::ConstIterator dictit = pubdict.Begin();
  for(; dictit != pubdict.End(); ++dictit)
    {
    const gdcm::Tag &dicttag = dictit->first;
    if( dicttag == Tag(0x6e65,0x6146) ) break;
    //const gdcm::DictEntry &dictentry = dictit->second;
    ds.Insert( CreateFakeElement( dicttag, false ) );
    }
  ds.Remove( gdcm::Tag(0x400,0x500) );
  ds.Remove( gdcm::Tag(0x12,0x62) );
  ds.Remove( gdcm::Tag(0x12,0x63) );

  // Make sure to override any UID stuff
  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, (uint32_t)strlen(u) );
  //ds.Insert( de );
  ds.Replace( de );

  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::RawDataStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
  ds.Replace( de ); // replace !

  gdcm::FileMetaInformation &fmi = f.GetHeader();
  //fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
  fmi.SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

  w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.72 GenLongSeqs.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

/*
 * This example is used to generate the file:
 *
 *
 * There is a flaw in the DICOM design were it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * We need to make sure that we can store numerous Item in a SQ
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  // Create a Sequence
  gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();

  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
  owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );

  size_t nitems = 1000;
  nitems += std::numeric_limits<uint32_t>::max();
  for(unsigned int idx = 0; idx < nitems; ++idx)
    {
    // Create a dataelement
    //gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
    //de.SetByteValue(ptr, ptr_len);
    //de.SetVR( gdcm::VR::OB );

    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    //gdcm::DataSet &nds = it.GetNestedDataSet();
    //nds.Insert(owner);
```

```
    //nds.Insert(de);

    sq->AddItem(it);
    }

  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds.Insert(owner);
  ds.Insert(des);

  gdcm::Writer w;
  w.SetFile( file );
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.73 GenSeqs.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfItems.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

/*
 * This example is used to generate the file:
 *
 * gdcmConformanceTests/SequenceWithUndefinedLengthNotConvertibleToDefinedLength.dcm
 *
 * There is a flaw in the DICOM design were it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * Deflated syntax was used in this case since this synthetic example can be
 * nicely compressed using this transfer syntax.
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
```

```
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  //const unsigned int nitems = 1000;
  const unsigned int ptr_len = 42; /*94967296 / nitems; */
  //assert( ptr_len == 42949672 );
  char *ptr = new char[ptr_len];
  memset(ptr,0,ptr_len);

  // Create a Sequence
  gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new gdcm::SequenceOfItems();
  sq->SetLengthToUndefined();

  const char owner_str[] = "GDCM CONFORMANCE TESTS";
  gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
  owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
  owner.SetVR( gdcm::VR::LO );

  for(unsigned int idx = 0; idx < 10/* nitems*/; ++idx)
    {
    // Create a dataelement
    gdcm::DataElement de( gdcm::Tag(0x4d4d, 0x1002) );
    de.SetByteValue(ptr, ptr_len);
    de.SetVR( gdcm::VR::OB );

    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    nds.Insert(owner);
    nds.Insert(de);

    sq->AddItem(it);
    }

  // Insert sequence into data set
  gdcm::DataElement des( gdcm::Tag(0x4d4d,0x1001) );
  des.SetVR(gdcm::VR::SQ);
  des.SetValue(*sq);
  des.SetVLToUndefined();

  ds.Insert(owner);
  ds.Insert(des);

  gdcm::Writer w;
  w.SetFile( file );
  //w.SetCheckFileMetaInformation( true );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.74 GenerateStandardSOPClasses.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
```

```
 */

#include "gdcmDefs.h"
#include "gdcmUIDs.h"
#include "gdcmGlobal.h"
#include "gdcmMediaStorage.h"
#include "gdcmSOPClassUIDToIOD.h"

int main(int , char *[])
{
  using gdcm::MediaStorage;
  gdcm::Global& g = gdcm::Global::GetInstance();
  if( !g.LoadResourcesFiles() )
    {
    std::cerr « "Could not LoadResourcesFiles" « std::endl;
    return 1;
    }

  const gdcm::Defs &defs = g.GetDefs();

  int ret = 0;

  //std::cout « "Table B.5-1 STANDARD SOP CLASSES" « std::endl;
  std::cout « "SOP Class Name,SOP Class UID,IOD Specification (defined in PS 3.3)" « std::endl;


  gdcm::MediaStorage::MSType mst;
  for ( mst = gdcm::MediaStorage::MediaStorageDirectoryStorage; mst < gdcm::MediaStorage::MS_END;
    mst = (gdcm::MediaStorage::MSType)(mst + 1) )
    {
    const char *iod = defs.GetIODNameFromMediaStorage(mst);
    gdcm::UIDs uid;
    uid.SetFromUID( gdcm::MediaStorage::GetMSString(mst) /*mst.GetString()*/ );
    if( iod )
      {
      const char *iod_ref = gdcm::SOPClassUIDToIOD::GetIOD(uid);
      if( iod_ref )
        {
        std::string iod_ref_str = iod_ref;
        //iod_ref_str += " IOD Modules";
        //if( iod_ref_str != iod )
          {
          //std::cout « "UID: " « uid « "   ";
          std::cout « '"' « uid.GetName() « '"' «"," « '"' «uid.GetString() « '"' «"," « '"' «iod « '"' «
      std::endl;
          //std::cout « "Incompatible IODs: [" « iod « "] versus ref= [" « iod_ref_str « "]" « std::endl;
          ++ret;
          }
        }
      }
    }


  return 0;
}
```

14.75 GetJPEGSamplePrecision.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example is a little helper to detect the famous SIEMENS JPEG lossless compressed image
 * where DICOM is declared as:
 *
 *  (0028,0100) US 16                                          # 2,1 Bits Allocated
```

```
 *  (0028,0101) US 12                                                # 2,1 Bits Stored
 *  (0028,0102) US 11                                                # 2,1 High Bit
 *  (0028,0103) US 0                                                 # 2,1 Pixel Representation
 *
 * But where JPEG is:
 *
 *        JPEG_SOF_Parameters:
 *                SamplePrecision = 16
 *                nLines = 192
 *                nSamplesPerLine = 192
 *                nComponentsInFrame = 1
 *                component 0
 *                        ComponentIdentifier = 1
 *                        HorizontalSamplingFactor = 1
 *                        VerticalSamplingFactor = 1
 *                        QuantizationTableDestinationSelector = 0
 *
 *
 * This case is valid. One simply has to use the 16bits jpeg decoder to decode the 12bits stored image.
 * This used to be an issue in GDCM 1.2.x (fixed in GDCM 1.2.5)
 *
 * The main return 0 (no error) when the file read is actually a potential problem. At the end of the main
 * function, the jpeg stream is stored in the filename specified as second argument
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmJPEGCodec.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.jpg" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « filename « std::endl;
    return 1;
    }

  // The output of gdcm::Reader is a gdcm::File
  const gdcm::File &file = reader.GetFile();
  const gdcm::Image &image = reader.GetImage();

  const gdcm::TransferSyntax &ts = file.GetHeader().GetDataSetTransferSyntax();

  if( ts != gdcm::TransferSyntax::JPEGLosslessProcess14 && ts != gdcm::TransferSyntax::JPEGLosslessProcess14_1 )
    {
    std::cerr « "Input is not a lossless JPEG" « std::endl;
    return 1;
    }

  // the dataset is the the set of element we are interested in:
  const gdcm::DataSet &ds = file.GetDataSet();

  const gdcm::Tag rawTag(0x7fe0, 0x0010); // Default to Pixel Data
  const gdcm::DataElement& pdde = ds.GetDataElement( rawTag );
  const gdcm::SequenceOfFragments *sf = pdde.GetSequenceOfFragments();
  if( sf )
    {
    std::ofstream output(outfilename, std::ios::binary);
    sf->WriteBuffer(output);
    }
  else
    {
    std::cerr « "Error" « std::endl;
    return 1;
    }

  gdcm::JPEGCodec jpeg;
  std::ifstream is(outfilename, std::ios::binary);
```

```
  gdcm::PixelFormat pf ( gdcm::PixelFormat::UINT8 ); // let's pretend it's a 8bits jpeg
  jpeg.SetPixelFormat( pf );
  gdcm::TransferSyntax ts_jpg;
  bool b = jpeg.GetHeaderInfo( is, ts_jpg );
  if( !b )
    {
    return 1;
    }

  //jpeg.Print( std::cout );
  if( jpeg.GetPixelFormat().GetBitsAllocated() != image.GetPixelFormat().GetBitsAllocated()
   || jpeg.GetPixelFormat().GetBitsStored() != image.GetPixelFormat().GetBitsStored() )
    {
    std::cerr « "There is a mismatch in between DICOM declared Pixel Format and Sample Precision used in the
      JPEG stream" « std::endl;
    return 0;
    }

  std::cout « jpeg.GetPixelFormat() « std::endl;
  std::cout « image.GetPixelFormat() « std::endl;

  return 1;
}
```

14.76 GetSequenceUltrasound.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmAttribute.h"

bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min, unsigned int* X_max, unsigned int* Y_max
      );

int main(int argc, char* argv[] )
{
  // Controllo del numero di argomenti introdotti da riga di comando
  if( argc < 2 )
    {
    std::cerr « "Usage: " « std::endl;
    std::cerr « argv[0] « " inputImageFile  " « std::endl;
    return EXIT_FAILURE;
    }


  unsigned int x_min = 1;
  unsigned int y_min = 1;
  unsigned int x_max = 1;
  unsigned int y_max = 1;

  if( Region ( argv[1], &x_min, &y_min, &x_max, &y_max ) )
    {
    std::cout « "x_min = " « x_min « std::endl;
    std::cout « "y_min = " « y_min « std::endl;
    std::cout « "x_max = " « x_max « std::endl;
    std::cout « "y_max = " « y_max « std::endl;
    }

  else
    {
    std::cout « "no\n";
    }

}

bool Region ( char* nomefile, unsigned int* X_min, unsigned int* Y_min, unsigned int* X_max, unsigned int* Y_max
```

```
      )
{
  gdcm::Reader reader;
  reader.SetFileName( nomefile );
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « nomefile « std::endl;
    return false;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  gdcm::Tag tsqur(0x0018,0x6011);
  if( !ds.FindDataElement( tsqur ) )
    {
    return false;
    }

  const gdcm::DataElement &squr= ds.GetDataElement( tsqur );
  //std::cout « squr « std::endl;
  const gdcm::SequenceOfItems *sqi = squr.GetValueAsSQ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return false;
    }
  //std::cout « sqi « std::endl;

  const gdcm::Item & item = sqi->GetItem(1);
  //std::cout « item « std::endl;
  const gdcm::DataSet& nestedds = item.GetNestedDataSet();
  //std::cout « nestedds « std::endl;

  gdcm::Tag tX0(0x0018,0x6018);
  gdcm::Tag tY0(0x0018,0x601a);
  gdcm::Tag tX1(0x0018,0x601c);
  gdcm::Tag tY1(0x0018,0x601e);

  if( (!nestedds.FindDataElement( tX0 ))||(!nestedds.FindDataElement( tY0 ))||(!nestedds.FindDataElement( tX1
      ))||(!nestedds.FindDataElement( tY1 )) )
    {
    return false;
    }

  const gdcm::DataElement& deX0 = nestedds.GetDataElement( tX0 );
  const gdcm::DataElement& deY0 = nestedds.GetDataElement( tY0 );
  const gdcm::DataElement& deX1 = nestedds.GetDataElement( tX1 );
  const gdcm::DataElement& deY1 = nestedds.GetDataElement( tY1 );
  //std::cout « deX0 « std::endl « deY0 « std::endl « deX1 « std::endl « deY1 « std::endl;

  //const gdcm::ByteValue *bvX0 = deX0.GetByteValue();
  //const gdcm::ByteValue *bvY0 = deY0.GetByteValue();
  //const gdcm::ByteValue *bvX1 = deX1.GetByteValue();
  //const gdcm::ByteValue *bvY1 = deY1.GetByteValue();
  //std::cout « bvX0 « std::endl « bvY0 « std::endl « bvX1 « std::endl « bvY1 « std::endl;

  gdcm::Attribute<0x0018,0x6018> atX0;
  gdcm::Attribute<0x0018,0x601a> atY0;
  gdcm::Attribute<0x0018,0x601c> atX1;
  gdcm::Attribute<0x0018,0x601e> atY1;
  atX0.SetFromDataElement( deX0 );
  atY0.SetFromDataElement( deY0 );
  atX1.SetFromDataElement( deX1 );
  atY1.SetFromDataElement( deY1 );
  uint32_t X0 = atX0.GetValue();
  uint32_t Y0 = atY0.GetValue();
  uint32_t X1 = atX1.GetValue();
  uint32_t Y1 = atY1.GetValue();
  std::cout « X0 « std::endl « Y0 « std::endl « X1 « std::endl « Y1 « std::endl;

  *X_min = static_cast<unsigned int>(X0);
  *Y_min = static_cast<unsigned int>(Y0);
  *X_max = static_cast<unsigned int>(X1);
  *Y_max = static_cast<unsigned int>(Y1);

  //std::cout « "X_min = " « *X_min « std::endl;
  //std::cout « "Y_min = " « *Y_min « std::endl;
  //std::cout « "X_max = " « *X_max « std::endl;
  //std::cout « "Y_max = " « *Y_max « std::endl;

  return true;
```

```
}
```

14.77 GetSubSequenceData.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImage.h"
#include "gdcmImageWriter.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>

#include <map>

/*
 * This example will extract the Movie from the private group of
 * GEMS_Ultrasound_MovieGroup_001 See Attribute
 * (7fe1,60,GEMS_Ultrasound_MovieGroup_001)
 *
 * The output file will be stored in `outvid.dcm` as
 * MultiframeGrayscaleByteSecondaryCaptureImageStorage
 */
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  reader.Read();

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  const PrivateTag tseq(0x7fe1,0x1,"GEMS_Ultrasound_MovieGroup_001");

  if( !ds.FindDataElement( tseq ) ) return 1;
  const DataElement& seq = ds.GetDataElement( tseq );

  SmartPointer<SequenceOfItems> sqi = seq.GetValueAsSQ();
  assert( sqi->GetNumberOfItems() == 1 );
  Item &item = sqi->GetItem(1);
  DataSet &subds = item.GetNestedDataSet();

  const PrivateTag tseq1(0x7fe1,0x10,"GEMS_Ultrasound_MovieGroup_001");

  if( !subds.FindDataElement( tseq1 ) ) return 1;
  const DataElement& seq1 = subds.GetDataElement( tseq1 );

  SmartPointer<SequenceOfItems> sqi2 = seq1.GetValueAsSQ();
  //int n = sqi2->GetNumberOfItems();
  int index = 1;
  Item &item2 = sqi2->GetItem(index);
  DataSet &subds2 = item2.GetNestedDataSet();

  const PrivateTag tseq2(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");

  if( !subds2.FindDataElement( tseq2 ) ) return 1;
  const DataElement& seq2 = subds2.GetDataElement( tseq2 );

//    std::cout « seq2 « std::endl;

  SmartPointer<SequenceOfItems> sqi3 = seq2.GetValueAsSQ();
```

```
  size_t ni3 = sqi3->GetNumberOfItems(); (void)ni3;
  assert( sqi3->GetNumberOfItems() >= 1 );
  Item &item3 = sqi3->GetItem(1);
  DataSet &subds3 = item3.GetNestedDataSet();

  const PrivateTag tseq6(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds3.FindDataElement( tseq6 ) ) return true;
  const DataElement& seq6 = subds3.GetDataElement( tseq6 );
  SmartPointer<SequenceOfItems> sqi6 = seq6.GetValueAsSQ();
  size_t ni6= sqi6->GetNumberOfItems();
  assert( sqi6->GetNumberOfItems() >= 1 );
  const PrivateTag tseq7(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001");
  int dimx = 0, dimy = 0;
  for( size_t i6 = 1; i6 <= ni6; ++i6 )
    {
    Item &item6 = sqi6->GetItem(i6);
    DataSet &subds6 = item6.GetNestedDataSet();

    if( subds6.FindDataElement( tseq7 ) )
      {
      Element<VR::SL, VM::VM4> el;
      el.SetFromDataElement( subds6.GetDataElement( tseq7 ) );
      std::cout « "El= " « el.GetValue() « std::endl;
      dimx = el.GetValue(0);
      dimy = el.GetValue(1);
      }
    }

  const PrivateTag tseq3(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001");
  if( !subds3.FindDataElement( tseq3 ) ) return true;
  const DataElement& seq3 = subds3.GetDataElement( tseq3 );

//    std::cout « seq3 « std::endl;

  SmartPointer<SequenceOfItems> sqi4 = seq3.GetValueAsSQ();
  size_t ni4= sqi4->GetNumberOfItems();
  assert( sqi4->GetNumberOfItems() >= 1 );
  const PrivateTag tseq8(0x7fe1,0x37,"GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq4(0x7fe1,0x43,"GEMS_Ultrasound_MovieGroup_001");
  const PrivateTag tseq5(0x7fe1,0x60,"GEMS_Ultrasound_MovieGroup_001");

  std::vector<char> imbuffer;
  int dimz = 0;
  for( size_t i4 = 1; i4 <= ni4; ++i4 )
    {
    Item &item4 = sqi4->GetItem(i4);
    DataSet &subds4 = item4.GetNestedDataSet();

    if( !subds4.FindDataElement( tseq8 ) ) return true;
    const DataElement& de8 = subds4.GetDataElement( tseq8 );
    Element<VR::UL,VM::VM1> ldimz;
    ldimz.SetFromDataElement( de8 );
    dimz += ldimz.GetValue();
    if( !subds4.FindDataElement( tseq4 ) ) return true;
    const DataElement& seq4 = subds4.GetDataElement( tseq4 );
    if( !subds4.FindDataElement( tseq5 ) ) return true;
    const DataElement& seq5 = subds4.GetDataElement( tseq5 );

    //    std::cout « seq4 « std::endl;
    //    std::cout « seq5 « std::endl;

    const ByteValue *bv4 = seq4.GetByteValue();
    (void)bv4;
#if 0
      {
      std::ofstream out( "/tmp/mo4", std::ios::binary );
      out.write( bv4->GetPointer(), bv4->GetLength());
      out.close();
      }
#endif
  const ByteValue *bv5 = seq5.GetByteValue();
#if 0
    {
    std::ofstream out( "/tmp/mo5", std::ios::binary );
    out.write( bv5->GetPointer(), bv5->GetLength());
    out.close();
    }
#endif

    std::cout « bv5->GetLength() « std::endl;
    imbuffer.insert( imbuffer.begin(), bv5->GetPointer(), bv5->GetPointer() + bv5->GetLength() );
```

```
    }
  DataElement fakedata;
  fakedata.SetByteValue( imbuffer.data(), (uint32_t)imbuffer.size() );


  gdcm::SmartPointer<gdcm::Image> im = new gdcm::Image;
  im->SetNumberOfDimensions( 3 );

  im->SetDimension(0, dimx );
  im->SetDimension(1, dimy );
  im->SetDimension(2, dimz );
  size_t l1 = imbuffer.size();
  (void)l1;
  size_t l2 = im->GetBufferLength();
  (void)l2;
  assert( im->GetBufferLength() == imbuffer.size() );
  im->SetPhotometricInterpretation( gdcm::PhotometricInterpretation::MONOCHROME2 );

  im->SetDataElement( fakedata );

  gdcm::ImageWriter w;
  w.SetImage( *im );
  DataSet &dataset = w.GetFile().GetDataSet();

  gdcm::UIDGenerator uid;
  gdcm::DataElement de( Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, (uint32_t)strlen(u) );
  //ds.Insert( de );
  dataset.Replace( de );

  de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( VR::UI );
  gdcm::MediaStorage ms(
    gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
  dataset.Replace( de ); // replace !

  w.SetFileName( "outvid.dcm" );
  if( !w.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.78 HelloVizWorld.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Basic example for dealing with a DICOM file that contains an Image
 * (read: Pixel Data element)
 */

#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
```

```
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  // Instantiate the image reader:
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « filename « std::endl;
    return 1;
    }
  // If we reach here, we know for sure 2 things:
  // 1. It is a valid DICOM
  // 2. And it contains an Image !

  // The output of superclass gdcm::Reader is a gdcm::File
  //gdcm::File &file = reader.GetFile();

  // The other output of gdcm::ImageReader is a gdcm::Image
  const gdcm::Image &image = reader.GetImage();

  // Let's get some property from the image:
  unsigned int ndim = image.GetNumberOfDimensions();
  // Dimensions of the image:
  const unsigned int *dims = image.GetDimensions();
  // Origin
  const double *origin = image.GetOrigin();
  const gdcm::PhotometricInterpretation &pi = image.GetPhotometricInterpretation();
  for(unsigned int i = 0; i < ndim; ++i)
    {
    std::cout « "Dim(" « i « "): " « dims[i] « std::endl;
    }
  for(unsigned int i = 0; i < ndim; ++i)
    {
    std::cout « "Origin(" « i « "): " « origin[i] « std::endl;
    }
  std::cout « "PhotometricInterpretation: " « pi « std::endl;

  // Write the modified DataSet back to disk
  gdcm::ImageWriter writer;
  writer.SetImage( image );
  writer.SetFileName( outfilename );
  //writer.SetFile( file ); // We purposely NOT copy the meta information from the input
                            // file, and instead only pass the image
  if( !writer.Write() )
    {
    std::cerr « "Could not write: " « outfilename « std::endl;
    return 1;
    }

  return 0;
}
```

14.79 HelloWorld.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example is ... guess what this is for :)
```

```
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"

#include <iostream>

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  // Instantiate the reader:
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « filename « std::endl;
    return 1;
    }

  // If we reach here, we know for sure only 1 thing:
  // It is a valid DICOM file (potentially an old ACR-NEMA 1.0/2.0 file)
  // (Maybe, it's NOT a Dicom image -could be a DICOMDIR, a RTSTRUCT, etc-)

  // The output of gdcm::Reader is a gdcm::File
  gdcm::File &file = reader.GetFile();

  // the dataset is the the set of element we are interested in:
  gdcm::DataSet &ds = file.GetDataSet();

  // Construct a static(*) type for Image Comments :
  gdcm::Attribute<0x0020,0x4000> imagecomments;
  imagecomments.SetValue( "Hello, World !" );

  // Now replace the Image Comments from the dataset with our:
  ds.Replace( imagecomments.GetAsDataElement() );

  // Write the modified DataSet back to disk
  gdcm::Writer writer;
  writer.CheckFileMetaInformationOff(); // Do not attempt to reconstruct the file meta to preserve the file
                                        // as close to the original as possible.
  writer.SetFileName( outfilename );
  writer.SetFile( file );
  if( !writer.Write() )
    {
    std::cerr « "Could not write: " « outfilename « std::endl;
    return 1;
    }

  return 0;
}

/*
 * (*) static type, means that extra DICOM information VR & VM are computed at compilation time.
 * The compiler is deducing those values from the template arguments of the class.
 */
```

14.80 LargeVRDSExplicit.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
     PURPOSE.  See the above copyright notice for more information.

=============================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFileExplicitFilter.h"
#include "gdcmSequenceOfItems.h"


bool interpolate(const double * pts, size_t npts, std::vector<double> &out )
{
  out.clear();
  for(size_t i = 0; i < 2*npts; ++i )
    {
    const size_t j = i / 2;
    if( i % 2 )
      {
      if( j != npts - 1 )
        {
        assert( 3*j+5 < 3*npts );
        const double midpointx = (pts[3*j+0] + pts[3*j+3]) / 2;
        const double midpointy = (pts[3*j+1] + pts[3*j+4]) / 2;
        const double midpointz = (pts[3*j+2] + pts[3*j+5]) / 2;
        out.push_back( midpointx );
        out.push_back( midpointy );
        out.push_back( midpointz );
        }
      }
    else
      {
      assert( j < npts );
      out.push_back( pts[3*j+0] );
      out.push_back( pts[3*j+1] );
      out.push_back( pts[3*j+2] );
      }
    }
  assert( out.size() == 2 * npts * 3 - 3 );
  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  gdcm::FileExplicitFilter fef;
  //fef.SetChangePrivateTags( changeprivatetags );
  fef.SetFile( reader.GetFile() );
  if( !fef.Change() )
    {
    std::cerr « "Failed to change: " « filename « std::endl;
    return 1;
    }

  // (3006,0039) SQ (Sequence with undefined length #=4)     # u/l, 1 ROIContourSequence
  gdcm::Tag tag(0x3006,0x0039);

  const gdcm::DataElement &roicsq = ds.GetDataElement( tag );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = roicsq.GetValueAsSQ();
  //sqi->SetNumberOfItems( 1 );
  const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
  const gdcm::DataSet& nestedds = item.GetNestedDataSet();

  gdcm::Tag tcsq(0x3006,0x0040);
  if( !nestedds.FindDataElement( tcsq ) )
    {
```

```
    return 0;
    }
  const gdcm::DataElement& csq = nestedds.GetDataElement( tcsq );
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi2 = csq.GetValueAsSQ();
  if( !sqi2 || !sqi2->GetNumberOfItems() )
    {
    return 0;
    }
  //unsigned int nitems = sqi2->GetNumberOfItems();
  gdcm::Item & item2 = sqi2->GetItem(1); // Item start at #1

  gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
  //item2.SetVLToUndefined();
  //std::cout « nestedds2 « std::endl;
  // (3006,0050) DS [43.57636\65.52504\-10.0\46.043102\62.564945\-10.0\49.126537\60.714... # 398,48 ContourData
  gdcm::Tag tcontourdata(0x3006,0x0050);
  const gdcm::DataElement & contourdata = nestedds2.GetDataElement( tcontourdata );
  //std::cout « contourdata « std::endl;

  //const gdcm::ByteValue *bv = contourdata.GetByteValue();
  gdcm::Attribute<0x3006,0x0046> ncontourpoints;
  ncontourpoints.Set( nestedds2 );

  gdcm::Attribute<0x3006,0x0050> at;
  at.SetFromDataElement( contourdata );
  const double* pts = at.GetValues();
  unsigned int npts = at.GetNumberOfValues() / 3;

  std::vector<double> out( pts, pts + npts * 3 );
  std::vector<double> out2;

  //const unsigned int niter = 7;
  const unsigned int niter = 8;
  for( unsigned int i = 0; i < niter; ++i)
    {
    //bool b =
    interpolate(out.data(), out.size() / 3, out2);
    //const double *pout = &out[0];
    out = out2;
    out2.clear();
    }
  assert( out.size() % 3 == 0 );

  gdcm::Attribute<0x3006,0x0050> at_interpolate;
  at_interpolate.SetNumberOfValues( (unsigned int)(out.size() / 3) );
  at_interpolate.SetValues( out.data(), (uint32_t)out.size() );

  ncontourpoints.SetValue( at_interpolate.GetNumberOfValues() / 3 );
  nestedds2.Replace( at_interpolate.GetAsDataElement() );
  nestedds2.Replace( ncontourpoints.GetAsDataElement() );

  //assert(0);

  // Let's take item one and subdivide it

  gdcm::TransferSyntax ts = gdcm::TransferSyntax::ImplicitVRLittleEndian;
  ts = gdcm::TransferSyntax::ExplicitVRLittleEndian;

  gdcm::FileMetaInformation &fmi = file.GetHeader();
  const char *tsuid = gdcm::TransferSyntax::GetTSString( ts );
  // const char * is ok since padding is \0 anyway...
  gdcm::DataElement de( gdcm::Tag(0x0002,0x0010) );
  de.SetByteValue( tsuid, (uint32_t)strlen(tsuid) );
  de.SetVR( gdcm::Attribute<0x0002, 0x0010>::GetVR() );
  fmi.Replace( de );
  fmi.Remove( gdcm::Tag(0x0002,0x0012) ); // will be regenerated
  fmi.Remove( gdcm::Tag(0x0002,0x0013) ); //  '   '    '
  fmi.SetDataSetTransferSyntax(ts);


  gdcm::Writer w;
  w.SetFile( file );
  w.SetFileName( outfilename );
  if (!w.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.81 MakeTemplate.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmFileAnonymizer.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"

int main(int argc, char *argv[])
{
  if( argc < 3 ) return 1;
  const char* filename = argv[1];
  const char* outfilename = argv[2];

  //gdcm::Trace::DebugOn();

  // Remove Pixel Data element:
  gdcm::FileAnonymizer fa;
  fa.SetInputFileName( filename );
  fa.SetOutputFileName( outfilename );

  fa.Empty( gdcm::Tag(0x7fe0,0x10) );
  // cannot replace in-place DICOM header:
  //fa.Replace( gdcm::Tag(0x2,0x2), "1.2.840.10008.5.1.4.1.1.7" );

  if( !fa.Write() )
    {
    std::cerr « "impossible to remove Pixel Data attribute" « std::endl;
    return 1;
    }

  // Update the DICOM Header:
  gdcm::Reader reader;
  reader.SetFileName( outfilename );
  if( !reader.Read() )
    {
    std::cerr « "could not read back" « std::endl;
    return 1;
    }

  gdcm::File & file = reader.GetFile();
  gdcm::FileMetaInformation &fmi = file.GetHeader();
  gdcm::TransferSyntax ts = gdcm::TransferSyntax::ImplicitVRLittleEndian;
  ts = gdcm::TransferSyntax::ExplicitVRLittleEndian;
  fmi.SetDataSetTransferSyntax(ts);

  gdcm::Writer writer;
  writer.SetFile( file );
  writer.SetFileName( outfilename ); // warning overwrite file !
  if( !writer.Write() )
    {
    std::cerr « "could not write back" « std::endl;
    return 1;
    }

  return 0;
}
```

14.82 MergeTwoFiles.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example will show how one can read in two DICOM files, use the dataset
 * from file1 and use image from file2  to save it in a 3rd file.
 *
 * Eg:
 * MergeTwoFiles gdcmData/012345.002.050.dcm gdcmData/test.acr merge.dcm
 */

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmWriter.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *file1 = argv[1];
  const char *file2 = argv[2];
  const char *file3 = argv[3];

  // Read file1
  gdcm::ImageReader reader1;
  reader1.SetFileName( file1 );
  if( !reader1.Read() )
    {
    return 1;
    }

  // Read file2
  gdcm::ImageReader reader2;
  reader2.SetFileName( file2 );
  if( !reader2.Read() )
    {
    return 1;
    }

  // Ok now let's take the DataSet from file1 and the Image from file2
  // Warning: if file2 is -for example- a Secondary Capture Storage, then it has no
  // Image Orientation (Patient) thus any Image Orientation (Patient) from file1
  // will be discarded...

  // let's be fancy. In case reader2 contains explicit, but reader1 is implicit
  // we would rather see an implicit output
  if( reader1.GetFile().GetHeader().GetDataSetTransferSyntax() == gdcm::TransferSyntax::ImplicitVRLittleEndian )
    {
    reader2.GetImage().SetTransferSyntax( gdcm::TransferSyntax::ImplicitVRLittleEndian );
    }

  gdcm::ImageWriter writer;
  writer.SetFileName( file3 );
  writer.SetFile( reader1.GetFile() );
  // ImageWriter will always use all of gdcm::Image information an override anything wrong from
  // reader1.GetFile(), including the Transfer Syntax
  writer.SetImage( reader2.GetImage() );

  gdcm::DataSet &ds = reader1.GetFile().GetDataSet();

  // Make sure that SOPInstanceUID are different
  // Simply removing it is sufficient as gdcm::ImageWriter will generate one by default
  // if not found.
  ds.Remove( gdcm::Tag(0x0008,0x0018) );
  if( !writer.Write() )
    {
    return 1;
    }

  return 0;
}
```

14.83 MrProtocol.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 *
 */

/*
28 - 'MrProtocol' VM 1, VR UN, SyngoDT 0, NoOfItems 6, Data '### ASCCONV BEGIN ###
ulVersion                                = 0xbee332
tSequenceFileName                        = "%SiemensSeq%\fl_fq_shphs"
tProtocolName                            = "flash+AF8-100+AF8-through-plane+AF8-V"
tReferenceImage0                         = "1.3.12.2.1107.5.2.9.16041.3000000706210610018120000 4658"
tReferenceImage1                         = "1.3.12.2.1107.5.2.9.16041.3000000706210610018120000 4635"
tReferenceImage2                         = "1.3.12.2.1107.5.2.9.16041.3000000706210610018120000 4683"
ucScanRegionPosValid                     = 0x1
sProtConsistencyInfo.tBaselineString     = "N4_VB11A_LATEST_20031004"
sProtConsistencyInfo.flNominalB0         = 1.494
sProtConsistencyInfo.flGMax              = 22
sProtConsistencyInfo.flRiseTime          = 10
sGRADSPEC.sEddyCompensationX.aflAmplitude[0] = 0.0141111
sGRADSPEC.sEddyCompensationX.aflAmplitude[1] = 0.057038
sGRADSPEC.sEddyCompensationX.aflAmplitude[2] = -0.00986504
sGRADSPEC.sEddyCompensationX.aflAmplitude[3] = 0.00247627
sGRADSPEC.sEddyCompensationX.aflAmplitude[4] = 0.0026377
sGRADSPEC.sEddyCompensationX.aflTimeConstant[0] = 1.53826
sGRADSPEC.sEddyCompensationX.aflTimeConstant[1] = 0.746617
sGRADSPEC.sEddyCompensationX.aflTimeConstant[2] = 0.339236
sGRADSPEC.sEddyCompensationX.aflTimeConstant[3] = 0.0309809
sGRADSPEC.sEddyCompensationX.aflTimeConstant[4] = 0.00067694
sGRADSPEC.sEddyCompensationY.aflAmplitude[0] = 0.0156411
sGRADSPEC.sEddyCompensationY.aflAmplitude[1] = 0.0440623
sGRADSPEC.sEddyCompensationY.aflAmplitude[2] = -0.00782663
sGRADSPEC.sEddyCompensationY.aflAmplitude[3] = 0.00186828
sGRADSPEC.sEddyCompensationY.aflAmplitude[4] = 0.00154504
sGRADSPEC.sEddyCompensationY.aflTimeConstant[0] = 1.47145
sGRADSPEC.sEddyCompensationY.aflTimeConstant[1] = 0.750538
sGRADSPEC.sEddyCompensationY.aflTimeConstant[2] = 0.339397
sGRADSPEC.sEddyCompensationY.aflTimeConstant[3] = 0.0312962
sGRADSPEC.sEddyCompensationY.aflTimeConstant[4] = 0.000895133
sGRADSPEC.sEddyCompensationZ.aflAmplitude[0] = 0.00618504
sGRADSPEC.sEddyCompensationZ.aflAmplitude[1] = 0.00313121
sGRADSPEC.sEddyCompensationZ.aflAmplitude[2] = 0.000289346
sGRADSPEC.sEddyCompensationZ.aflAmplitude[3] = -0.00019677
sGRADSPEC.sEddyCompensationZ.aflAmplitude[4] = 7.66445e-005
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[0] = 3.37462
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[1] = 0.999351
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[2] = 0.0174646
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[3] = 0.0110094
sGRADSPEC.sEddyCompensationZ.aflTimeConstant[4] = 0.00199922
sGRADSPEC.bEddyCompensationValid         = 1
sGRADSPEC.sB0CompensationX.aflAmplitude[0] = 0.307474
sGRADSPEC.sB0CompensationX.aflAmplitude[1] = 0.029337
sGRADSPEC.sB0CompensationX.aflAmplitude[2] = -0.187118
sGRADSPEC.sB0CompensationX.aflTimeConstant[0] = 0.98583
sGRADSPEC.sB0CompensationX.aflTimeConstant[1] = 0.0308443
sGRADSPEC.sB0CompensationX.aflTimeConstant[2] = 0.000466792
sGRADSPEC.sB0CompensationY.aflAmplitude[0] = 0.365257
sGRADSPEC.sB0CompensationY.aflAmplitude[1] = -0.318647
sGRADSPEC.sB0CompensationY.aflAmplitude[2] = -0.0118978
sGRADSPEC.sB0CompensationY.aflTimeConstant[0] = 0.61535
sGRADSPEC.sB0CompensationY.aflTimeConstant[1] = 0.488831
sGRADSPEC.sB0CompensationY.aflTimeConstant[2] = 0.00199991
sGRADSPEC.sB0CompensationZ.aflAmplitude[0] = -0.44647
sGRADSPEC.sB0CompensationZ.aflAmplitude[1] = -0.0455154
sGRADSPEC.sB0CompensationZ.aflAmplitude[2] = -0.0304901
sGRADSPEC.sB0CompensationZ.aflTimeConstant[0] = 0.959231
sGRADSPEC.sB0CompensationZ.aflTimeConstant[1] = 0.0720189
```

```
sGRADSPEC.sB0CompensationZ.aflTimeConstant[2] = 0.00190141
sGRADSPEC.bB0CompensationValid           = 1
sGRADSPEC.sCrossTermCompensationXY.aflAmplitude[0] = 0.00105046
sGRADSPEC.sCrossTermCompensationXY.aflTimeConstant[0] = 0.842014
sGRADSPEC.sCrossTermCompensationXZ.aflAmplitude[0] = -0.00150189
sGRADSPEC.sCrossTermCompensationXZ.aflTimeConstant[0] = 0.736169
sGRADSPEC.sCrossTermCompensationYX.aflAmplitude[0] = -5.5278e-005
sGRADSPEC.sCrossTermCompensationYX.aflTimeConstant[0] = 0.228697
sGRADSPEC.sCrossTermCompensationYZ.aflAmplitude[0] = 0.000307999
sGRADSPEC.sCrossTermCompensationYZ.aflTimeConstant[0] = 1.19431
sGRADSPEC.sCrossTermCompensationZX.aflAmplitude[0] = -0.000286868
sGRADSPEC.sCrossTermCompensationZX.aflTimeConstant[0] = 0.665979
sGRADSPEC.sCrossTermCompensationZY.aflAmplitude[0] = 0.000355175
sGRADSPEC.sCrossTermCompensationZY.aflTimeConstant[0] = 0.844189
sGRADSPEC.bCrossTermCompensationValid    = 1
sGRADSPEC.lOffsetX                       = 25
sGRADSPEC.lOffsetY                       = 84
sGRADSPEC.lOffsetZ                       = 47
sGRADSPEC.bOffsetValid                   = 1
sGRADSPEC.lDelayX                        = 12
sGRADSPEC.lDelayY                        = 11
sGRADSPEC.lDelayZ                        = 9
sGRADSPEC.bDelayValid                    = 1
sGRADSPEC.flSensitivityX                 = 0.000264087
sGRADSPEC.flSensitivityY                 = 0.000272009
sGRADSPEC.flSensitivityZ                 = 0.000272677
sGRADSPEC.bSensitivityValid              = 1
sGRADSPEC.alShimCurrent[0]               = 183
sGRADSPEC.alShimCurrent[1]               = -25
sGRADSPEC.alShimCurrent[2]               = -85
sGRADSPEC.alShimCurrent[3]               = 378
sGRADSPEC.alShimCurrent[4]               = 82
sGRADSPEC.bShimCurrentValid              = 1
sGRADSPEC.ucMode                         = 0x2
sTXSPEC.asNucleusInfo[0].tNucleus        = "1H"
sTXSPEC.asNucleusInfo[0].lFrequency      = 63684693
sTXSPEC.asNucleusInfo[0].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[0].flReferenceAmplitude = 359.734
sTXSPEC.asNucleusInfo[0].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[0].flAmplitudeCorrection = 1
sTXSPEC.asNucleusInfo[0].bAmplitudeCorrectionValid = 1
sTXSPEC.asNucleusInfo[1].bFrequencyValid = 1
sTXSPEC.asNucleusInfo[1].bReferenceAmplitudeValid = 1
sTXSPEC.asNucleusInfo[1].bAmplitudeCorrectionValid = 1
sTXSPEC.aRFPULSE[0].tName                = "03GreFCE"
sTXSPEC.aRFPULSE[0].bAmplitudeValid      = 0x1
sTXSPEC.aRFPULSE[0].flAmplitude          = 147.095
sTXSPEC.aRFPULSE[1].tName                = "02GreFCE"
sTXSPEC.aRFPULSE[1].bAmplitudeValid      = 0x1
sTXSPEC.aRFPULSE[1].flAmplitude          = 147.095
sTXSPEC.aRFPULSE[2].tName                = "01GreFCE"
sTXSPEC.aRFPULSE[2].bAmplitudeValid      = 0x1
sTXSPEC.aRFPULSE[2].flAmplitude          = 147.095
sTXSPEC.lNoOfTraPulses                   = 3
sTXSPEC.lBTB1ParallelCapacity            = 2
sTXSPEC.lBTB1SerialCapacity              = 24
sTXSPEC.lBTB2ParallelCapacity            = 2
sTXSPEC.lBTB2SerialCapacity              = 26
sTXSPEC.bBTBValid                        = 1
sTXSPEC.flKDynMagnitudeMin               = 0.5
sTXSPEC.flKDynMagnitudeMax               = 1.5
sTXSPEC.flKDynMagnitudeClipLow           = 0.96
sTXSPEC.flKDynMagnitudeClipHigh          = 1.04
sTXSPEC.flKDynPhaseMax                   = 0.698132
sTXSPEC.flKDynPhaseClip                  = 0.174533
sTXSPEC.bKDynValid                       = 1
sTXSPEC.ucRFPulseType                    = 0x1
sTXSPEC.ucExcitMode                      = 0x1
sTXSPEC.ucSimultaneousExcitation         = 0x1
sRXSPEC.lGain                            = 1
sRXSPEC.bGainValid                       = 1
sRXSPEC.aFFT_SCALE[0].lRxChannel         = 1
sRXSPEC.aFFT_SCALE[0].flFactor           = 1.06857
sRXSPEC.aFFT_SCALE[0].bValid             = 1
sRXSPEC.aFFT_SCALE[1].lRxChannel         = 2
sRXSPEC.aFFT_SCALE[1].flFactor           = 1.07454
sRXSPEC.aFFT_SCALE[1].bValid             = 1
sRXSPEC.aFFT_SCALE[2].lRxChannel         = 3
sRXSPEC.aFFT_SCALE[2].flFactor           = 1.06622
sRXSPEC.aFFT_SCALE[2].bValid             = 1
sRXSPEC.aFFT_SCALE[3].lRxChannel         = 4
```

```
sRXSPEC.aFFT_SCALE[3].flFactor          = 1.06524
sRXSPEC.aFFT_SCALE[3].bValid            = 1
sRXSPEC.aFFT_SCALE[4].lRxChannel        = 5
sRXSPEC.aFFT_SCALE[4].flFactor          = 0.982692
sRXSPEC.aFFT_SCALE[4].bValid            = 1
sRXSPEC.aFFT_SCALE[5].lRxChannel        = 6
sRXSPEC.aFFT_SCALE[5].flFactor          = 0.988603
sRXSPEC.aFFT_SCALE[5].bValid            = 1
sRXSPEC.aFFT_SCALE[6].lRxChannel        = 7
sRXSPEC.aFFT_SCALE[6].flFactor          = 0.981538
sRXSPEC.aFFT_SCALE[6].bValid            = 1
sRXSPEC.aFFT_SCALE[7].lRxChannel        = 8
sRXSPEC.aFFT_SCALE[7].flFactor          = 1.00856
sRXSPEC.aFFT_SCALE[7].bValid            = 1
sRXSPEC.bVariCapVoltagesValid           = 1
sRXSPEC.alDwellTime[0]                  = 8500
sAdjFreSpec.ulMode                      = 0x1
sAdjFreSpec.ucAdjWithBC                 = 0x1
sAdjTraSpec.ucAdjWithBC                 = 0x1
sAdjShimSpec.ulMode                     = 0x1
sAdjShimSpec.ucAdjWithBC                = 0x1
sAdjWatSupSpec.ulMode                   = 0x1
sAdjWatSupSpec.ucAdjWithBC              = 0x1
alTR[0]                                 = 37000
lContrasts                              = 1
alTE[0]                                 = 4000
acFlowComp[0]                           = 1
lCombinedEchoes                         = 1
sSliceArray.asSlice[0].sPosition.dSag   = 35.31199581
sSliceArray.asSlice[0].sPosition.dCor   = -8.387765754
sSliceArray.asSlice[0].sPosition.dTra   = -23.13178296
sSliceArray.asSlice[0].sNormal.dSag     = 0.771051253
sSliceArray.asSlice[0].sNormal.dCor     = 0.5863890019
sSliceArray.asSlice[0].sNormal.dTra     = -0.2482496801
sSliceArray.asSlice[0].dThickness       = 6
sSliceArray.asSlice[0].dPhaseFOV        = 187.5
sSliceArray.asSlice[0].dReadoutFOV      = 250
sSliceArray.lSize                       = 1
sSliceArray.lSag                        = 1
sSliceArray.lConc                       = 1
sSliceArray.ucMode                      = 0x1
sSliceArray.sTSat.dThickness            = 40
sSliceArray.sTSat.dGap                  = 10
sGroupArray.asGroup[0].nSize            = 1
sGroupArray.asGroup[0].dDistFact        = 0.2
sGroupArray.anMember[1]                 = -1
sGroupArray.lSize                       = 1
sGroupArray.sPSat.dThickness            = 50
sGroupArray.sPSat.dGap                  = 10
sAutoAlign.dAAMatrix[0]                 = 1
sAutoAlign.dAAMatrix[5]                 = 1
sAutoAlign.dAAMatrix[10]                = 1
sAutoAlign.dAAMatrix[15]                = 1
sNavigatorPara.ucRespComp               = 0x4
sPrepPulses.ucFatSat                    = 0x4
sPrepPulses.ucWaterSat                  = 0x4
sPrepPulses.ucInversion                 = 0x4
sPrepPulses.ucSatRecovery               = 0x1
sPrepPulses.ucFatSatMode                = 0x2
sKSpace.lBaseResolution                 = 256
sKSpace.lPhaseEncodingLines             = 192
sKSpace.dPhaseResolution                = 1
sKSpace.lPartitions                     = 32
sKSpace.lImagesPerSlab                  = 32
sKSpace.dSliceResolution                = 1
sKSpace.ucPhasePartialFourier           = 0x10
sKSpace.ucSlicePartialFourier           = 0x10
sKSpace.ucAveragingMode                 = 0x2
sKSpace.ucMultiSliceMode                = 0x1
sKSpace.ucDimension                     = 0x2
sKSpace.ucAsymmetricEchoAllowed         = 0x1
sKSpace.unReordering                    = 0x1
sFastImaging.lEPIFactor                 = 1
sFastImaging.lTurboFactor               = 1
sFastImaging.lSegments                  = 3
sFastImaging.ulEnableRFSpoiling         = 0x1
sPhysioImaging.lSignal1                 = 2
sPhysioImaging.lMethod1                 = 2
sPhysioImaging.lSignal2                 = 1
sPhysioImaging.lMethod2                 = 1
sPhysioImaging.lPhases                  = 21
```

```
sPhysioImaging.lRetroGatedImages         = 16
sPhysioImaging.sPhysioECG.lScanWindow    = 805
sPhysioImaging.sPhysioECG.lTriggerPulses = 1
sPhysioImaging.sPhysioECG.lTriggerWindow = 5
sPhysioImaging.sPhysioECG.lArrhythmiaDetection = 1
sPhysioImaging.sPhysioECG.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioECG.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioPulse.lTriggerPulses = 1
sPhysioImaging.sPhysioPulse.lTriggerWindow = 5
sPhysioImaging.sPhysioPulse.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioPulse.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioExt.lTriggerPulses = 1
sPhysioImaging.sPhysioExt.lTriggerWindow = 5
sPhysioImaging.sPhysioExt.lCardiacGateOnThreshold = 100000
sPhysioImaging.sPhysioExt.lCardiacGateOffThreshold = 700000
sPhysioImaging.sPhysioResp.lRespGateThreshold = 20
sPhysioImaging.sPhysioResp.lRespGatePhase = 2
sPhysioImaging.sPhysioResp.dGatingRatio  = 0.3
sSpecPara.lPhaseCyclingType              = 1
sSpecPara.lPhaseEncodingType             = 1
sSpecPara.lRFExcitationBandwidth         = 1
sSpecPara.ucRemoveOversampling           = 0x1
sSpecPara.lDecouplingType                = 1
sSpecPara.lNOEType                       = 1
sSpecPara.lExcitationType                = 1
sSpecPara.lSpectralSuppression           = 1
sDiffusion.ulMode                        = 0x1
sAngio.sFlowArray.asElm[0].nVelocity     = 100
sAngio.sFlowArray.asElm[0].nDir          = 0x4
sAngio.sFlowArray.lSize                  = 1
sAngio.ucPCFlowMode                      = 0x2
sAngio.ucTOFInflow                       = 0x4
sAngio.ucRephasedImage                   = 0x1
sAngio.ucPhaseImage                      = 0x1
sEllipticalFilter.ucMode                 = 0x1
sPat.lAccelFactPE                        = 1
sPat.lAccelFact3D                        = 1
sPat.ucPATMode                           = 0x1
sPat.ucRefScanMode                       = 0x1
ucAutoMovie                              = 0x1
ucDisableChangeStoreImages               = 0x1
ucReconstructionMode                     = 0x1
ucPHAPSMode                              = 0x1
ucDixon                                  = 0x1
lAverages                                = 2
adFlipAngleDegree[0]                     = 30
lScanTimeSec                             = 103
lTotalScanTimeSec                        = 112
dRefSNR                                  = 165404.1473
dRefSNR_VOI                              = 165404.1473
tdefaultEVAProt                          = "%SiemensEvaDefProt%\Inline\Inline.evp"
tcurrentEVAProt                          = "%CURRENTEVAPROT%\EVA2A5.tmp"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[0].sCoilElementID.tElement = "PP6"
sCOIL_SELECT_MEAS.asList[0].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[0].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[1].sCoilElementID.tElement = "PP5"
sCOIL_SELECT_MEAS.asList[1].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[1].lRxChannelConnected = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[2].sCoilElementID.tElement = "PP3"
sCOIL_SELECT_MEAS.asList[2].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[2].lRxChannelConnected = 2
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[3].sCoilElementID.tElement = "PP4"
sCOIL_SELECT_MEAS.asList[3].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[3].lRxChannelConnected = 3
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[4].sCoilElementID.tElement = "PP2"
sCOIL_SELECT_MEAS.asList[4].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[4].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tCoilID = "6_Ch_Body_P"
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[5].sCoilElementID.tElement = "PP1"
sCOIL_SELECT_MEAS.asList[5].lElementSelected = 1
```

```
sCOIL_SELECT_MEAS.asList[5].lRxChannelConnected = 4
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[6].sCoilElementID.tElement = "PA6"
sCOIL_SELECT_MEAS.asList[6].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[6].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[7].sCoilElementID.tElement = "PA5"
sCOIL_SELECT_MEAS.asList[7].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[7].lRxChannelConnected = 5
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[8].sCoilElementID.tElement = "PA3"
sCOIL_SELECT_MEAS.asList[8].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[8].lRxChannelConnected = 6
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[9].sCoilElementID.tElement = "PA4"
sCOIL_SELECT_MEAS.asList[9].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[9].lRxChannelConnected = 7
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[10].sCoilElementID.tElement = "PA2"
sCOIL_SELECT_MEAS.asList[10].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[10].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tCoilID = "6_Ch_Body_A"
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.lCoilCopy = 1
sCOIL_SELECT_MEAS.asList[11].sCoilElementID.tElement = "PA1"
sCOIL_SELECT_MEAS.asList[11].lElementSelected = 1
sCOIL_SELECT_MEAS.asList[11].lRxChannelConnected = 8
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[0] = 0xff
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[1] = 0x76
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[2] = 0x78
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[3] = 0x87
sCOIL_SELECT_MEAS.sCOILPLUGS.aulPlugId[4] = 0x67
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[0] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[1] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[2] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[3] = 0x2
sCOIL_SELECT_MEAS.sCOILPLUGS.auiNmbrOfNibbles[4] = 0x2
sEFISPEC.bEFIDataValid                   = 1
### ASCCONV END ###
'
*/

/*
 * Table of equivalence:
 *
ulVersion                                = 0xbee332
<=>
27 - 'MrProtocolVersion' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '12510002'
*/

#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"

#include <map>

#include <math.h>

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }

  gdcm::CSAHeader csa;
```

```
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  //const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
  const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

  if( ds.FindDataElement( t2 ) )
    {
    csa.LoadFromDataElement( ds.GetDataElement( t2 ) );
    //csa.Print( std::cout );
    }

  if( !csa.FindCSAElementByName( "MrProtocol" ) )
    {
    return 1;
    }
  const gdcm::CSAElement &csael = csa.GetCSAElementByName( "MrProtocol" );
  //std::cout « csael « std::endl;

  const gdcm::ByteValue *bv = csael.GetByteValue();
  if( !bv )
    {
    return 1;
    }
  std::string str(bv->GetPointer(), bv->GetLength());
  std::istringstream is(str);
  std::string s;
  typedef std::map< std::string, std::string > MyMapType;
  MyMapType mymap;
  while( std::getline(is, s ) )
    {
    std::string::size_type pos = s.find( '=' );
    if( pos != std::string::npos )
      {
      std::string sub1 = s.substr(0, pos);
      sub1.erase( sub1.find_last_not_of(' ') + 1);
      std::string sub2 = s.substr(pos+1); // skip the '=' char
      sub2.erase( 0, sub2.find_first_not_of(' '));
      //std::cout « sub1 « std::endl;
      mymap.insert( MyMapType::value_type(sub1, sub2) );
      }
    else
      {
      // ### ASCCONV BEGIN ###
      // ### ASCCONV END ###
      }
    }
  const char fourierstr[] = "sKSpace.ucSlicePartialFourier";
  const gdcm::CSAHeaderDict &csadict = gdcm::Global::GetInstance().GetDicts().GetCSAHeaderDict();
  const gdcm::CSAHeaderDictEntry &fourier = csadict.GetCSAHeaderDictEntry( fourierstr );
  std::cout « fourier « std::endl;
  MyMapType::const_iterator it = mymap.find ( fourierstr );
  if( it == mymap.end() ) return 1;
  //std::cout « it->second « std::endl;
  const std::string &partial_fourier = it->second;
  if( partial_fourier == "0x1" )
    {
    std::cout « "partial fourier is 4/8" « std::endl;
    }
  else if( partial_fourier == "0x2" )
    {
    std::cout « "partial fourier is 5/8" « std::endl;
    }
  else if( partial_fourier == "0x4" )
    {
    std::cout « "partial fourier is 6/8" « std::endl;
    }
  else if( partial_fourier == "0x8" )
    {
    std::cout « "partial fourier is 7/8" « std::endl;
    }
  else if( partial_fourier == "0x10" )
    {
    std::cout « "partial fourier is 8/8" « std::endl;
    }
  else
    {
    std::cerr « "Impossible: " « partial_fourier « std::endl;
    return 1;
    }
/*
This is the Flip Angle:
```

```
adFlipAngleDegree[0]                    = 30

One can find it also in the protocol:

...
      <ParamFunctor.""TlmapFunctor"">
      {
        <Class> ""TlmapFunctor@IceImagePostProcFunctors""

        <ParamBool.""EXECUTE"">  { }
        <ParamDouble.""Flip1_deg"">  { <Precision> 16  14.7378520000000000  }
...

*/
  // Below is an attempt to play with the CSAHeader dict:
#if 0
  const char gspec[] = "sGRADSPEC.flSensitivityX";
  it = mymap.find( gspec );
  if( it == mymap.end() ) return 1;
  const std::string &dummy = it->second;
  std::cout « dummy « std::endl;

  const gdcm::CSAHeaderDictEntry &csaentry = csadict.GetCSAHeaderDictEntry( gspec );
  std::cout « csaentry  « std::endl;
#endif


/*
    sSliceArray.ucMode -- should be in (1, 2, 4)
    enum SeriesMode
    {
      ASCENDING   = 0x01,
      DESCENDING  = 0x02,
      INTERLEAVED = 0x04
    };
*/
  const char sliceorderstr[] = "sSliceArray.ucMode";
  const gdcm::CSAHeaderDictEntry &sliceorder = csadict.GetCSAHeaderDictEntry( sliceorderstr );
  std::cout « sliceorder « std::endl;

  it = mymap.find ( sliceorderstr );
  if( it == mymap.end() ) return 1;
  const std::string &slice_order = it->second;
  if( slice_order == "0x1" )
    {
    std::cout « "slice_order: ASCENDING" « std::endl;
    }
  else if( slice_order == "0x2" )
    {
    std::cout « "slice_order: DESCENDING" « std::endl;
    }
  else if( slice_order == "0x4" )
    {
    std::cout « "slice_order: INTERLEAVED" « std::endl;
    }
  else
    {
    std::cerr « "Impossible: " « slice_order « std::endl;
    return 1;
    }

  gdcm::MrProtocol mrprot;
  if(  csa.GetMrProtocol(ds, mrprot) )
  {
    std::cout « mrprot « std::endl;
  }

  return 0;
}
```

14.84 PrintLUT.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```
  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 */

#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " input.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];

  // Instantiate the image reader:
  gdcm::ImageReader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « filename « std::endl;
    return 1;
    }
  const gdcm::Image &image = reader.GetImage();

  const gdcm::LookupTable & lut = image.GetLUT();
  lut.Print( std::cout );

  return 0;
}
```

14.85 PublicDict.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Dummy example to show GDCM Dict(s) API (Part 6) + Collected Private Attributes:
 */

#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmCSAHeader.h"
#include "gdcmPrivateTag.h"

int main(int , char *[])
{
  const gdcm::Global& g = gdcm::Global::GetInstance(); // sum of all knowledge !
  const gdcm::Dicts &dicts = g.GetDicts();
```

```
  const gdcm::Dict &pub = dicts.GetPublicDict(); // Part 6

  //std::cout « pub « std::endl;

  // 3 different ways to access the same information

  // 1. From the public dict only:
  gdcm::Tag patient_name(0x10,0x10);
  const gdcm::DictEntry &entry1 = pub.GetDictEntry(patient_name);
  std::cout « entry1 « std::endl;

  // 2. From all dicts:
  const gdcm::DictEntry &entry2 = dicts.GetDictEntry(patient_name);
  std::cout « entry2 « std::endl;

  // 3. This solution is the most flexible solution as you can request using the same
  // API either a public tag or a private tag
  const char *strowner = nullptr;
  const gdcm::DictEntry &entry3 = dicts.GetDictEntry(patient_name,strowner);
  std::cout « entry3 « std::endl;

  // Private attributes:

  // try with a private tag now:
  const gdcm::PrivateTag &private_tag = gdcm::CSAHeader::GetCSAImageHeaderInfoTag();
  //std::cout « private_tag « std::endl;
  const gdcm::DictEntry &entry4 = dicts.GetDictEntry(private_tag,private_tag.GetOwner());
  std::cout « entry4 « std::endl;

  // Let's pretend that private lookup is on 0x10xx elements:
  gdcm::PrivateTag dummy = private_tag;
  dummy.SetElement( (uint16_t)(0x1000 + dummy.GetElement()) );
  const gdcm::DictEntry &entry5 = dicts.GetDictEntry(dummy,dummy.GetOwner());
  std::cout « entry5 « std::endl;


  return 0;
}
```

14.86 QIDO-RS.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmJSON.h"

/*
 * Simple QIDO-RS round-trip to test implementation of gdcm::JSON
 * See Sup166 for details
 */
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  using namespace gdcm;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() ) return 1;

  gdcm::JSON json;
  json.PrettyPrintOn();
  std::stringstream ss;
  const gdcm::File & f = reader.GetFile();
```

```
  json.Code( f.GetDataSet(), ss);

  std::cout « ss.str() « std::endl;

  gdcm::Writer w;
  gdcm::File & ff = w.GetFile();
  ff.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );
  if( !json.Decode(ss, ff.GetDataSet() ) )
    {
    std::cerr « "Could not decode" « std::endl;
    return 1;
    }
  w.SetFileName( "/tmp/debug.dcm" );
  if( !w.Write() ) return 1;

  return 0;
}
```

14.87 ReadAndDumpDICOMDIR.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to read and dump a DICOMDIR File
 *
 * Thanks:
 *   Tom Marynowski (lordglub gmail) for contributing this example
 */
#include "gdcmReader.h"
#include "gdcmMediaStorage.h"

typedef std::set<gdcm::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];

  gdcm::Reader reader;
  reader.SetFileName( filename);
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « filename « std::endl;
    return 1;
    }
  std::stringstream strm;

  gdcm::File &file = reader.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();
  gdcm::FileMetaInformation &fmi = file.GetHeader();

  gdcm::MediaStorage ms;
  ms.SetFromFile(file);
  if( ms != gdcm::MediaStorage::MediaStorageDirectoryStorage )
    {
    std::cout « "This file is not a DICOMDIR" « std::endl;
    return 1;
    }

  if (fmi.FindDataElement( gdcm::Tag (0x0002, 0x0002)))
    {   strm.str("");
    fmi.GetDataElement( gdcm::Tag (0x0002, 0x0002) ).GetValue().Print(strm);
    }
  else
```

```
    {
    std::cerr « " Media Storage Sop Class UID not present" « std::endl;
    }

  //TODO il faut trimer strm.str() avant la comparaison au cas ou...
  if ("1.2.840.10008.1.3.10"!=strm.str())
    {
    std::cout « "This file is not a DICOMDIR" « std::endl;
    return 1;
    }

  ConstIterator it = ds.GetDES().begin();

  for( ; it != ds.GetDES().end(); ++it)
    {

    if (it->GetTag()==gdcm::Tag (0x0004, 0x1220))
      {

      const gdcm::DataElement &de = (*it);
      // ne pas utiliser GetSequenceOfItems pour extraire les items
      gdcm::SmartPointer<gdcm::SequenceOfItems> sqi =de.GetValueAsSQ();
      unsigned int itemused = 1;
      while (itemused<=sqi->GetNumberOfItems())

        {
        strm.str("");

        if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
          sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

        //TODO il faut trimer strm.str() avant la comparaison
        while((strm.str()=="PATIENT")||((strm.str()=="PATIENT ")))
          {
          std::cout « strm.str() « std::endl;
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0010, 0x0010)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0010, 0x0010)).GetValue().Print(strm);
          std::cout « "PATIENT NAME : " « strm.str() « std::endl;


          //PATIENT ID
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0010, 0x0020)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0010, 0x0020)).GetValue().Print(strm);
          std::cout « "PATIENT ID : " « strm.str() « std::endl;

          /*ADD TAG TO READ HERE*/
          std::cout « "========================== "  « std::endl;
          itemused++;
          strm.str("");
          if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

          //TODO il faut trimer strm.str() avant la comparaison
          while((strm.str()=="STUDY")||((strm.str()=="STUDY ")))
            {
            std::cout « "  " « strm.str() « std::endl;
            //UID
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0020, 0x000d)))
              sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0020, 0x000d)).GetValue().Print(strm);
            std::cout « "      STUDY UID : " « strm.str() « std::endl;

            //STUDY DATE
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x0020)))
              sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x0020)).GetValue().Print(strm);
            std::cout « "      STUDY DATE : " « strm.str() « std::endl;

            //STUDY DESCRIPTION
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x1030)))
              sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x1030)).GetValue().Print(strm);
            std::cout « "      STUDY DESCRIPTION : " « strm.str() « std::endl;

            /*ADD TAG TO READ HERE*/
            std::cout « "      " « "========================== "  « std::endl;

            itemused++;
            strm.str("");
```

```
            if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
              sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

            //TODO il faut trimer strm.str() avant la comparaison
            while((strm.str()=="SERIES")|||((strm.str()=="SERIES ")))
              {
              std::cout « "      " « strm.str() « std::endl;
              strm.str("");
              if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0020, 0x000e)))
                sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0020, 0x000e)).GetValue().Print(strm);
              std::cout « "          SERIE UID" « strm.str() « std::endl;

              //SERIE MODALITY
              strm.str("");
              if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x0060)))
                sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x0060)).GetValue().Print(strm);
              std::cout « "          SERIE MODALITY" « strm.str() « std::endl;

              //SERIE DESCRIPTION
              strm.str("");
              if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0008, 0x103e)))
                sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0008, 0x103e)).GetValue().Print(strm);
              std::cout « "          SERIE DESCRIPTION" « strm.str() « std::endl;


              /*ADD TAG TO READ HERE*/

              std::cout « "          " « "========================== "  « std::endl;
              itemused++;
              strm.str("");
              if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
                sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);


              //TODO il faut trimer strm.str() avant la comparaison
              while ((strm.str()=="IMAGE")|||((strm.str()=="IMAGE ")))
                // if(tmp=="IMAGE")
                {
                std::cout « "          " « strm.str() « std::endl;


                //UID
                strm.str("");
                if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1511)))
                  sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1511)).GetValue().Print(strm);
                std::cout « "              IMAGE UID : " « strm.str() « std::endl;

                //PATH de l'image
                strm.str("");
                if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1500)))
                  sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1500)).GetValue().Print(strm);
                std::cout « "              IMAGE PATH : " « strm.str() « std::endl;
                /*ADD TAG TO READ HERE*/



                if(itemused < sqi->GetNumberOfItems())
                  {itemused++;
                  }else{break;}

                strm.str("");

                if (sqi->GetItem(itemused).FindDataElement(gdcm::Tag (0x0004, 0x1430)))
                  sqi->GetItem(itemused).GetDataElement(gdcm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

                }
              }
            }
          }
        itemused++;
        }
      }
    }
  return 0;
}
```

14.88 ReadAndDumpDICOMDIR2.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2017 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to read and dump a DICOMDIR File
 *
 * Thanks:
 *   Tom Marynowski (lordglub gmail) for contributing the original
 *    ReadAndDumpDICOMDIR.cxx example
 *   Mihail Isakov for contributing offset calculation code here:
 *    https://sourceforge.net/p/gdcm/mailman/gdcm-developers/?viewmonth=201707&viewday=15
 *   Tod Baudais for combining the above and cleaning up this example
 */

#include <string>
#include <unordered_map>
#include <iostream>
#include <memory>

#include "gdcmReader.h"
#include "gdcmAttribute.h"
#include "gdcmDirectory.h"

//==============================================================================
//==============================================================================

#define TAG_MEDIA_STORAGE_SOP_CLASS_UID 0x0002,0x0002
#define TAG_DIRECTORY_RECORD_SEQUENCE 0x0004,0x1220
#define TAG_DIRECTORY_RECORD_TYPE 0x0004,0x1430
#define TAG_PATIENTS_NAME 0x0010,0x0010
#define TAG_PATIENT_ID 0x0010,0x0020
#define TAG_STUDY_DATE 0x0008,0x0020
#define TAG_STUDY_DESCRIPTION 0x0008,0x1030
#define TAG_MODALITY 0x0008,0x0060
#define TAG_SERIES_DESCRIPTION 0x0008,0x103E
#define TAG_REFERENCED_FILE_ID 0x0004,0x1500
#define TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET 0x0004,0x1420
#define TAG_NEXT_DIRECTORY_RECORD_OFFSET 0x0004,0x1400

//==============================================================================
// Some handy utility functions
//==============================================================================

std::string left_trim(const std::string &s) {
    std::string ss(s);
    ss.erase(ss.begin(), std::find_if(ss.begin(), ss.end(), std::not1(std::ptr_fun<int, int>(std::isspace))));
    return ss;
}

std::string right_trim(const std::string &s) {
    std::string ss(s);
    ss.erase(std::find_if(ss.rbegin(), ss.rend(), std::not1(std::ptr_fun<int, int>(std::isspace))).base(),
      ss.end());
    return ss;
}

std::string trim(const std::string &s) {
    return left_trim(right_trim(s));
}

//==============================================================================
// This code could be put in a header file somewhere
//==============================================================================

class DICOMDIRReader {
    public:
                                DICOMDIRReader      (void) {}
                                DICOMDIRReader      (const DICOMDIRReader &rhs) = delete;
```

```
                            DICOMDIRReader      (DICOMDIRReader &&rhs) = delete;
    DICOMDIRReader &        operator =          (const DICOMDIRReader &rhs) = delete;
    DICOMDIRReader &        operator =          (DICOMDIRReader &&rhs) = delete;
    virtual                 ~DICOMDIRReader     (void) {}

public:
    struct Common {
        int64_t child_offset;
        int64_t sibling_offset;
    };

    struct Image: public Common {
        std::string path;
    };

    struct Series: public Common {
        std::string modality;
        std::string description;

        std::vector<std::shared_ptr<Image» children;
    };

    struct Study: public Common {
        std::string date;
        std::string description;

        std::vector<std::shared_ptr<Series» children;
    };

    struct Patient: public Common {
        std::string name;
        std::string id;

        std::vector<std::shared_ptr<Study» children;
    };

    struct Other: public Common {
    };

    const std::vector<std::shared_ptr<Patient»&    load        (const std::string &path);

    const std::vector<std::shared_ptr<Patient»&    patients    (void)  {   return _patients;   }

private:

    template <class T>
    std::string     get_string              (const T &ds, const gdcm::Tag &tag)
    {
        std::stringstream strm;
        if (ds.FindDataElement(tag)) {
            auto &de = ds.GetDataElement(tag);
            if (!de.IsEmpty() && !de.IsUndefinedLength())
                de.GetValue().Print(strm);
        }
        return trim(strm.str());
    }

    template <class P, class C, class O>
    void            reassemble_hierarchy    (P &parent_offsets, C &child_offsets, O &other_offsets)
    {
        for (auto &parent : parent_offsets) {
            int64_t sibling_offset;
            auto c = child_offsets[parent.second->child_offset];
            if (!c) {
                auto o = other_offsets[parent.second->child_offset];
                if (!o) {
                    continue;
                } else {
                    sibling_offset = o->sibling_offset;
                }
            } else {
                parent.second->children.push_back(c);
                sibling_offset = c->sibling_offset;
            }

            // Get all siblings
            while (sibling_offset) {
                c = child_offsets[sibling_offset];
                if (!c) {
                    auto o = other_offsets[sibling_offset];
                    if (!o) {
```

```
                            break;
                        } else {
                            sibling_offset = o->sibling_offset;
                        }
                    } else {
                        parent.second->children.push_back(c);
                        sibling_offset = c->sibling_offset;
                    }
                }
            }
        }

        std::vector<std::shared_ptr<Patient» _patients;
};

//==============================================================================
// This code could be put in an implementation file somewhere
//==============================================================================

const std::vector<std::shared_ptr<DICOMDIRReader::Patient»& DICOMDIRReader::load (const std::string &path)
{
    _patients.clear();

    //
    // Read the dataset from the DICOMDIR file
    //

    gdcm::Reader reader;
    reader.SetFileName(path.c_str());
    if(!reader.Read()) {
        throw std::runtime_error("Unable to read file");
    }

    // Retrieve information from file
    auto &file = reader.GetFile();
    auto &data_set = file.GetDataSet();
    auto &file_meta_information = file.GetHeader();

    // Retrieve and check the Media Storage class from file
    gdcm::MediaStorage media_storage;
    media_storage.SetFromFile(file);
    if(media_storage != gdcm::MediaStorage::MediaStorageDirectoryStorage) {
        throw std::runtime_error("This file is not a DICOMDIR");
    }

    auto media_storage_sop_class_uid = get_string(file_meta_information,
      gdcm::Tag(TAG_MEDIA_STORAGE_SOP_CLASS_UID));

    // Make sure we have a DICOMDIR file
    if (media_storage_sop_class_uid != "1.2.840.10008.1.3.10") {
        throw std::runtime_error("This file is not a DICOMDIR");
    }

    //
    // Offset to first item courtesy of Mihail Isakov
    //

    gdcm::VL first_item_offset = 0;
    auto it = data_set.Begin();
    for(; it != data_set.End() && it->GetTag() != gdcm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE); ++it) {
        first_item_offset += it->GetLength<gdcm::ExplicitDataElement>();
    }
    // Tag (4 bytes)
    first_item_offset += it->GetTag().GetLength();
    // VR field
    first_item_offset += it->GetVR().GetLength();
    // VL field
    // For Explicit VR: adventitiously VL field length = VR field length,
    // for SQ 4 bytes:
    // http://dicom.nema.org/medical/dicom/current/output/html/part05.html#table_7.1-1
    first_item_offset += it->GetVR().GetLength();

    //
    // Iterate all data elements
    //

    // For each item in data set
    for(auto data_element : data_set.GetDES()) {

        // Only look at Directory sequence
        if (data_element.GetTag() != gdcm::Tag(TAG_DIRECTORY_RECORD_SEQUENCE))
```

```
        continue;

    auto item_sequence = data_element.GetValueAsSQ();
    auto num_items = item_sequence->GetNumberOfItems();

    //
    // Compute an offset table
    //

    // Start calculation of offset to each item courtesy of Mihail Isakov
    std::vector<int64_t> item_offsets(num_items+1);
    item_offsets[0] = file_meta_information.GetFullLength() + static_cast<int64_t>(first_item_offset);

    //
    // Extract out all of the items
    //

    std::unordered_map<int64_t, std::shared_ptr<Patient» patient_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Study» study_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Series» series_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Image» image_offsets;
    std::unordered_map<int64_t, std::shared_ptr<Other» other_offsets;

    for (uint32_t item_index = 1; item_index <= num_items; ++item_index) {
        auto &item = item_sequence->GetItem(item_index);

        // Add offset for item to offset table
        item_offsets[item_index] = item_offsets[item_index-1] + item.GetLength<gdcm::ExplicitDataElement>();

        // Child offset
        gdcm::Attribute<TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET> child_offset;
        child_offset.SetFromDataElement(item.GetDataElement(gdcm::Tag
(TAG_REFERENCED_LOWER_LEVEL_DIRECTORY_ENTITY_OFFSET)));

        // Sibling offset
        gdcm::Attribute<TAG_NEXT_DIRECTORY_RECORD_OFFSET> sibling_offset;
        sibling_offset.SetFromDataElement(item.GetDataElement(gdcm::Tag
(TAG_NEXT_DIRECTORY_RECORD_OFFSET)));

        // Record Type
        auto record_type = trim(get_string(item, gdcm::Tag (TAG_DIRECTORY_RECORD_TYPE)));

        // std::cout « "record_type " « record_type « " at " « item_offsets[item_index-1] « std::endl;
        // std::cout « " child_offset " « child_offset.GetValue() « std::endl;
        // std::cout « " sibling_offset " « sibling_offset.GetValue() « std::endl;

        // Extract patient information
        if (record_type == "PATIENT") {
            auto patient = std::make_shared<Patient>();
            patient->name = get_string(item, gdcm::Tag (TAG_PATIENTS_NAME));
            patient->id = get_string(item, gdcm::Tag (TAG_PATIENT_ID));

            patient->child_offset = child_offset.GetValue();
            patient->sibling_offset = sibling_offset.GetValue();
            patient_offsets[item_offsets[item_index-1]] = patient;

        // Extract study information
        } else if (record_type == "STUDY") {
            auto study = std::make_shared<Study>();
            study->date = get_string(item, gdcm::Tag (TAG_STUDY_DATE));
            study->description = get_string(item, gdcm::Tag (TAG_STUDY_DESCRIPTION));

            study->child_offset = child_offset.GetValue();
            study->sibling_offset = sibling_offset.GetValue();
            study_offsets[item_offsets[item_index-1]] = study;

        // Extract series information
        } else if (record_type == "SERIES") {
            auto series = std::make_shared<Series>();
            series->modality = get_string(item, gdcm::Tag (TAG_MODALITY));
            series->description = get_string(item, gdcm::Tag (TAG_SERIES_DESCRIPTION));

            series->child_offset = child_offset.GetValue();
            series->sibling_offset = sibling_offset.GetValue();
            series_offsets[item_offsets[item_index-1]] = series;

        // Extract image information
        } else if (record_type == "IMAGE") {
            auto image = std::make_shared<Image>();
```

```
                image->path = get_string(item, gdcm::Tag (TAG_REFERENCED_FILE_ID));

                image->child_offset = child_offset.GetValue();
                image->sibling_offset = sibling_offset.GetValue();
                image_offsets[item_offsets[item_index-1]] = image;
            } else {
                auto other = std::make_shared<Other>();

                other->child_offset = child_offset.GetValue();
                other->sibling_offset = sibling_offset.GetValue();
                other_offsets[item_offsets[item_index-1]] = other;
            }
        }

        // Check validity
        if (patient_offsets.size() == 0)
            throw std::runtime_error("Unable to find patient record");

        reassemble_hierarchy(series_offsets, image_offsets, other_offsets);
        reassemble_hierarchy(study_offsets, series_offsets, other_offsets);
        reassemble_hierarchy(patient_offsets, study_offsets, other_offsets);

        // Set the new root
        for (auto &patient : patient_offsets) {
            _patients.push_back(patient.second);
        }
    }

    return _patients;
}

//==============================================================================
// Quick test
//==============================================================================

int main(int argc, const char *argv[]) {
    DICOMDIRReader reader;

    try {
        if (argc != 2)
            throw std::runtime_error("Wrong number of arguments");

        auto &patients = reader.load(argv[1]);

        for (auto &patient : patients) {

            std::cout « "PATIENT" « std::endl;
            std::cout « "NAME: " « patient->name « std::endl;
            std::cout « "ID: " « patient->id « std::endl;

            int x = 0;
            for (auto &study : patient->children) {
                std::cout « "    STUDY" « std::endl;
                std::cout « "    DESCRIPTION: " « study->description « std::endl;
                std::cout « "    DATE: " « study->date « std::endl;

                for (auto &series : study->children) {
                    x+=1;
                    std::cout « "        SERIES " « x « std::endl;
                    std::cout « "        DESCRIPTION: " « series->description « std::endl;
                    std::cout « "        MODALITY: " « series->modality « std::endl;

                    for (auto &image : series->children) {
                        std::cout « "            IMAGE PATH: " « image->path « std::endl;
                    }
                }
            }
        }
    }
    catch (...) {
        // TODO handle this
        return EXIT_FAILURE;
    }

    return EXIT_SUCCESS;
}
```

14.89 ReadAndPrintAttributes.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This small example will show how one can read and print
 * a DICOM Attribute using different technique (by tag or by name)
 */

#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmAttribute.h"
#include "gdcmStringFilter.h"

#include <iostream>

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " input.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];

  // Instantiate the reader:
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Could not read: " « filename « std::endl;
    return 1;
    }

  // The output of gdcm::Reader is a gdcm::File
  gdcm::File &file = reader.GetFile();

  // the dataset is the the set of element we are interested in:
  gdcm::DataSet &ds = file.GetDataSet();

  const gdcm::Global& g = gdcm::Global::GetInstance();
  const gdcm::Dicts &dicts = g.GetDicts();
  const gdcm::Dict &pubdict = dicts.GetPublicDict();

  using namespace gdcm;

  // In this example we will show why using name to lookup attribute can be
  // dangerous.
  Tag tPatientName(0x0,0x0);
  //const DictEntry &de1 =
  pubdict.GetDictEntryByName("Patient Name", tPatientName);

  std::cout « "Found: " « tPatientName « std::endl;

  // Indeed the attribute could not be found. Since DICOM 2003, Patient Name
  // has become Patient's Name.

  Tag tPatientsName;
  //const DictEntry &de2 =
  pubdict.GetDictEntryByName("Patient's Name", tPatientsName);

  std::cout « "Found: " « tPatientsName « std::endl;

  // Let's try to read an arbitrary DICOM Attribute:
  Tag tDoseGridScaling;
  //const DictEntry &de3 =
```

```
  pubdict.GetDictEntryByName("Dose Grid Scaling", tDoseGridScaling);

  std::cout « "Found: " « tDoseGridScaling « std::endl;

  if( ds.FindDataElement( tDoseGridScaling ) )
    {
    gdcm::StringFilter sf;
    sf.SetFile(file);
    std::cout « "Attribute Value as String: " « sf.ToString( tDoseGridScaling ) « std::endl;

    // Let's check the name again:
    std::pair<std::string, std::string> pss
      = sf.ToStringPair( tDoseGridScaling );
    std::cout « "Attribute Name Checked: " « pss.first « std::endl;
    std::cout « "Attribute Value (string): " « pss.second « std::endl;

    //const DataElement &dgs = ds.GetDataElement( tDoseGridScaling );

    // Let's assume for a moment we knew the tag number:
    Attribute<0x3004,0x000e> at;
    assert( at.GetTag() == tDoseGridScaling );
    at.SetFromDataSet( ds );
    // For the sake of long term maintenance, we will not write
    // that this particular attribute is stored as a double. What if
    // a user made a mistake. It is much safer to rely on GDCM internal
    // mechanism to deduce the VR::DS type (represented as a ieee double)
    Attribute<0x3004,0x000e>::ArrayType v = at.GetValue();
    std::cout « "DoseGridScaling=" « v « std::endl;
    }

  return 0;
}
```

14.90 ReadExplicitLengthSQIVR.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmByteValue.h"
#include "gdcmDataSet.h"
#include "gdcmImplicitDataElement.h"
#include "gdcmPrivateTag.h"
#include "gdcmReader.h"
#include "gdcmSequenceOfItems.h"

using namespace gdcm;

int main(int argc, char *argv[])
{
  if ( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader r;
  r.SetFileName( filename );
  r.Read();


  //gdcm::PrivateTag pt(0xe1,0x42,"ELSCINT1");
  //gdcm::Tag pt(0x88,0x200);
  gdcm::Tag pt(0x8,0x1140);
  DataSet &ds = r.GetFile().GetDataSet();
  const DataElement &de = ds.GetDataElement( pt );

  std::cout « de « std::endl;
  const ByteValue *bv = de.GetByteValue();
  SmartPointer<SequenceOfItems> sqi = new SequenceOfItems;
  sqi->SetLength( bv->GetLength() );
```

```
  std::stringstream ss;
  ss.str( std::string( bv->GetPointer(), bv->GetLength() ) );
  sqi->Read<ImplicitDataElement,SwapperNoOp>( ss );

  std::cout « *sqi « std::endl;

  return 0;
}
```

14.91 ReadGEMSSDO.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmDataElement.h"
#include "gdcmPrivateTag.h"

#include <iostream>
#include <string>

using namespace gdcm;

struct SDOElement
{
  typedef std::vector<std::string>::size_type SizeType;
  const char *GetData(SizeType index) const {
    return Data[index].c_str();
  }
  SizeType GetNumberOfData() const {
    return Data.size();
  }
  void SetData(SizeType index, const char *data) {
    Data[index] = data;
  }
  const char *GetDataFormat() const {
    return DataFormat.c_str();
  }
  void SetDataFormat(const char *dataformat, SizeType num) {
    DataFormat = dataformat;
    Data.resize( num );
  }
  void Print( std::ostream &os ) const {
    os « DataFormat « ":" « std::endl;
    std::vector<std::string>::const_iterator it = Data.begin();
    size_t s = 0;
    for( ; it != Data.end(); ++it )
      {
      os « "  (" « s++ « ") " « *it « std::endl;
      }
  }
private:
  std::string DataFormat;
  std::vector<std::string> Data;
};

class SDOHeader
{
public:
  typedef std::vector<SDOElement> SDOElements;
  typedef SDOElements::size_type SizeType;
  SizeType GetNumberOfSDOElements() const {
  return InternalSDODataSet.size();
  }
  void AddSDOElement(SDOElement const &sdoelement) {
  InternalSDODataSet.push_back( sdoelement );
```

```
  }
  const SDOElement &GetSDOElement(SizeType index) const {
  return InternalSDODataSet[index];
  }
  const SDOElement &GetSDOElementByName(const char *) const {
    return InternalSDODataSet[0];
  }
  void LoadFromAttributes(std::string const &s1, std::string const &s2)
    {
  std::string tok;
  std::string tok2;
  std::stringstream strstr(s1);
  std::stringstream strstr2(s2);

  SDOElement element;
  // Do format
  size_t count = 0;
  while ( std::getline ( strstr2, tok, '\\' ) )
    {
    //std::cout « tok « " ";
    std::getline ( strstr2, tok2, '\\' );
    //std::cout « tok2 « std::endl;
    count += atoi( tok2.c_str() );
    element.SetDataFormat( tok.c_str(), atoi( tok2.c_str() ) );
    for( size_t t = 0; t < element.GetNumberOfData(); ++t )
      {
      std::getline ( strstr, tok, '\\' );
      element.SetData(t, tok.c_str() );
      }
    AddSDOElement( element );
    }
  //while ( std::getline ( strstr, tok, '^' ) )
//  while ( std::getline ( strstr, tok, '\\' ) )
//    {
//    std::cout « tok « std::endl;
//    count++;
//    }
//  std::cout « "Count: " « count « std::endl;
//  count = 0;

//  std::cout « "Count: " « count « std::endl;

    }
  void Print( std::ostream &os ) const {
    SDOElements::const_iterator it = InternalSDODataSet.begin();
    for( ; it != InternalSDODataSet.end(); ++it )
      {
      it->Print ( os );
      }
  }
private:
  SDOElements InternalSDODataSet;
};

bool sdo_decode( DataElement const &stringdata, DataElement const &stringdataformat )
{
  const char *sd = stringdata.GetByteValue()->GetPointer();
  const size_t len_sd = stringdata.GetByteValue()->GetLength();

  std::string s1 = std::string( sd, len_sd );

  const char *sdf = stringdataformat.GetByteValue()->GetPointer();
  const size_t len_sdf = stringdataformat.GetByteValue()->GetLength();

  std::string s2 = std::string( sdf, len_sdf );

//  std::cout « s1 « std::endl;
//  std::cout « s2 « std::endl;

  SDOHeader header;
  header.LoadFromAttributes( s1, s2 );

  header.Print( std::cout );

  return true;
}

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
```

```
    std::cerr « argv[0] « " input.dcm" « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  File &file = reader.GetFile();
  DataSet &ds = file.GetDataSet();

  // StringData (0033,xx1F) 3 "GEMS_GENIE_1" List of SDO parameters stored as
  // list of strings
  const PrivateTag tstringdata(0x33,0x1f,"GEMS_GENIE_1");
  // StringDataFormat (0033,xx23) 3 "GEMS_GENIE_1" Format of string parameters;
  // contains information about name and number of strings in list
  const PrivateTag tstringdataformat(0x33,0x23,"GEMS_GENIE_1");

  if( !ds.FindDataElement( tstringdata ) ) return 1;
  const DataElement& stringdata = ds.GetDataElement( tstringdata );
  if( !ds.FindDataElement( tstringdataformat ) ) return 1;
  const DataElement& stringdataformat = ds.GetDataElement( tstringdataformat );

  sdo_decode( stringdata, stringdataformat );

  return 0;
}
```

14.92 ReadMultiTimesException.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// The intention of this sample program is to provoke bad_alloc exceptions in gdcm code

#include "gdcmImageReader.h"

int main(int argc, char* argv[])
{
  // We pre-allocate some memory (about 1Gb) to help the issue to show up earlier
  char *dummyBuffer = new char[1024*1024*1100]; (void)dummyBuffer;
  // Check the number of parameters given
  if (argc < 3)
    {
    std::cerr « "Usage: " « argv[0] « " Filename numberOfTries" « std::endl;
    return 1;
    }

  std::cout « "We are going to read the file: " « argv[1] « " " « argv[2] « " times" « std::endl;
  // We hold the pointers in an array to avoid the memory to be released
  // We read the input file n-times
  for (int i = 0; i < atoi(argv[2]); ++i)
    {
    gdcm::ImageReader reader;
    std::cout « "Reading try: " « i « std::endl;
    // Read files
    reader.SetFileName(argv[1]);
    try
      {
      reader.Read();
      gdcm::Image & img = reader.GetImage();
      unsigned long len = img.GetBufferLength();
      char *buffer = new char[ len ];
```

```
      img.GetBuffer( buffer ); // do NOT de-allocate buffer !
      }
    catch (std::bad_alloc &ba)
      {
      (void)ba;
      std::cerr « "BAD ALLOC Exception caught!" « std::endl;
      }
    catch (...)
      {
      std::cerr « "Exception caught!" « std::endl;
      }
    }

  return 0;
}
```

14.93 ReadUTF8QtDir.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * GDCM API expect a const char * as input for SetFileName
 * In order to use this API from Qt, here is a simple test that
 * shows how to do it in a portable manner:
 *
 * http://doc.qt.nokia.com/latest/qdir.html#navigation-and-directory-operations
 */

#include "gdcmReader.h"
#include "gdcmDirectory.h"

#include <QDir>
#include <QString>
#include <QCoreApplication>

#include <string>
#include <fstream>

#include <stdio.h> // fopen

static int TestBothFuncs(const char *info , const char *ba_str)
{
  int res = 0;
  FILE *f = fopen( ba_str, "r" );
  if( f )
    {
    std::cout « info « " fopen: " « ba_str « std::endl;
    fclose(f);
    ++res;
    }
  gdcm::Reader reader;
  std::ifstream is( ba_str, std::ios::binary );
  if( is.is_open() )
    {
    std::cout « info « " is_open: " « ba_str « std::endl;
    ++res;
    }
  reader.SetStream( is );
  if( reader.CanRead() == true )
    {
    std::cout « info « " SetStream/CanRead:" « ba_str « std::endl;
    ++res;
    }
  is.close();
  reader.SetFileName( ba_str );
```

```
  if( reader.CanRead() == true )
    {
    std::cout « info « " SetFileName/CanRead:" « ba_str « std::endl;
    ++res;
    }
  return 4 - res;
}

static int scanFolder(const char dirname[])
{
  int res = 0;
  gdcm::Directory dir;
  unsigned int nfiles = dir.Load( dirname, true );
  const gdcm::Directory::FilenamesType &filenames = dir.GetFilenames();

  for( unsigned int i = 0; i < nfiles; ++i )
    {
    const char *ba_str = filenames[i].c_str();
    res += TestBothFuncs("GDCM",ba_str);
    }
  return res;
}

static int scanFolderQt(QDir const &dir, QStringList& files)
{
  int res = 0;
  QFileInfoList children = dir.entryInfoList(QDir::AllEntries|QDir::NoDotAndDotDot);
  for ( int i=0; i<children.count(); i++ ) {
    QFileInfo file = children.at(i);
    if ( file.isDir() == true ) {
      res += scanFolderQt(QDir(file.absoluteFilePath()), files);
      continue;
    }
    // Convert back from the internal representation to 8bits
    // toLocal8Bit() returns by copy. Need to store explicitly the QByteArray
    QByteArray str = file.absoluteFilePath().toLocal8Bit();
    const char *ba_str1 = str.constData();
    res += TestBothFuncs("QString", ba_str1);
    }
  return res;
}

int main(int argc, char *argv[])
{
  // very important:
  QCoreApplication qCoreApp( argc , argv );
  if( argc < 2 )
    {
    std::cerr « argv[0] « " dir " « std::endl;
    return 1;
    }

  int res = 0;
  const char *dirname = argv[1];
  res += scanFolder( dirname );

  QDir dir( QString::fromLocal8Bit(dirname) );
  QStringList files;
  res += scanFolderQt( dir, files);

  if( res )
    std::cerr « "Problem with UTF-8" « std::endl;
  else
    std::cerr « "Success with UTF-8" « std::endl;

  return res;
}
```

14.94 SimpleScanner.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
```

```
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Simple example to show how to use Scanner API.
 * It exposes the three different cases:
 *  - DICOM Attribute is present and has a value
 *  - DICOM Attribute is present and has no value
 *  - DICOM Attribute is not present at all
 * It also shows the purpose of the function 'IsKey' to detect whether or
 * not the file has been read by the gdcm::Scanner. Technically most of the time
 * if a file is not a 'Key' this is because it is not a DICOM file. You need to use
 * gdcm::System::FileExists to decide whether or not the file actually exist on the disk.
 *
 * It was tested on this particular image:
 * ./SimpleScanner gdcmData/012345.002.050.dcm
 */

#include "gdcmStrictScanner.h"
#include "gdcmSimpleSubjectWatcher.h"
#include "gdcmFileNameEvent.h"

class MyFileWatcher : public gdcm::SimpleSubjectWatcher
{
public:
  MyFileWatcher(gdcm::Subject * s, const char *comment = ""):
    gdcm::SimpleSubjectWatcher(s,comment){}
  void ShowFileName(gdcm::Subject *, const gdcm::Event &evt) override
    {
    const gdcm::FileNameEvent &pe = dynamic_cast<const gdcm::FileNameEvent&>(evt);
    const char *fn = pe.GetFileName();
    std::cout « "FileName: " « fn « " FileSize: " « gdcm::System::FileSize( fn ) « std::endl;
    }
};

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char filename_invalid[] = "this is a file that may not exist on this disk.dcm";


  gdcm::SmartPointer<gdcm::StrictScanner> sp = new gdcm::StrictScanner;
  gdcm::StrictScanner &s = *sp;
  //gdcm::SimpleSubjectWatcher w(&s, "TestFileName" );
  MyFileWatcher w(&s, "TestFileName" );

  const gdcm::Tag tag_array[] = {
    gdcm::Tag(0x8,0x50),
    gdcm::Tag(0x8,0x51),
    gdcm::Tag(0x8,0x60),
    gdcm::Tag(0x8,0x80),
  };
  s.AddTag( tag_array[0] );
  s.AddTag( tag_array[1] );
  s.AddTag( tag_array[2] );
  s.AddTag( tag_array[3] );

  gdcm::Directory::FilenamesType filenames;
  filenames.push_back( filename );
  filenames.push_back( filename_invalid );

  if( !s.Scan( filenames ) )
    {
    return 1;
    }

  //s.Print( std::cout );


  for(gdcm::Directory::FilenamesType::const_iterator it = filenames.begin();
    it != filenames.end(); ++it )
    {
    if( s.IsKey( it->c_str() ) )
```

```
      {
      std::cout « "INFO:" « it->c_str() « " is a proper Key for the Scanner (this is a DICOM file)" « std::endl;
      }
    else
      {
      std::cout « "INFO:" « it->c_str() « " is not a proper Key for the Scanner (this is either not a DICOM file
      or file does not exist)" « std::endl;
      }
    }

  gdcm::StrictScanner::TagToValue const &ttv = s.GetMapping(filename);

  const gdcm::Tag *ptag = tag_array;
  for( ; ptag != tag_array + 3; ++ptag )
    {
    gdcm::StrictScanner::TagToValue::const_iterator it = ttv.find( *ptag );
    if( it != ttv.end() )
      {
      std::cout « *ptag « " was properly found in this file" « std::endl;
      // it contains a pair of value. the first one is the actual tag, so the following is always true:
      //  *ptag == it->first
      // The second part is the actual value (stored as RAW strings). You will have to reinterpret this string
      // if VR for *ptag is not VR::VRASCII !
      const char *value = it->second;
      if( *value )
        {
        std::cout « "  It has the value: " « value « std::endl;
        }
      else
        {
        std::cout « "  It has no value (empty)" « std::endl;
        }
      }
    else
      {
      std::cout « "Sorry " « *ptag « " could not be found in this file" « std::endl;
      }
    }

  return 0;
}
```

14.95 SortImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 */
#include "gdcmSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"


bool mysort(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  //gdcm::Attribute<0x0020,0x0013> at1; // Instance Number
  gdcm::Attribute<0x0018,0x1060> at1; // Trigger Time
  gdcm::Attribute<0x0020,0x0032> at11; // Image Position (Patient)
  at1.Set( ds1 );
  at11.Set( ds1 );
  //gdcm::Attribute<0x0020,0x0013> at2;
  gdcm::Attribute<0x0018,0x1060> at2;
  gdcm::Attribute<0x0020,0x0032> at22;
  at2.Set( ds2 );
```

```
  at22.Set( ds2 );
  if( at11 == at22 )
    {
    return at1 < at2;
    }
  return at11 < at22;
}

bool mysort_part1(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0018,0x1060> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0018,0x1060> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

bool mysort_part2(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0020,0x0032> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0032> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

// technically all files are in the same Frame of Reference, so this function
// should be a no-op
bool mysort_dummy(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0020,0x0052> at1; // FrameOfReferenceUID
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0052> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

int main(int argc, char *argv[])
{
  if (argc < 2 ) return 1;
  const char *dirname = argv[1];
  gdcm::Directory dir;
  unsigned int nfiles = dir.Load( dirname );

  dir.Print( std::cout );

  gdcm::Sorter sorter;
  sorter.SetSortFunction( mysort );
  sorter.Sort( dir.GetFilenames() );

  std::cout « "Sorter:" « std::endl;
  sorter.Print( std::cout );

  gdcm::Sorter sorter2;
  sorter2.SetSortFunction( mysort_part1 );
  sorter2.StableSort( dir.GetFilenames() );
  sorter2.SetSortFunction( mysort_part2 );
  sorter2.StableSort( sorter2.GetFilenames() ); // IMPORTANT
  sorter2.SetSortFunction( mysort_dummy );
  sorter2.StableSort( sorter2.GetFilenames() ); // IMPORTANT

  std::cout « "Sorter2:" « std::endl;
  sorter2.Print( std::cout );

  gdcm::Scanner s;
  s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
  //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
  s.Scan( dir.GetFilenames() );

  //s.Print( std::cout );

  // Count how many different IPP there are:
  const gdcm::Scanner::ValuesType &values = s.GetValues();
  size_t nvalues = values.size();
  std::cout « "There are " « nvalues « " different type of values" « std::endl;

  //std::cout « "nfiles=" « nfiles « std::endl;
  if( nfiles % nvalues != 0 )
    {
    std::cerr « "Impossible: this is a not a proper series" « std::endl;
    return 1;
```

```
    }
  std::cout « "Series is composed of " « (nfiles/nvalues) « " different 3D volumes" « std::endl;

  return 0;
}
```

14.96 StreamImageReaderTest.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// This work was realised during the GSOC 2011 by Manoj Alwani

#include "gdcmStreamImageReader.h"
#include "gdcmFileMetaInformation.h"
#include "gdcmSystem.h"
#include "gdcmFilename.h"
#include "gdcmByteSwap.h"
#include "gdcmTrace.h"
#include "gdcmTesting.h"
#include "gdcmImageHelper.h"
#include "gdcmImageReader.h"
#include "gdcmImage.h"
#include "gdcmMediaStorage.h"
#include "gdcmRAWCodec.h"
#include "gdcmJPEGLSCodec.h"
#include "gdcmUIDGenerator.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"

bool StreamImageRead(gdcm::StreamImageWriter & theStreamWriter,
  const char* filename, const char* outfilename, int resolution)
{
  gdcm::StreamImageReader reader;

  reader.SetFileName( filename );

  if (!reader.ReadImageInformation())
    {
    std::cerr « "unable to read image information" « std::endl;
    return 1; //unable to read tags as expected.
    }
  //let's be tricky; each image will be read in portions, first the top half, then the bottom
  //that way, we can test how the stream handles fragmentation of the data
  //we could also loop this to get various different size combinations, but I'm not sure
  //that's useful, yet.
  std::vector<unsigned int> extent =
    gdcm::ImageHelper::GetDimensionsValue(reader.GetFile());
  // std::cout « extent[0];
  //at this point, these values aren't used, but may be in the future
  //unsigned short xmin = 0;
  //unsigned short xmax = extent[0];
  //unsigned short ymin = 0;
  //unsigned short ymax = extent[1];
  //unsigned short zmin = 0;
  //unsigned short zmax = extent[2];

  std::cout« "\n Row: "«extent[0] «"\n Col :"« extent[1]« "\n Resolution :"« extent[2] « std::endl;

  int a =1;
  for (int i=1; i<=(extent[2]-resolution);++i)
```

```
    a = a*2;

  reader.DefinePixelExtent(0, extent[0]/a, 0, extent[1]/a, resolution-1, resolution);

  unsigned long len = reader.DefineProperBufferLength();
  char* finalBuffer = new char[len];
  memset(finalBuffer, 0, sizeof(char)*len);

  if (reader.CanReadImage())
    {
    bool result = reader.Read(finalBuffer, len);
    if( !result )
      {
      std::cout « "res2 failure:" « filename « std::endl;
      delete [] finalBuffer;
      return 1;
      }
    else
      {
      std::cout« "Able to read";
      }
    }
  else
    {
    std::cerr« "Not able to put in buffer"« std::endl;
    }
/*
    //now, read in smaller buffer extents
    reader.DefinePixelExtent(xmin, xmax, ymin, ymax);
    len = reader.DefineProperBufferLength();

    char* buffer = new char[len];
    bool res2 = reader.Read(buffer, len);
    if( !res2 ){
      std::cerr « "res2 failure:" « filename « std::endl;
      return 1;
    }
    //copy the result into finalBuffer
    memcpy(finalBuffer, buffer, len);

    //now read the next half of the image
    ymin = ymax;
    ymax = extent[1];

    reader.DefinePixelExtent(xmin, xmax, ymin, ymax);

    //std::cerr « "Success to read image from file: " « filename « std::endl;
    unsigned long len2 = reader.DefineProperBufferLength();

    char* buffer2 = new char[len2];
    bool res3 = reader.Read(buffer2, len2);
    if( !res3 ){
      std::cerr « "res3 failure:" « filename « std::endl;
      return 1;
    }
    //copy the result into finalBuffer
    memcpy(&(finalBuffer[len]), buffer2, len2);

    delete [] buffer;
    delete [] buffer2;
*/

  gdcm::Writer w;
  gdcm::File &file = w.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  file.GetHeader().SetDataSetTransferSyntax( gdcm::TransferSyntax::ExplicitVRLittleEndian );

  gdcm::UIDGenerator uid;
  gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
  de.SetVR( gdcm::VR::UI );
  const char *u = uid.Generate();
  de.SetByteValue( u, strlen(u) );
  ds.Insert( de );

  gdcm::DataElement de1( gdcm::Tag(0x8,0x16) );
  de1.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms( gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage );
  de1.SetByteValue( ms.GetString(), strlen(ms.GetString()));
```

```
  ds.Insert( de1 );

  const char mystr[] = "MONOCHROME2 ";
  gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
  //de.SetTag(gdcm::Tag(0x28,0x04));
  de2.SetVR( gdcm::VR::CS );
  de2.SetByteValue(mystr, strlen(mystr));
  ds.Insert( de2 );

   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
   ds.Insert( Number_Of_Frames.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0010> row = {extent[0]/a};//
   ds.Insert( row.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0011> col = {extent[1]/a};//
   ds.Insert( col.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0100> at = {8};
   ds.Insert( at.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0002> at1 = {1};//
   ds.Insert( at1.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0101> at2 = {8};
   ds.Insert( at2.GetAsDataElement() );

   gdcm::Attribute<0x0028,0x0102> at3 = {7};
   ds.Insert( at3.GetAsDataElement() );
    /*
   ds1.Remove( gdcm::Tag(0x0028,0x0008) );

   gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
   ds1.Insert( Number_Of_Frames.GetAsDataElement() );
*/
   theStreamWriter.SetFile(file);

   if (!theStreamWriter.WriteImageInformation())
     {
     std::cerr « "unable to write image information" « std::endl;
     return 1; //the CanWrite function should prevent getting here, else,
     //that's a test failure
     }
    std::vector<unsigned int> extent1 = gdcm::ImageHelper::GetDimensionsValue(file);

    unsigned short xmax = extent1[0];
    unsigned short ymax = extent1[1];
    unsigned short theChunkSize = 1;
    unsigned short ychunk = extent1[1]/theChunkSize; //go in chunk sizes of theChunkSize
    unsigned short zmax = 1;

    std::cout« "\n Row: "«extent1[0] «"\n Col :"« extent1[1]« "\n Resolution :"« extent1[2] « std::endl;

    if (xmax == 0 || ymax == 0)
      {
      std::cerr « "Image has no size, unable to write zero-sized image." « std::endl;
      return 0;
      }

    int z, y, nexty;
    unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
    //the bytes sequentially.  So, store how far you got in the buffer with each iteration.

    for (z = 0; z < zmax; ++z){
      for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout « "\n" «len;
        char* finalBuffer1 = new char[len];
        memcpy(finalBuffer1, &(finalBuffer[prevLen]), len);
        std::cout « "\nable to write";

        if (!theStreamWriter.Write(finalBuffer1, len)){
          std::cerr « "writing failure:" « "output.dcm" « " at y = " « y « " and z= " « z « std::endl;
          delete [] finalBuffer1;
          delete [] finalBuffer;
          return 1;
        }
        delete [] finalBuffer1;
```

```
        prevLen += len;
      }
    }
   delete [] finalBuffer;
   std::cout « "all is set";

  return true;
}


int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm Resolution" « std::endl;
    return 1;
    }

  const char *filename = argv[1];
  const char *outfilename = argv[2];
  char *res = argv[3];

  int resolution = atoi(res);

  gdcm::StreamImageWriter theStreamWriter;

  std::ofstream of;
  of.open( outfilename, std::ios::out | std::ios::binary );
  theStreamWriter.SetStream(of);

  // else
  // First of get rid of warning/debug message
  gdcm::Trace::DebugOn();
  gdcm::Trace::WarningOn();

  if(!StreamImageRead( theStreamWriter, filename, outfilename, resolution))
    return 1;

  uint16_t firstTag1 =  0xfffe;
  uint16_t secondTag1 = 0xe0dd;
  uint32_t thirdTag1 =  0x00000000;
  //uint16_t fourthTag1 = 0xffff;
  const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
  char* tmpBuffer2 = new char[theBufferSize1];
  memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
  memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
  memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
  //memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
  assert( of && !of.eof() && of.good() );
  of.write(tmpBuffer2, theBufferSize1);
  of.flush();
  assert( of );

  return 0;
}
```

14.97 TemplateEmptyImage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmFileStreamer.h"
#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmImageRegionReader.h"
#include "gdcmImageHelper.h"
```

```
#include "gdcmWriter.h"
#include "gdcmImageWriter.h"
#include "gdcmTagKeywords.h"
#include "gdcmUIDGenerator.h"

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  const char * filename = argv[1];
  gdcm::ImageRegionReader irr;
  irr.SetFileName( filename );
  const bool b3 = irr.ReadInformation();
  std::cout « b3 « std::endl;
  gdcm::Image & img = irr.GetImage();
  std::cout « img « std::endl;
  //  const gdcm::Region & r = irr.GetRegion();
  //  std::cout « r « std::endl;
  gdcm::ImageWriter w;
  gdcm::File & file = w.GetFile();
  gdcm::DataSet & ds = file.GetDataSet();

  gdcm::UIDGenerator uid;
  namespace kwd = gdcm::Keywords;
  kwd::FrameOfReferenceUID frameref;
  frameref.SetValue( uid.Generate() );
  // ContentDate
  char date[22];
  const size_t datelen = 8;
  int res = gdcm::System::GetCurrentDateTime(date);
  (void)res;
  kwd::ContentDate contentdate;
  // Do not copy the whole cstring:
  contentdate.SetValue( gdcm::DAComp( date, datelen ) );
  ds.Insert( contentdate.GetAsDataElement() );
  // ContentTime
  const size_t timelen = 6 + 1 + 6; // time + milliseconds
  kwd::ContentTime contenttime;
  // Do not copy the whole cstring:
  contenttime.SetValue( gdcm::TMComp(date+datelen, timelen) );
  ds.Insert( contenttime.GetAsDataElement() );
  gdcm::MediaStorage ms0 = w.ComputeTargetMediaStorage();
  std::cout « ms0 « std::endl;
  kwd::SeriesNumber seriesnumber = { 1 };
  kwd::InstanceNumber instancenum = { 1 };
  kwd::StudyID studyid = { "St1" };
  kwd::PatientID patientid = { "P1" };
  kwd::SOPClassUID sopclassuid;
  kwd::PositionReferenceIndicator pri;
  //kwd::Laterality lat;
  //kwd::BodyPartExamined bodypartex = { "HEAD" };
  kwd::BodyPartExamined bodypartex = { "ANKLE" };
  kwd::PatientOrientation pator;
  kwd::BurnedInAnnotation bia = { "NO" };
  kwd::ConversionType convtype = { "SYN" };
  kwd::PresentationLUTShape plutshape = { "IDENTITY" }; // MONOCHROME2
  // gdcm will pick the Word in case Byte class is not compatible:
  gdcm::MediaStorage ms = gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage;
  sopclassuid.SetValue( ms.GetString() );
  ds.Insert( instancenum.GetAsDataElement() );
  ds.Insert( sopclassuid.GetAsDataElement() );
  ds.Insert( seriesnumber.GetAsDataElement() );
  ds.Insert( patientid.GetAsDataElement() );
  ds.Insert( studyid.GetAsDataElement() );
  ds.Insert( frameref.GetAsDataElement() );
  ds.Insert( pri.GetAsDataElement() );
  //ds.Insert( lat.GetAsDataElement() );
  ds.Insert( bodypartex.GetAsDataElement() );
  ds.Insert( pator.GetAsDataElement() );
  ds.Insert( bia.GetAsDataElement() );
  ds.Insert( convtype.GetAsDataElement() );
  ds.Insert( plutshape.GetAsDataElement() );
  //    gdcm::MediaStorage ms1 = w.ComputeTargetMediaStorage();
  //    std::cout « ms1 « std::endl;
  std::cout « ds « std::endl;
  gdcm::PixelFormat & pf = img.GetPixelFormat();
  pf.SetPixelRepresentation(0); // always overwrite
  img.SetSlope(1);
  img.SetIntercept(0);
  w.SetImage( img );
  w.SetFileName( "TemplateImage.dcm" );
  if( !w.Write() )
```

```
  {
    return 1;
  }

  return 0;
}
```

14.98 TraverseModules.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 */

#include "gdcmDefs.h"
#include "gdcmGlobal.h"
#include "gdcmIODs.h"
#include "gdcmIOD.h"
#include "gdcmMacros.h"
#include "gdcmIODEntry.h"
#include "gdcmModules.h"
#include "gdcmModule.h"
#include "gdcmAnonymizer.h"
#include "gdcmDicts.h"

int main(int , char *[])
{
  using namespace gdcm;
  static Global &g = Global::GetInstance();

  if( !g.LoadResourcesFiles() )
    {
    return 1;
    }

  static const Defs &defs = g.GetDefs();
  static const Modules &modules = defs.GetModules();
  static const IODs &iods = defs.GetIODs();
  static const Macros &macros = defs.GetMacros();
  static const Dicts &dicts = g.GetDicts();

  std::vector<Tag> tags = gdcm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes();
  for( std::vector<Tag>::const_iterator tit = tags.begin(); tit != tags.end(); ++tit )
    {
    const Tag &tag = *tit;
    const DictEntry &dictentry = dicts.GetDictEntry(tag);
    std::cout « "Processing Attribute: " « tag « " " « dictentry « std::endl;

    IODs::IODMapTypeConstIterator it = iods.Begin();
    for( ; it != iods.End(); ++it )
      {
      const IODs::IODName &name = it->first;
      const IOD &iod = it->second;

      const size_t niods = iod.GetNumberOfIODs();
      // Iterate over each iod entry in order:
      for(unsigned int idx = 0; idx < niods; ++idx)
        {
        const IODEntry &iodentry = iod.GetIODEntry(idx);
        const char *ref = iodentry.GetRef();
        //Usage::UsageType ut = iodentry.GetUsageType();

        const Module &module = modules.GetModule( ref );
        if( module.FindModuleEntryInMacros(macros, tag ) )
```

```
          {
          const ModuleEntry &module_entry = module.GetModuleEntryInMacros(macros,tag);
          Type type = module_entry.GetType();
          std::cout « "IOD Name: " « name « std::endl;
          std::cout « "Type: " « type « std::endl;
          }
        }

      }
    }

  return 0;
}
```

14.99 VolumeSorter.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 */
#include "gdcmSorter.h"
#include "gdcmIPPSorter.h"
#include "gdcmScanner.h"
#include "gdcmDataSet.h"
#include "gdcmAttribute.h"
#include "gdcmTesting.h"


bool mysort1(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0020,0x000d> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x000d> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

bool mysort2(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  gdcm::Attribute<0x0020,0x000e> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x000e> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

bool mysort3(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  // This is a floating point number is the comparison ok ?
  gdcm::Attribute<0x0020,0x0037> at1;
  at1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0037> at2;
  at2.Set( ds2 );
  return at1 < at2;
}

bool mysort4(gdcm::DataSet const & ds1, gdcm::DataSet const & ds2 )
{
  // Do the IPP sorting here
  gdcm::Attribute<0x0020,0x0032> ipp1;
  gdcm::Attribute<0x0020,0x0037> iop1;
  ipp1.Set( ds1 );
  iop1.Set( ds1 );
  gdcm::Attribute<0x0020,0x0032> ipp2;
  gdcm::Attribute<0x0020,0x0037> iop2;
```

```
  ipp2.Set( ds2 );
  iop2.Set( ds2 );
  if( iop1 != iop2 )
    {
    return false;
    }

  // else
  double normal[3];
  normal[0] = iop1[1]*iop1[5] - iop1[2]*iop1[4];
  normal[1] = iop1[2]*iop1[3] - iop1[0]*iop1[5];
  normal[2] = iop1[0]*iop1[4] - iop1[1]*iop1[3];
  double dist1 = 0;
  for (int i = 0; i < 3; ++i) dist1 += normal[i]*ipp1[i];
  double dist2 = 0;
  for (int i = 0; i < 3; ++i) dist2 += normal[i]*ipp2[i];

  std::cout « dist1 « "," « dist2 « std::endl;
  return dist1 < dist2;

}


int main(int argc, char *argv[])
{
  const char *extradataroot = gdcm::Testing::GetDataExtraRoot();
  std::string dir1;
  if( argc < 2 )
    {
    if( !extradataroot )
      {
      return 1;
      }
    dir1 = extradataroot;
    dir1 += "/gdcmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
  else
    {
    dir1 = argv[1];
    }

  gdcm::Directory d;
  d.Load( dir1, true ); // recursive !
  const gdcm::Directory::FilenamesType &l1 = d.GetFilenames();
  const size_t nfiles = l1.size();
  std::cout « nfiles « std::endl;

  //if( nfiles != 280 )
  //  {
  //  return 1;
  //  }

  //d.Print( std::cout );

  gdcm::Scanner s0;
  const gdcm::Tag t1(0x0020,0x000d); // Study Instance UID
  const gdcm::Tag t2(0x0020,0x000e); // Series Instance UID
  //const gdcm::Tag t3(0x0010,0x0010); // Patient's Name
  s0.AddTag( t1 );
  s0.AddTag( t2 );
  //s0.AddTag( t3 );
  //s0.AddTag( t4 );
  //s0.AddTag( t5 );
  //s0.AddTag( t6 );
  bool b = s0.Scan( d.GetFilenames() );
  if( !b )
    {
    std::cerr « "Scanner failed" « std::endl;
    return 1;
    }

  //s0.Print( std::cout );

  // Only get the DICOM files:
  gdcm::Directory::FilenamesType l2 = s0.GetKeys();
  const size_t nfiles2 = l2.size();
  std::cout « nfiles2 « std::endl;

  if ( nfiles2 > nfiles )
    {
    return 1;
```

```
    }


  gdcm::Sorter sorter;
  sorter.SetSortFunction( mysort1 );
  sorter.StableSort( l2 );

  sorter.SetSortFunction( mysort2 );
  sorter.StableSort( sorter.GetFilenames() );

  sorter.SetSortFunction( mysort3 );
  sorter.StableSort( sorter.GetFilenames() );

  sorter.SetSortFunction( mysort4 );
  sorter.StableSort( sorter.GetFilenames() );

  //sorter.Print( std::cout );

  // Let's try to check our result:
  // assume that IPP is precise enough so that we can test floating point equality:
  size_t nvalues = 0;
{
  gdcm::Scanner s;
  s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
  //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
  s.Scan( d.GetFilenames() );

  //s.Print( std::cout );

  const gdcm::Scanner::ValuesType &values = s.GetValues();
  nvalues = values.size();
  std::cout « "There are " « nvalues « " different type of values" « std::endl;
  assert( nfiles2 % nvalues == 0 );
  std::cout « "Series is composed of " « (nfiles/nvalues) « " different 3D volumes" « std::endl;
}

  gdcm::Directory::FilenamesType sorted_files = sorter.GetFilenames();

  // Which means we can take nvalues files at a time and execute gdcm::IPPSorter on it:
  gdcm::IPPSorter ippsorter;
  gdcm::Directory::FilenamesType sub( sorted_files.begin(), sorted_files.begin() + nvalues);
  std::cout « sub.size() « std::endl;
  std::cout « sub[0] « std::endl;
  std::cout « sub[nvalues-1] « std::endl;
  ippsorter.SetComputeZSpacing( false );
  if( !ippsorter.Sort( sub ) )
    {
    std::cerr « "Could not sort" « std::endl;
    return 1;
    }

  std::cout « "IPPSorter:" « std::endl;
  ippsorter.Print( std::cout );


  return 0;
}
```

14.100 csa2img.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * I do not know what the format is, just guessing from info found on the net:
 *
```

```
 * http://atonal.ucdavis.edu/matlab/fmri/spm5/spm_dicom_convert.m
 *
 * This example is an attempt at understanding the format used by SIEMENS
 * their "SIEMENS CSA NON-IMAGE" DICOM file (1.3.12.2.1107.5.9.1)
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 */
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmCSAHeader.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  // gdcmDataExtra/gdcmNonImageData/exCSA_Non-Image_Storage.dcm
  // PHANTOM.MR.CARDIO_COEUR_S_QUENCE_DE_REP_RAGE.9.257.2008.03.20.14.53.25.578125.43151705.IMA
  const char *filename = argv[1];

  gdcm::Reader reader; // Do not use ImageReader
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }

  gdcm::CSAHeader csa;
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  const gdcm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
  //std::cout « t1 « std::endl;
  //const gdcm::PrivateTag &t2 = csa.GetCSASeriesHeaderInfoTag();

  if( ds.FindDataElement( t1 ) )
    {
    csa.LoadFromDataElement( ds.GetDataElement( t1 ) );
    csa.Print( std::cout );
    }
  int dims[2] = {};
  if( csa.FindCSAElementByName( "Columns" ) )
    {
    const gdcm::CSAElement &csael = csa.GetCSAElementByName( "Columns" );
    std::cout « csael « std::endl;
    //const gdcm::ByteValue *bv = csael.GetByteValue();
    gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el;
    el.Set( csael.GetValue() );
    dims[0] = el.GetValue();
    std::cout « "Columns:" « el.GetValue() « std::endl;
    }

  if( csa.FindCSAElementByName( "Rows" ) )
    {
    const gdcm::CSAElement &csael2 = csa.GetCSAElementByName( "Rows" );
    std::cout « csael2 « std::endl;
    gdcm::Element<gdcm::VR::IS, gdcm::VM::VM1> el2;
    el2.Set( csael2.GetValue() );
    dims[1] = el2.GetValue();
    std::cout « "Rows:" « el2.GetValue() « std::endl;
    }

  double spacing[2] = { 1. , 1. };
  bool spacingfound = false;
  if( csa.FindCSAElementByName( "PixelSpacing" ) )
    {
    const gdcm::CSAElement &csael3 = csa.GetCSAElementByName( "PixelSpacing" );
    if( !csael3.IsEmpty() )
      {
      std::cout « csael3 « std::endl;
      gdcm::Element<gdcm::VR::DS, gdcm::VM::VM2> el3;
      el3.Set( csael3.GetValue() );
      spacing[0] = el3.GetValue(0);
```

```
      spacing[1] = el3.GetValue(1);
      std::cout « "PixelSpacing:" « el3.GetValue() « "," « el3.GetValue(1) « std::endl;
      spacingfound = true;
      }
    }

  if( !spacingfound )
    {
    std::cerr « "Problem with PixelSpacing" « std::endl;
    //return 1;
    }
  if( !dims[0] || !dims[1] )
    {
    std::cerr « "Problem with dims" « std::endl;
    return 1;
    }

  gdcm::ImageWriter writer;

  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 2 ); // good default
  image.SetDimension(0, dims[0] );
  image.SetDimension(1, dims[1] );
  image.SetSpacing(0, spacing[0] );
  image.SetSpacing(1, spacing[1] );
  gdcm::PixelFormat pixeltype = gdcm::PixelFormat::INT16; // bytepix = spm_type('int16','bits')/8;

  //unsigned long l = image.GetBufferLength();
  //const int p =  l / (dims[0] * dims[1]);

  //image.SetNumberOfDimensions( 3 );
  //image.SetDimension(2, p / pixeltype.GetPixelSize() );

  gdcm::PhotometricInterpretation pi;
  pi = gdcm::PhotometricInterpretation::MONOCHROME2;
  //pixeltype.SetSamplesPerPixel(  );
  image.SetPhotometricInterpretation( pi );
  image.SetPixelFormat( pixeltype );
  //image.SetIntercept( inputimage.GetIntercept() );
  //image.SetSlope( inputimage.GetSlope() );

  //gdcm::DataElement pixeldata( gdcm::Tag(0x7fe1,0x1010) );
  //pixeldata.SetByteValue( &outbuf[0], outbuf.size() );
  gdcm::PrivateTag csanonimaget(0x7fe1,0x10,"SIEMENS CSA NON-IMAGE");
  const gdcm::DataElement &pixeldata = ds.GetDataElement( csanonimaget );
  image.SetDataElement( pixeldata );

  std::string outfilename = "outcsa.dcm";
  //writer.SetFile( reader.GetFile() );
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr « "could not write: " « outfilename « std::endl;
    return 1;
    }


  return 0;
}
```

14.101 iU22tomultisc.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
```

```
 * iU22 Raw Data extractor
 */
#include "gdcmReader.h"
#include "gdcmImageWriter.h"
#include "gdcmAttribute.h"
#include "gdcmPrivateTag.h"

#include <math.h>

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  // IM_001
  const char *filename = argv[1];

  gdcm::Reader reader; // Do not use ImageReader
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }

 // * The data is simply 8-bit unsigned in the obvious x/y/z order
 // * 200D,300B contains the data
 // * 200D,3001 contains the no. of voxels (416,412,256 in this case)
 // * 200D,3003 contains the voxel sizes (0.156184527398215 /
 // 0.1223749613981957 / 0.328479990704639 in this case)

  const gdcm::File &file = reader.GetFile();
  const gdcm::DataSet &ds = file.GetDataSet();
  const gdcm::PrivateTag trawdataus( 0x200d, 0x0b, "Philips US Imaging DD 033" );
  const gdcm::DataElement &rawdataus = ds.GetDataElement( trawdataus );

  const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x01, "Philips US Imaging DD 036" );
  const gdcm::DataElement &colsrowsframes = ds.GetDataElement( tcolsrowsframes );
  // const gdcm::PrivateTag tcolsrowsframes( 0x200d, 0x02, "Philips US Imaging DD 036" );
  // this is just a duplicate previous tag.
  const gdcm::PrivateTag tvoxelspacing( 0x200d, 0x03, "Philips US Imaging DD 036" );
  const gdcm::DataElement &voxelspacing = ds.GetDataElement( tvoxelspacing );

  gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3> dims; // Use DS to interpret value stored in LO
  dims.SetFromDataElement( colsrowsframes );

  gdcm::Element<gdcm::VR::DS,gdcm::VM::VM3> spacing;
  spacing.SetFromDataElement( voxelspacing );

  gdcm::ImageWriter writer;

  gdcm::Image &image = writer.GetImage();
  image.SetNumberOfDimensions( 3 ); // good default
  image.SetDimension(0, (unsigned int)dims[0] );
  image.SetDimension(1, (unsigned int)dims[1] );
  image.SetDimension(2, (unsigned int)dims[2] );
  image.SetSpacing(0, spacing[0] );
  image.SetSpacing(1, spacing[1] );
  image.SetSpacing(2, spacing[2] );
  gdcm::PixelFormat pixeltype = gdcm::PixelFormat::UINT8;

  gdcm::PhotometricInterpretation pi;
  pi = gdcm::PhotometricInterpretation::MONOCHROME2;
  image.SetPhotometricInterpretation( pi );
  image.SetPixelFormat( pixeltype );

  image.SetDataElement( rawdataus );

  std::string outfilename = "outiu22.dcm";

  gdcm::DataElement de( gdcm::Tag(0x8,0x16) ); // SOP Class UID
  de.SetVR( gdcm::VR::UI );
  gdcm::MediaStorage ms(
    gdcm::MediaStorage::UltrasoundMultiFrameImageStorage );
//    gdcm::MediaStorage::MultiframeGrayscaleByteSecondaryCaptureImageStorage );
  de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()));
  writer.GetFile().GetDataSet().Replace( de );

  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr « "could not write: " « outfilename « std::endl;
    return 1;
```

```
    }

  return 0;
}
```

14.102 pmsct_rgb1.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to rewrite a ELSCINT1/PMSCT_RGB1 compressed
 * image so that it is readable by most 3rd party software (DICOM does
 * not specify this particular encoding).
 * This is required for the sake of interoperability with any standard
 * conforming DICOM system.
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Jean-Pierre Roux for providing the sample datasets
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

void delta_decode(const unsigned char *data_in, size_t data_size,
  std::vector<unsigned char> &new_stream, unsigned short pc, size_t w, size_t h)
{
  const size_t plane_size = h * w;
  const size_t outputlen = 3 * plane_size;
  new_stream.resize( outputlen );

  assert( data_size != outputlen );
  if( data_size == outputlen )
    {
    return;
    }
  typedef unsigned char byte;
  enum {
    COLORMODE  = 0x81,
    ESCMODE    = 0x82,
    REPEATMODE = 0x83
  };

  const byte* src = (const byte*)data_in;
  byte* dest = (byte*)new_stream.data();
  union { byte gray; byte rgb[3]; } pixel;
  pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
  // always start in grayscale mode
  bool graymode = true;
  size_t dx = 1;
  size_t dy = 3;
  // algorithm works with both planar configuration
  // It does produce surprising greenish background color for planar
  // configuration is 0, while the nested Icon SQ display a nice black
  // background
```

```
if (pc)
  {
  dx = plane_size;
  dy = 1;
  }
size_t ps = plane_size;

// The following is highly unoptimized as we have nested if statement in a while loop
// we need to switch from one algorithm to the other (RGB <-> GRAY)
while (ps)
  {
  // next byte:
  byte b = *src++;
  assert( src < data_in + data_size );
  // mode selection:
  switch ( b )
    {
  case ESCMODE:
    // Used to treat a byte 81/82/83 as a normal byte
    if (graymode)
      {
      pixel.gray += *src++;
      dest[0*dx] = pixel.gray;
      dest[1*dx] = pixel.gray;
      dest[2*dx] = pixel.gray;
      }
    else
      {
      pixel.rgb[0] += *src++;
      pixel.rgb[1] += *src++;
      pixel.rgb[2] += *src++;
      dest[0*dx] = pixel.rgb[0];
      dest[1*dx] = pixel.rgb[1];
      dest[2*dx] = pixel.rgb[2];
      }
    dest += dy;
    ps--;
    break;
  case REPEATMODE:
    // repeat mode (RLE)
    b = *src++;
    ps -= b;
    if (graymode)
      {
      while (b-- > 0)
        {
        dest[0*dx] = pixel.gray;
        dest[1*dx] = pixel.gray;
        dest[2*dx] = pixel.gray;
        dest += dy;
        }
      }
    else
      {
      while (b-- > 0)
        {
        dest[0*dx] = pixel.rgb[0];
        dest[1*dx] = pixel.rgb[1];
        dest[2*dx] = pixel.rgb[2];
        dest += dy;
        }
      }
    break;
  case COLORMODE:
    // We are switching from one mode to the other. The stream contains an intermixed
    // compression of RGB codec and GRAY codec. Each one not knowing of the other
    // reset old value to 0.
    if (graymode)
      {
      graymode = false;
      pixel.rgb[0] = pixel.rgb[1] = pixel.rgb[2] = 0;
      }
    else
      {
      graymode = true;
      pixel.gray = 0;
      }
    break;
  default:
    // This is identical to ESCMODE, it would be nicer to use fall-through
    if (graymode)
```

```
        {
        pixel.gray += b;
        dest[0*dx] = pixel.gray;
        dest[1*dx] = pixel.gray;
        dest[2*dx] = pixel.gray;
        }
      else
        {
        pixel.rgb[0] += b;
        pixel.rgb[1] += *src++;
        pixel.rgb[2] += *src++;
        dest[0*dx] = pixel.rgb[0];
        dest[1*dx] = pixel.rgb[1];
        dest[2*dx] = pixel.rgb[2];
        }
      dest += dy;
      ps--;
      break;
      } // end switch
    } // end while
}

int main(int argc, char *argv [])
{
  if( argc < 2 ) return 1;
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  // (07a1,1011) CS [PMSCT_RGB1]                                    # 10,1 Tamar Compression Type
  const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
  if( !ds.FindDataElement( tcompressiontype ) ) return 1;
  const gdcm::DataElement& compressiontype = ds.GetDataElement( tcompressiontype );
  if ( compressiontype.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv = compressiontype.GetByteValue();
  std::string comprle = "PMSCT_RLE1";
  std::string comprgb = "PMSCT_RGB1";
  bool isrle = false;
  bool isrgb = false;
  if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
    {
    isrle = true;
    return 1;
    }
  if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) == 0 )
    {
    isrgb = true;
    }
  if( !isrgb && !isrle ) return 1;

  const gdcm::PrivateTag tcompressedpixeldata(0x07a1,0x000a,"ELSCINT1");
  if( !ds.FindDataElement( tcompressedpixeldata) ) return 1;
  const gdcm::DataElement& compressionpixeldata = ds.GetDataElement( tcompressedpixeldata);
  if ( compressionpixeldata.IsEmpty() ) return 1;
  const gdcm::ByteValue * bv2 = compressionpixeldata.GetByteValue();

  gdcm::Attribute<0x0028,0x0006> at0;
  at0.SetFromDataSet( ds );
  gdcm::Attribute<0x0028,0x0010> at1;
  at1.SetFromDataSet( ds );
  gdcm::Attribute<0x0028,0x0011> at2;
  at2.SetFromDataSet( ds );

  std::vector<unsigned char> buffer;
  delta_decode((const unsigned char*)bv2->GetPointer(), bv2->GetLength(), buffer,
    at0.GetValue(), at1.GetValue(), at2.GetValue() );

  gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
  pixeldata.SetVR( gdcm::VR::OW );
  pixeldata.SetByteValue( (char*)buffer.data(), (uint32_t)buffer.size() );
  // TODO we should check that decompress byte buffer match the expected size (row*col*...)

  // Add the pixel data element
```

```
  reader.GetFile().GetDataSet().Replace( pixeldata );

  reader.GetFile().GetHeader().SetDataSetTransferSyntax(
    gdcm::TransferSyntax::ExplicitVRLittleEndian);
  gdcm::Writer writer;
  writer.SetFile( reader.GetFile() );

  // Cleanup stuff:
  // remove the compressed pixel data:
  // FIXME: should I remove more private tags ? all of them ?
  // oh well this is just an example
  // use gdcm::Anonymizer::RemovePrivateTags if needed...
  writer.GetFile().GetDataSet().Remove( compressionpixeldata.GetTag() );
  std::string outfilename;
  if (argc > 2)
     outfilename = argv[2];
  else
     outfilename = "outrgb.dcm";
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr « "Failed to write" « std::endl;
    return 1;
    }

  std::cout « "success !" « std::endl;

  return 0;
}
```

14.103 rle2img.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * This example shows how to rewrite a ELSCINT1/PMSCT_RLE1 compressed
 * image so that it is readable by most 3rd party software (DICOM does
 * not specify this particular encoding).
 * This is required for the sake of interoperability with any standard
 * conforming DICOM system.
 *
 * Everything done in this code is for the sole purpose of writing interoperable
 * software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
 * If you believe anything in this code violates any law or any of your rights,
 * please contact us (gdcm-developers@lists.sourceforge.net) so that we can
 * find a solution.
 *
 * Everything you do with this code is at your own risk, since decompression
 * algorithm was not written from specification documents.
 *
 * Special thanks to:
 * Mauro Maiorca for bringing to our attention on this new ELSCINT1
 * compression algorithm : PMSCT_RLE1 (different from the 'LOSSLESS RICE')
 * See post at:
 * http://groups.google.com/group/comp.protocols.dicom/msg/f2b99bf706a7f8ca
 *
 * Thanks to Jesus Spinola, for more datasets,
 * http://www.itk.org/pipermail/insight-users/2008-April/025571.html
 *
 * And last but not least, a very big thank to Ivo van Poorten, without
 * whom we would still be looking at this compressed byte stream as if
 * it was RLE compressed.
 */
#include "gdcmReader.h"
#include "gdcmPrivateTag.h"
```

```
#include "gdcmAttribute.h"
#include "gdcmImageWriter.h"

/* FIXME: Why is PhilipsLosslessRice.dcm a 512x512 image ... */
void delta_decode(const char *inbuffer, size_t length, std::vector<unsigned short> &output)
{
  // RLE pass
  std::vector<char> temp;
  for(size_t i = 0; i < length; ++i)
    {
    if( inbuffer[i] == (char)0xa5 )
      {
      //unsigned char repeat = (unsigned char)inbuffer[i+1] + 1;
      //assert( (unsigned char)inbuffer[i+1] != 255 );
      int repeat = (unsigned char)inbuffer[i+1] + 1;
      char value = inbuffer[i+2];
      while(repeat)
        {
        temp.push_back( value );
        --repeat;
        }
      i+=2;
      }
    else
      {
      temp.push_back( inbuffer[i] );
      }
    }

  // Delta encoding pass
  unsigned short delta = 0;
  for(size_t i = 0; i < temp.size(); ++i)
    {
    if( temp[i] == 0x5a )
      {
      unsigned char v1 = (unsigned char)temp[i+1];
      unsigned char v2 = (unsigned char)temp[i+2];
      unsigned short value = (unsigned short)(v2 * 256 + v1);
      output.push_back( value );
      delta = value;
      i+=2;
      }
    else
      {
      unsigned short value = (unsigned short)(temp[i] + delta);
      output.push_back( value );
      delta = value;
      }
    //assert( output[output.size()-1] == ref[output.size()-1] );
    }

  if ( output.size() % 2 )
    {
    output.resize( output.size() - 1 );
    }
  std::cout « length « " -> " « output.size() * 2 « std::endl;
}

int main(int argc, char *argv [])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « "input.dcm [output.dcm]" « std::endl;
    std::cerr « "will default to 'outrle.dcm' unless output.dcm is specified."
      « std::endl;
    return 1;
    }
  const char *filename = argv[1];
  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    std::cerr « "Failed to read: " « filename « std::endl;
    return 1;
    }
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();

  // (07a1,1011) CS [PMSCT_RLE1]                                      # 10,1 Tamar Compression Type
  const gdcm::PrivateTag tcompressiontype(0x07a1,0x0011,"ELSCINT1");
  if( !ds.FindDataElement( tcompressiontype ) ) return 1;
  const gdcm::DataElement& compressiontype = ds.GetDataElement( tcompressiontype );
```

```
if ( compressiontype.IsEmpty() ) return 1;
const gdcm::ByteValue * bv = compressiontype.GetByteValue();
std::string comprle = "PMSCT_RLE1";
std::string comprgb = "PMSCT_RGB1";
bool isrle = false;
bool isrgb = false;
if( strncmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
  {
  isrle = true;
  }
if( strncmp( bv->GetPointer(), comprgb.c_str(), comprgb.size() ) == 0 )
  {
  isrgb = true;
  std::cerr « "See: pmsct_rgb1.cxx instead" « std::endl;
  return 1;
  }
if( !isrgb && !isrle ) return 1;

// check if compressed pixel data reside in private or standard tag
const gdcm::PrivateTag tprivatepixeldata(0x07a1,0x100a,"ELSCINT1");
const gdcm::Tag tstandardpixeldata(0x7fe0, 0x0010);
gdcm::Tag tpixeldata;
if(ds.FindDataElement(tprivatepixeldata)) tpixeldata = tprivatepixeldata;
else if(ds.FindDataElement(tstandardpixeldata)) tpixeldata = tstandardpixeldata;
if(!ds.FindDataElement(tpixeldata)) return 1;

const gdcm::DataElement& compressionpixeldata = ds.GetDataElement( tpixeldata);
if ( compressionpixeldata.IsEmpty() ) return 1;
const gdcm::ByteValue * bv2 = compressionpixeldata.GetByteValue();

gdcm::Attribute<0x0028,0x0010> at1;
at1.SetFromDataSet( ds );
gdcm::Attribute<0x0028,0x0011> at2;
at2.SetFromDataSet( ds );

gdcm::DataElement pixeldata;
// if standard voxel data element does not exist, create it
if( !reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
{
  pixeldata = gdcm::DataElement( tpixeldata, 0, gdcm::VR::OW );
}
else{
  pixeldata = reader.GetFile().GetDataSet().GetDataElement( tpixeldata );
}

pixeldata.SetVR( gdcm::VR::OW );
gdcm::VL bv2l = bv2->GetLength();
gdcm::VL at1l = at1.GetValue() * at2.GetValue() * 2; /* sizeof(unsigned short) == 2 */
// Handle special case that is not compressed:
if( bv2l == at1l )
  {
  pixeldata.SetByteValue( bv2->GetPointer(), bv2->GetLength() );
  }
else
  {
  std::vector<unsigned short> buffer;
  delta_decode(bv2->GetPointer(), bv2->GetLength(), buffer);
  pixeldata.SetByteValue( (char*)buffer.data(), (uint32_t)(buffer.size() * sizeof( unsigned short )) );
  }
// TODO we should check that decompress byte buffer match the expected size (row*col*...)

// Add the pixel data element
if( reader.GetFile().GetDataSet().FindDataElement( tpixeldata ) )
{
  reader.GetFile().GetDataSet().Replace( pixeldata );
}
else
{
  reader.GetFile().GetDataSet().ReplaceEmpty( pixeldata );
}


reader.GetFile().GetHeader().SetDataSetTransferSyntax(
  gdcm::TransferSyntax::ExplicitVRLittleEndian);
gdcm::Writer writer;
writer.SetFile( reader.GetFile() );

// Cleanup stuff:
// This makes the code equivalent to Philips workstation IntelliSpace Portal
if( writer.GetFile().GetDataSet().FindDataElement( tcompressiontype ) )
{
```

```
    writer.GetFile().GetDataSet().Remove( gdcm::Tag(0x07a1,0x1011) );
  }
  if( writer.GetFile().GetDataSet().FindDataElement( tprivatepixeldata ) )
  {
    writer.GetFile().GetDataSet().Remove( gdcm::Tag(0x07a1,0x100a) );
  }

  std::string outfilename;
  if (argc > 2)
     outfilename = argv[2];
  else
     outfilename = "outrle.dcm";
  writer.SetFileName( outfilename.c_str() );
  if( !writer.Write() )
    {
    std::cerr « "Failed to write" « std::endl;
    return 1;
    }

  std::cout « "success !" « std::endl;

  return 0;
}
```

14.104 uid_unique.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmUIDGenerator.h"

#include <iostream>
#include <string>
#include <set>

int main()
{
  gdcm::UIDGenerator uid;
  //const char myroot[] = "9876543210.9876543210.9876543210.9876543210.9876543210"; // fails in ~40000 tries
  const char myroot[] = "9876543210.9876543210.9876543210";
  uid.SetRoot( myroot );
  std::set<std::string> uids;
  uint64_t wrap = 0;
  uint64_t c = 0;
  while(true)
    {
    const char *unique = uid.Generate();
    //std::cout « unique « std::endl;
    if( c % 10000 == 0 )
      {
      std::cout « "wrap=" « wrap « ",c=" « c « std::endl;
      }
    ++c;
    if( c == 0 )
      {
      wrap++;
      }
    if ( uids.count(unique) == 1 )
      {
      std::cerr « "Failed with: " « unique « std::endl;
      return 1;
      }
    uids.insert( unique );
    }
}
```

14.105 DecompressImage.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This example will take in a DICOM file, and tries to decompress it (actually write it
 * as ImplicitVRLittleEndian Transfer Syntax).
 *
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/DecompressImage.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java DecompressImage gdcmData/012345.002.050.dcm out.dcm
 */
import gdcm.*;

public class DecompressImage
{
  public static void main(String[] args) throws Exception
    {
    String file1 = args[0];
    String file2 = args[1];
    ImageReader reader = new ImageReader();
    reader.SetFileName( file1 );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read: " + file1 );
      }

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
    change.SetInput( reader.GetImage() );
    if( !change.Change() )
      {
      throw new Exception("Could not change: " + file1 );
      }

    Image out = change.GetOutput();
    System.out.println( out.toString() );

    // Set the Source Application Entity Title
    FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );

    ImageWriter writer = new ImageWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( out );
    ret = writer.Write();
    if( !ret )
      {
      throw new Exception("Could not write: " + file2 );
      }

    }
}
```

14.106 DecompressPixmap.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This example will take in a DICOM file, and tries to decompress it (actually write it
 * as ImplicitVRLittleEndian Transfer Syntax).
 *
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/DecompressPixmap.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java DecompressPixmap gdcmData/012345.002.050.dcm out.dcm
 */
import gdcm.*;

public class DecompressPixmap
{
  public static void main(String[] args) throws Exception
    {
    String file1 = args[0];
    String file2 = args[1];
    PixmapReader reader = new PixmapReader();
    reader.SetFileName( file1 );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read: " + file1 );
      }

    ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
    change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
    PixmapToPixmapFilter filter = (PixmapToPixmapFilter)change;
    filter.SetInput( reader.GetPixmap() );
    if( !change.Change() )
      {
      throw new Exception("Could not change: " + file1 );
      }

    // The following does not work in Java/swig 2.0.7
    //Pixmap p = ((PixmapToPixmapFilter)change).GetOutput();
    Pixmap p = change.GetOutputAsPixmap();  // be explicit
    //System.out.println( p.toString() );

    // Set the Source Application Entity Title
    FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );

    PixmapWriter writer = new PixmapWriter();
    writer.SetFileName( file2 );
    writer.SetFile( reader.GetFile() );
    writer.SetImage( p );
    ret = writer.Write();
    if( !ret )
      {
      throw new Exception("Could not write: " + file2 );
      }

    }
}
```

14.107 ExtractImageRegion.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
```

```
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java ExtractImageRegion input.dcm
 */
import gdcm.*;
import java.io.FileOutputStream;

public class ExtractImageRegion
{
  public static void main(String[] args) throws Exception
    {
    String filename = args[0];

    // instantiate the reader:
    ImageRegionReader reader = new ImageRegionReader();
    reader.SetFileName( filename );

    // pull DICOM info:
    if (!reader.ReadInformation()) return;
    // Get file infos
    File f = reader.GetFile();

    // get some info about image
    UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
    PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
    int pixelsize = pf.GetPixelSize();

    // buffer to get the pixels
    long buffer_length = dims.get(0) * dims.get(1) * pixelsize;
    byte[] buffer = new byte[ (int)buffer_length ];

    // define a simple box region.
    BoxRegion box = new BoxRegion();
    for (int z = 0; z < dims.get(2); z++)
      {
      // Define that I want the image 0, full size (dimx x dimy pixels)
      // and do that for each z:
      box.SetDomain(0, dims.get(0) - 1, 0, dims.get(1) - 1, z, z);
      //System.Console.WriteLine( box.toString() );
      reader.SetRegion( box );

      // reader will try to load the uncompressed image region into buffer.
      // the call returns an error when buffer.Length is too small. For instance
      // one can call:
      // long buf_len = reader.ComputeBufferLength(); // take into account pixel size
      // to get the exact size of minimum buffer
      if (reader.ReadIntoBuffer(buffer, buffer_length))
        {
        FileOutputStream fos = new FileOutputStream("/tmp/frame.raw");
        fos.write(buffer);
        fos.close();
        }
      else
        {
        throw new Exception("can't read pixels error");
        }
      }

    }
}
```

14.108 FileAnonymize.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library
```

```
  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

import gdcm.*;

public class FileAnonymize
{
  public static class MyWatcher extends SimpleSubjectWatcher
    {
    public MyWatcher(Subject s) { super(s,"Override String"); }
    protected void ShowProgress(Subject caller, Event evt)
      {
      ProgressEvent pe = ProgressEvent.Cast(evt);
      System.out.println( "This is my progress: " + pe.GetProgress() );
      }
    }

  public static void main(String[] args) throws Exception
    {
    String input = args[0];
    String output = args[1];

    FileAnonymizer fa = new FileAnonymizer();
    fa.SetInputFileName( input );
    fa.SetOutputFileName( output );

    // Empty Operations
    // It will create elements, since those tags are non-registered public elements (2011):
    fa.Empty( new Tag(0x0008,0x1313) );
    fa.Empty( new Tag(0x0008,0x1317) );
    // Remove Operations
    // The following Tag are actually carefully chosen, since they refer to SQ:
    fa.Remove( new Tag(0x0008,0x2112) );
    fa.Remove( new Tag(0x0008,0x9215) );
    // Replace Operations
    // do not call replace operation on SQ attribute !
    fa.Replace( new Tag(0x0018,0x5100), "MYVALUE " );
    fa.Replace( new Tag(0x0008,0x1160), "MYOTHERVAL" );

    if( !fa.Write() )
      {
      System.out.println( "Could not write" );
      return;
      }

    System.out.println( "success" );
    }
}
```

14.109 HelloSimple.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/HelloSimple.java -d .
 *
 * Usage:
```

```
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java HelloSimple gdcmData/012345.002.050.dcm
 */
import gdcm.*;

public class HelloSimple
{
  public static void main(String[] args) throws Exception
    {
    String filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read: " + filename );
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();

    System.out.println( ds.toString() );

    System.out.println("Success reading: " + filename );
    }
}
```

14.110 ReadFiles.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import gdcm.*;
import java.io.File;

public class ReadFiles
{
  static int i = 0;
  public static void process(String path)
    {
    //String path = file.getPath();
    assert PosixEmulation.FileExists(path) : "Problem converting to 8bits";

    System.out.println("Reading: " + path );
    System.out.println("File: " + i++);
    Reader r = new Reader();
    try
      {
      r.SetFileName( path );
      TagSetType skip = new TagSetType();
      skip.insert( new Tag(0x7fe0,0x10) );
      boolean b = r.ReadUpToTag( new Tag(0x88,0x200), skip );
      //System.out.println("DS:\n" + r.GetFile().GetDataSet().toString() );
      }
    finally
      {
      r.delete(); // will properly call C++ destructor and close file descriptor
      }
    }

  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
```

```
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }

  public static void waiting (int n)
    {
    long t0, t1;
    t0 =  System.currentTimeMillis();
    do
      {
      t1 = System.currentTimeMillis();
      }
    while ((t1 - t0) < (n * 1000));
    }

  public static void main(String[] args) throws Exception
    {
    String directory = args[0];

    Directory gdir = new Directory();
    long n = gdir.Load( directory, true );
    System.out.println( gdir.toString() );
    FilenamesType files = gdir.GetFilenames();
    for( long i = 0; i < n; ++i )
      {
      String path = files.get( (int)i );
      process( path );
      }

    System.out.println( "Java API" );

    //waiting( 10 );
    for( int i = 0; i < 2; ++i )
      {
      File dir = new File(directory);
      visitAllFiles(dir);
      }
    }
}
```

14.111 ScanDirectory.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

import gdcm.*;
import gdcm.Reader;
import gdcm.LookupTable;
import java.io.File;
import java.io.*;
import java.awt.image.*;
import javax.imageio.ImageIO;

public class ScanDirectory
{
  public static class MyWatcher extends SimpleSubjectWatcher
    {
    public MyWatcher(Subject s) { super(s,"Override String"); }
    protected void ShowProgress(Subject caller, Event evt)
      {
```

```
      ProgressEvent pe = ProgressEvent.Cast(evt);
      System.out.println( "This is my progress: " + pe.GetProgress() );
      }
    }

  public static byte[] GetAsByte(Bitmap input)
    {
    long len = input.GetBufferLength();
    byte[] buffer = new byte[ (int)len ];
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
      {
      ImageChangePhotometricInterpretation icpi = new ImageChangePhotometricInterpretation();
      icpi.SetInput( input );
      icpi.SetPhotometricInterpretation(
        new PhotometricInterpretation(
          PhotometricInterpretation.PIType.MONOCHROME2 ) );
      if( icpi.Change() )
        {
        Bitmap output = icpi.GetOutput();
        output.GetArray( buffer );
        }
      return buffer;
      }
    else
      {
      input.GetArray( buffer );
      return buffer;
      }
    }
  public static short[] GetAsShort(Bitmap input)
    {
    long len = input.GetBufferLength(); // length in bytes
    short[] buffer = new short[ (int)len / 2 ];
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1 )
      {
      ImageChangePhotometricInterpretation icpi = new ImageChangePhotometricInterpretation();
      icpi.SetInput( input );
      icpi.SetPhotometricInterpretation(
        new PhotometricInterpretation(
          PhotometricInterpretation.PIType.MONOCHROME2 ) );
      if( icpi.Change() )
        {
        Bitmap output = icpi.GetOutput();
        output.GetArray( buffer );
        }
      return buffer;
      }
    else
      {
      input.GetArray( buffer );
      return buffer;
      }
    }
  public static boolean WritePNG(Bitmap input, String outfilename )
    {
    int imageType = BufferedImage.TYPE_CUSTOM;
    PixelFormat pf = input.GetPixelFormat();
    PhotometricInterpretation pi = input.GetPhotometricInterpretation();
    // We need to handle both public and private icon
    // It could well be that we are getting an RGB Icon or 16 bits Icon:
    ColorModel colorModel = null;
    if( pf.GetSamplesPerPixel() == 1 )
      {
      if( pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME1
         || pi.GetType() == PhotometricInterpretation.PIType.MONOCHROME2 )
        {
        if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
          {
          imageType = BufferedImage.TYPE_BYTE_GRAY;
          }
        else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT12 )
          {
          imageType = BufferedImage.TYPE_USHORT_GRAY;
          }
        else if( pf.GetScalarType() == PixelFormat.ScalarType.UINT16 )
          {
          imageType = BufferedImage.TYPE_USHORT_GRAY;
          }
        }
```

```
      else if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
        {
        LookupTable lut = input.GetLUT();
        long rl = lut.GetLUTLength( LookupTable.LookupTableType.RED );
        byte[] rbuf = new byte[ (int)rl ];
        long rl2 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
        assert rl == rl2;
        long gl = lut.GetLUTLength( LookupTable.LookupTableType.GREEN );
        byte[] gbuf = new byte[ (int)gl ];
        long gl2 = lut.GetLUT( LookupTable.LookupTableType.GREEN, gbuf );
        assert gl == gl2;
        long bl = lut.GetLUTLength( LookupTable.LookupTableType.BLUE );
        byte[] bbuf = new byte[ (int)bl ];
        long bl2 = lut.GetLUT( LookupTable.LookupTableType.BLUE, bbuf );
        assert bl == bl2;
        colorModel = new IndexColorModel(8, (int)rl, rbuf, gbuf, bbuf);
        // For code below
        imageType = BufferedImage.TYPE_BYTE_GRAY;
        }
      }
    else if( pf.GetSamplesPerPixel() == 3 )
      {
      if( pf.GetScalarType() == PixelFormat.ScalarType.UINT8 )
        {
        // FIXME should be TYPE_3BYTE_RGB
        imageType = BufferedImage.TYPE_3BYTE_BGR;
        }
      }
    //System.out.println( "pf: " + pf.toString() );
    //System.out.println( "pi: " + pi.toString() );
    long width  = input.GetDimension(0);
    long height = input.GetDimension(0);
    BufferedImage bi;
    if( pi.GetType() == PhotometricInterpretation.PIType.PALETTE_COLOR )
      {
      bi = new BufferedImage(colorModel,
        colorModel.createCompatibleWritableRaster((int)width, (int)height),
        false, null);
      }
    else
      {
      bi = new BufferedImage((int)width,(int)height,imageType);
      }
    WritableRaster wr = bi.getRaster();
    //System.out.println( "imagetype: " + imageType );
    if( imageType == BufferedImage.TYPE_BYTE_GRAY
      || imageType == BufferedImage.TYPE_3BYTE_BGR )
      {
      byte[] buffer = GetAsByte( input );
      wr.setDataElements (0, 0, (int)width, (int)height, buffer);
      }
    else if( imageType == BufferedImage.TYPE_USHORT_GRAY )
      {
      short[] buffer = GetAsShort( input );
      wr.setDataElements (0, 0, (int)width, (int)height, buffer);
      }

    File outputfile = new File( outfilename );
    try {
      ImageIO.write(bi, "png", outputfile);
    } catch (IOException e) {
      return false;
    }
    return true;
    }

  public static void main(String[] args) throws Exception
    {
    String directory = args[0];

    Directory d = new Directory();
    long nfiles = d.Load( directory, true );
    if(nfiles == 0)
      {
      throw new Exception("No files found");
      }
//    System.out.println( "Files:\n" + d.toString() );
    FilenamesType fns = d.GetFilenames();

    //Scanner s = new Scanner();
    SmartPtrScan sscan = Scanner.New();
```

```
Scanner s = sscan.__ref__();
//SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
MyWatcher watcher = new MyWatcher(s);
Tag[] tagarray = {
  new Tag(0x0010, 0x0010),    // PatientName
  new Tag(0x0010, 0x0020),    // PatientID
  new Tag(0x0010, 0x0030),    // PatientBirthDate
  new Tag(0x0010, 0x0040),    // PatientSex
  new Tag(0x0010, 0x1010),    // PatientAge
  new Tag(0x0020, 0x000d),    // StudyInstanceUID
  new Tag(0x0020, 0x0010),    // StudyID
  new Tag(0x0008, 0x0020),    // StudyDate
  new Tag(0x0008, 0x1030),    // StudyDescription
  new Tag(0x0020, 0x000e),    // SeriesInstanceUID
  new Tag(0x0020, 0x0011),    // SeriesNumber
  new Tag(0x0008, 0x0021),    // SeriesDate
  new Tag(0x0008, 0x103e),    // SeriesDescription
  new Tag(0x0008, 0x0090),    // ReferringPhysicianName
  new Tag(0x0008, 0x0060),    // Modality
  new Tag(0x0054, 0x0400),    // ImageID ?? Should be Instance number ??
  new Tag(0x0008, 0x0018),    // SOPInstanceUID
  new Tag(0x0008, 0x0032),    // AcquisitionTime
  new Tag(0x0008, 0x0033),    // ContentTime
  new Tag(0x0020, 0x0013),    // InstanceNumber
  new Tag(0x0020, 0x1041),    // SliceLocation
  new Tag(0x0018, 0x0050),    // SliceThickness ?? Eg. Enhanced MR Image Storage
  new Tag(0x0008, 0x0080),    // InstitutionName
  new Tag(0x0028, 0x1050),    // WindowCenter
  new Tag(0x0028, 0x1051),    // WindowWidth
};
for( Tag t : tagarray ) {
  //System.out.println( "Tag: " + t.toString() );
  s.AddTag( t );
}
boolean b = s.Scan( fns );
if(!b)
  {
  throw new Exception("Could not scan");
  }
String fn0 = fns.get(0);
TagToValue mappings = s.GetMapping( fn0 );
System.out.println( "mappings size: " + mappings.size() );
for( Tag tag : tagarray ) {
  if( mappings.has_key( tag ) ) {
    String val = mappings.get( tag );
    System.out.println( "tag/val: " + tag + "->" + val );
  }
}

for( long idx = 0; idx < fns.size(); ++idx )
  {
  Reader r = new Reader();
  String fn = fns.get( (int)idx );
  String outfn = fn + ".png";
  r.SetFileName( fn );
  TagSetType tst = new TagSetType();
  tst.insert( new Tag(0x7fe0,0x10) );
  b = r.ReadUpToTag( new Tag(0x88,0x200), tst );
  UIntArrayType dims = ImageHelper.GetDimensionsValue( r.GetFile() );
  if( b )
    {
    IconImageFilter iif = new IconImageFilter();
    System.out.println( "Processing: " + fn );

    iif.SetFile( r.GetFile() );
    b = iif.Extract();
    if( b )
      {
      Bitmap icon = iif.GetIconImage(0);
      WritePNG(icon, outfn);
      }
    else
      {
      ImageReader ir = new ImageReader();
      ir.SetFileName( fn );
      if( ir.Read() )
        {
        Image img = ir.GetImage();
        StringFilter sf = new StringFilter();
        sf.SetFile( r.GetFile() );
        String strval = sf.ToString( new Tag(0x0028,0x0120) );
```

```
            IconImageGenerator iig = new IconImageGenerator();
            iig.SetPixmap( img );
            iig.AutoPixelMinMax( true );
            try {
              double val = Double.parseDouble( strval );
              iig.SetOutsideValuePixel( val );
            }
            catch ( NumberFormatException e) {
              }
            iig.ConvertRGBToPaletteColor( false );
            long idims[] = { 128, 128};
            iig.SetOutputDimensions( idims );
            iig.Generate();
            Bitmap icon = iig.GetIconImage();
            WritePNG(icon, outfn);
            }
          }
        }
      }

    System.out.println( "Scan:\n" + s.toString() );

    System.out.println( "success" );
    }
}
```

14.112 SimplePrint.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
/*
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/SimplePrint.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java SimplePrint gdcmData/012345.002.050.dcm
 */
import gdcm.*;

public class SimplePrint
{
  public static void RecurseDataSet(File f, DataSet ds, String indent)
    {
    JavaDataSet cds = new JavaDataSet(ds);
    while(!cds.IsAtEnd())
      {
      DataElement de = cds.GetCurrent();
      // Compute VR from the toplevel file, and the currently processed dataset:
      VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

      if( vr.Compatible( new VR(VR.VRType.SQ) ) )
        {
        long uvl = de.GetVL().GetValueLength(); // Test cast is ok
        System.out.println(  indent + de.GetTag().toString() + ":" + uvl ); // why not ?
        //SequenceOfItems sq = de.GetSequenceOfItems();
        // GetValueAsSQ handle more cases than GetSequenceOfItems
        SmartPtrSQ sq = de.GetValueAsSQ();
        long n = sq.GetNumberOfItems();
        for( long i = 1; i <= n; i++) // item starts at 1, not 0
          {
          Item item = sq.GetItem( i );
          DataSet nested = item.GetNestedDataSet();
          RecurseDataSet( f, nested, indent + "  " );
          }
        }
```

```
      else
        {
        System.out.println( indent + de.toString() );
        }
      cds.Next();
      }
    }

  public static void main(String[] args) throws Exception
    {
    String filename = args[0];
    Reader reader = new Reader();
    reader.SetFileName( filename );
    boolean ret = reader.Read();
    if( !ret )
      {
      throw new Exception("Could not read: " + filename );
      }
    File f = reader.GetFile();
    DataSet ds = f.GetDataSet();

    RecurseDataSet( f, ds, "" );
    }
}
```

14.113 AddPrivateAttribute.py

```
00001
00014
00015 """
00016 Usage:
00017
00018   python AddPrivateAttribute.py input.dcm output.dcm
00019
00020
00021 """
00022
00023 import sys
00024 import gdcm
00025
00026 if __name__ == "__main__":
00027
00028   file1 = sys.argv[1]
00029   file2 = sys.argv[2]
00030
00031   r = gdcm.Reader()
00032   r.SetFileName( file1 )
00033   if not r.Read():
00034     sys.exit(1)
00035
00036   f = r.GetFile()
00037   ds = f.GetDataSet()
00038
00039   # Create a dataelement
00040   de = gdcm.DataElement(gdcm.Tag(0x0051, 0x1011))
00041   de.SetByteStringValue("p2")
00042   de.SetVR(gdcm.VR(gdcm.VR.SH))
00043
00044   ds.Insert(de)
00045
00046   w = gdcm.Writer()
00047   w.SetFile( f )
00048   w.SetFileName( file2 )
00049   if not w.Write():
00050     sys.exit(1)
```

14.114 ConvertMPL.py

```
00001
00014
00015 """
00016 display a DICOM image with matPlotLib via numpy
```

```

Caveats:
- Does not support UINT12/INT12

Usage:

 python ConvertNumpy.py "IM000000"

Thanks:
  plotting example - Ray Schumacher 2009
"""

import gdcm
import numpy
from pylab import *


def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
                gdcm.PixelFormat.INT8   :numpy.uint8,
                gdcm.PixelFormat.UINT16 :numpy.uint16,
                gdcm.PixelFormat.INT16  :numpy.int16,
                gdcm.PixelFormat.UINT32 :numpy.uint32,
                gdcm.PixelFormat.INT32  :numpy.int32,
                gdcm.PixelFormat.FLOAT32:numpy.float32,
                gdcm.PixelFormat.FLOAT64:numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat().GetScalarType()
    print 'pf', pf
    print image.GetPixelFormat().GetScalarTypeAsString()
    assert pf in get_gdcm_to_numpy_typemap().keys(), \
           "Unsupported array type %s"%pf
    d = image.GetDimension(0), image.GetDimension(1)
    print 'Image Size: %d x %d' % (d[0], d[1])
    dtype = get_numpy_array_type(pf)
    gdcm_array = image.GetBuffer()

    result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)

    result.shape = d
    return result

if __name__ == "__main__":
  import sys
  r = gdcm.ImageReader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read():  sys.exit(1)
  numpy_array = gdcm_to_numpy( r.GetImage() )

  subplot(111)# one plot, on left
  title(filename)

  imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)

  subplots_adjust(bottom=0.1, right=0.8, top=0.9)
  cax = axes([0.85, 0.1, 0.075, 0.8])
  colorbar(cax=cax)
  title('values')
  get_current_fig_manager().window.title('plot')
  show()
```

14.115 ConvertNumpy.py

```

"""
```

```
This module add support for converting a gdcm.Image to a numpy array.

Caveats:
- Does not support UINT12/INT12

Removed:
- float16 is defined in GDCM API but no implementation exist for it ...
"""

import gdcm
import numpy

def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.uint8,
                gdcm.PixelFormat.INT8   :numpy.int8,
                #gdcm.PixelFormat.UINT12 :numpy.uint12,
                #gdcm.PixelFormat.INT12  :numpy.int12,
                gdcm.PixelFormat.UINT16 :numpy.uint16,
                gdcm.PixelFormat.INT16  :numpy.int16,
                gdcm.PixelFormat.UINT32 :numpy.uint32,
                gdcm.PixelFormat.INT32  :numpy.int32,
                #gdcm.PixelFormat.FLOAT16:numpy.float16,
                gdcm.PixelFormat.FLOAT32:numpy.float32,
                gdcm.PixelFormat.FLOAT64:numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat()

    assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \
           "Unsupported array type %s"%pf

    shape = image.GetDimension(0) * image.GetDimension(1), pf.GetSamplesPerPixel()
    if image.GetNumberOfDimensions() == 3:
      shape = shape[0] * image.GetDimension(2), shape[1]

    dtype = get_numpy_array_type(pf.GetScalarType())
    gdcm_array = image.GetBuffer()
    result = numpy.frombuffer(gdcm_array, dtype=dtype)
    result.shape = shape
    return result

if __name__ == "__main__":
  import sys
  r = gdcm.ImageReader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read():
    sys.exit(1)

  numpy_array = gdcm_to_numpy( r.GetImage() )
  print numpy_array
```

14.116 ConvertPIL.py

```

"""
save a DICOM image with PIL via numpy

Caveats:
- Does not support UINT12/INT12

Usage:

 python ConvertNumpy.py "IM000000"

Thanks:
  plotting example - Ray Schumacher 2009
```

```
"""

import gdcm
import numpy
from PIL import Image, ImageOps


def get_gdcm_to_numpy_typemap():
    """Returns the GDCM Pixel Format to numpy array type mapping."""
    _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
                gdcm.PixelFormat.INT8   :numpy.uint8,
                gdcm.PixelFormat.UINT16 :numpy.uint16,
                gdcm.PixelFormat.INT16  :numpy.int16,
                gdcm.PixelFormat.UINT32 :numpy.uint32,
                gdcm.PixelFormat.INT32  :numpy.int32,
                gdcm.PixelFormat.FLOAT32:numpy.float32,
                gdcm.PixelFormat.FLOAT64:numpy.float64 }
    return _gdcm_np

def get_numpy_array_type(gdcm_pixel_format):
    """Returns a numpy array typecode given a GDCM Pixel Format."""
    return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]

def gdcm_to_numpy(image):
    """Converts a GDCM image to a numpy array.
    """
    pf = image.GetPixelFormat().GetScalarType()
    print 'pf', pf
    print image.GetPixelFormat().GetScalarTypeAsString()
    assert pf in get_gdcm_to_numpy_typemap().keys(), \
           "Unsupported array type %s"%pf
    d = image.GetDimension(0), image.GetDimension(1)
    print 'Image Size: %d x %d' % (d[0], d[1])
    dtype = get_numpy_array_type(pf)
    gdcm_array = image.GetBuffer()
    result = numpy.frombuffer(gdcm_array, dtype=dtype)
    maxV = float(result[result.argmax()])

    result = numpy.log(result+50)
    maxV = float(result[result.argmax()])
    result = result*(2.**8/maxV)
    result.shape = d
    return result

if __name__ == "__main__":
  import sys
  r = gdcm.ImageReader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read():  sys.exit(1)
  numpy_array = gdcm_to_numpy( r.GetImage() )

  pilImage = Image.frombuffer('L',
                              numpy_array.shape,
                              numpy_array.astype(numpy.uint8),
                              'raw','L',0,1)

  pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
  pilImage.save(sys.argv[1]+'.jpg')
```

14.117 CreateRAWStorage.py

```


"""
    <uid value="1.2.840.10008.5.1.4.1.1.66" name="Raw Data Storage" type="SOP Class" part="PS 3.4" retired="false"/>
"""

import gdcm
import sys,os

if __name__ == "__main__":
  r = gdcm.Reader()
  # Will require Testing...
  dataroot = gdcm.Testing.GetDataRoot()
```

```
filename = os.path.join( dataroot, '012345.002.050.dcm' )
r.SetFileName( filename )
r.Read()
f = r.GetFile()
ds = f.GetDataSet()

uid = "1.2.840.10008.5.1.4.1.1.66"
#  f = gdcm.File()
#  ds = f.GetDataSet()
de = gdcm.DataElement( gdcm.Tag(0x0008,0x0016) )
de.SetByteStringValue( uid )
vr = gdcm.VR( gdcm.VR.UI )
de.SetVR( vr )
ds.Replace( de )

ano = gdcm.Anonymizer()
ano.SetFile( r.GetFile() )
ano.RemovePrivateTags()
ano.RemoveGroupLength()
taglist = [
gdcm.Tag(0x0008,0x0008),
gdcm.Tag(0x0008,0x0022),
gdcm.Tag(0x0008,0x0032),
gdcm.Tag(0x0008,0x2111),
gdcm.Tag(0x0008,0x1150),
gdcm.Tag(0x0008,0x1155),
gdcm.Tag(0x0008,0x0100),
gdcm.Tag(0x0008,0x0102),
gdcm.Tag(0x0008,0x0104),
gdcm.Tag(0x0040,0xa170),
gdcm.Tag(0x0008,0x2112),
gdcm.Tag(0x0008,0x0100),
gdcm.Tag(0x0008,0x0102),
gdcm.Tag(0x0008,0x0104),
gdcm.Tag(0x0008,0x9215),
gdcm.Tag(0x0018,0x0010),
gdcm.Tag(0x0018,0x0022),
gdcm.Tag(0x0018,0x0050),
gdcm.Tag(0x0018,0x0060),
gdcm.Tag(0x0018,0x0088),
gdcm.Tag(0x0018,0x0090),
gdcm.Tag(0x0018,0x1040),
gdcm.Tag(0x0018,0x1100),
gdcm.Tag(0x0018,0x1110),
gdcm.Tag(0x0018,0x1111),
gdcm.Tag(0x0018,0x1120),
gdcm.Tag(0x0018,0x1130),
gdcm.Tag(0x0018,0x1150),
gdcm.Tag(0x0018,0x1151),
gdcm.Tag(0x0018,0x1152),
gdcm.Tag(0x0018,0x1160),
gdcm.Tag(0x0018,0x1190),
gdcm.Tag(0x0018,0x1210),
gdcm.Tag(0x0020,0x0012),
gdcm.Tag(0x0020,0x0032),
gdcm.Tag(0x0020,0x0037),
gdcm.Tag(0x0020,0x1041),
gdcm.Tag(0x0020,0x4000),
gdcm.Tag(0x0028,0x0002),
gdcm.Tag(0x0028,0x0004),
gdcm.Tag(0x0028,0x0010),
gdcm.Tag(0x0028,0x0011),
gdcm.Tag(0x0028,0x0030),
gdcm.Tag(0x0028,0x0100),
gdcm.Tag(0x0028,0x0101),
gdcm.Tag(0x0028,0x0102),
gdcm.Tag(0x0028,0x0103),
gdcm.Tag(0x0028,0x1052),
gdcm.Tag(0x0028,0x1053),
gdcm.Tag(0x0028,0x2110),
gdcm.Tag(0x0028,0x2112),
gdcm.Tag(0x7fe0,0x0010),
gdcm.Tag(0x0018,0x0020),
gdcm.Tag(0x0018,0x0021),
gdcm.Tag(0x0018,0x0023),
gdcm.Tag(0x0018,0x0025),
gdcm.Tag(0x0018,0x0080),
gdcm.Tag(0x0018,0x0081),
gdcm.Tag(0x0018,0x0083),
gdcm.Tag(0x0018,0x0084),
gdcm.Tag(0x0018,0x0085),
```

```
  gdcm.Tag(0x0018,0x0086),
  gdcm.Tag(0x0018,0x0087),
  gdcm.Tag(0x0018,0x0091),
  gdcm.Tag(0x0018,0x0093),
  gdcm.Tag(0x0018,0x0094),
  gdcm.Tag(0x0018,0x0095),
  gdcm.Tag(0x0018,0x1088),
  gdcm.Tag(0x0018,0x1090),
  gdcm.Tag(0x0018,0x1094),
  gdcm.Tag(0x0018,0x1250),
  gdcm.Tag(0x0018,0x1251),
  gdcm.Tag(0x0018,0x1310),
  gdcm.Tag(0x0018,0x1312),
  gdcm.Tag(0x0018,0x1314),
  gdcm.Tag(0x0018,0x1315),
  gdcm.Tag(0x0018,0x1316),
  gdcm.Tag(0x0020,0x0110),
  gdcm.Tag(0x0028,0x0120),
  gdcm.Tag(0x0028,0x1050),
  gdcm.Tag(0x0028,0x1051)
  ]
  for tag in taglist:
    #print tag
    ano.Remove( tag )

  # special handling
  gen = gdcm.UIDGenerator()
  ano.Replace( gdcm.Tag(0x0008,0x9123), gen.Generate() )
  #ano.Empty( gdcm.Tag(0x0040,0x0555) )


#
#  uid = gen.Generate()
#  de.SetTag( gdcm.Tag(0x0008,0x0018) )
#  de.SetByteStringValue( uid )
#  ds.Insert( de )

  # init FMI now:
  #fmi = f.GetHeader()
  #ts = gdcm.TransferSyntax()
  #print ts
  #fmi.SetDataSetTransferSyntax( ts ) # default
  #print fmi.GetDataSetTransferSyntax()
  #de.SetTag( gdcm.Tag(0x0002,0x0010) )
  #uid = "1.2.840.10008.1.2"
  #de.SetByteStringValue( uid )
  #fmi.Insert( de )
#  f.SetHeader( r.GetFile().GetHeader() )

  writer = gdcm.Writer()
  writer.SetFile( ano.GetFile() )
  writer.SetFileName( "rawstorage.dcm" );
  writer.Write()
```

14.118 DecompressImage.py

```


"""
Usage:

 python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
"""

import gdcm
import sys

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.ImageReader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)
```

```

  # check GetFragment API:
  pd = r.GetFile().GetDataSet().GetDataElement(gdcm.Tag(0x7fe0, 0x0010))
  frags = pd.GetSequenceOfFragments();
  frags.GetFragment(0);

  ir = r.GetImage()
  w = gdcm.ImageWriter()
  image = w.GetImage()

  image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
  dims = ir.GetDimensions();
  print ir.GetDimension(0);
  print ir.GetDimension(1);
  print "Dims:",dims

  #  Just for fun:
  dircos =  ir.GetDirectionCosines()
  t = gdcm.Orientation.GetType(tuple(dircos))
  l = gdcm.Orientation.GetLabel(t)
  print "Orientation label:",l

  image.SetDimension(0, ir.GetDimension(0) );
  image.SetDimension(1, ir.GetDimension(1) );

  pixeltype = ir.GetPixelFormat();
  image.SetPixelFormat( pixeltype );

  pi = ir.GetPhotometricInterpretation();
  image.SetPhotometricInterpretation( pi );

  pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
  str1 = ir.GetBuffer()
  #print ir.GetBufferLength()
  pixeldata.SetByteStringValue( str1 )
  image.SetDataElement( pixeldata )

  w.SetFileName( file2 )
  w.SetFile( r.GetFile() )
  w.SetImage( image )
  if not w.Write():
    sys.exit(1)
```

14.119 DumbAnonymizer.py

```

"""
This example shows how one can use the gdcm.Anonymizer in 'dumb' mode.
This class becomes really handy when one knows which particular tag to fill in.

Usage:

 python DumbAnonymizer.py gdcmData/012345.002.050.dcm out.dcm

"""

import gdcm

# http://www.oid-info.com/get/1.3.6.1.4.17434
THERALYS_ORG_ROOT = "1.3.6.1.4.17434"

tag_rules={
  # Value
  (0x0012,0x0010):("Value","MySponsorName"),
  (0x0012,0x0020):("Value","MyProtocolID"),
  (0x0012,0x0021):("Value","MyProtocolName"),
  (0x0012,0x0062):("Value","YES"),
  (0x0012,0x0063):("Value","MyDeidentificationMethod"),

  # Method
  #(0x0002,0x0003):("Method","GenerateMSOPId"),
  #(0x0008,0x1155):("Method","GenerateMSOPId"),
  (0x0008,0x0018):("Method","GenerateMSOPId"),
  (0x0010,0x0010):("Method","GetSponsorInitials"),
  (0x0010,0x0020):("Method","GetSponsorId"),
```

```
  (0x0012,0x0030):("Method","GetSiteId"),
  (0x0012,0x0031):("Method","GetSiteName"),
  (0x0012,0x0040):("Method","GetSponsorId"),
  (0x0012,0x0050):("Method","GetTPId"),
  (0x0018,0x0022):("Method","KeepIfExist"),
  (0x0018,0x1315):("Method","KeepIfExist"),
  (0x0020,0x000d):("Method","GenerateStudyId"),
  (0x0020,0x000e):("Method","GenerateSeriesId"),
  (0x0020,0x1002):("Method","GetNumberOfFrames"),
  (0x0020,0x0020):("Method","GetPatientOrientation"),
  # Other:
  (0x0012,0x0051):("Patient Field","Type Examen"),
  (0x0018,0x1250):("Sequence Field","Receive Coil"),
  (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
  (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
  (0x0018,0x0082):("Sequence Field","Inversion Time"),
}

class MyAnon:
  def __init__(self):
    self.studyuid = None
    self.seriesuid = None
    generator = gdcm.UIDGenerator()
    if not self.studyuid:
      self.studyuid = generator.Generate()
    if not self.seriesuid:
      self.seriesuid = generator.Generate()
  def GetSponsorInitials(self):
    return "dummy^foobar"
  def GenerateStudyId(self):
    return self.studyuid
  def GenerateSeriesId(self):
    return self.seriesuid
  #def GenerateMSOPId(self):
  def GenerateMSOPId(self):
    generator = gdcm.UIDGenerator()
    return generator.Generate()
  def GetSiteId(self):
    return "MySiteId"
  def GetSiteName(self):
    return "MySiteName"
  def GetSponsorId(self):
    return "MySponsorId"
  def GetTPId(self):
    return "MyTP"

if __name__ == "__main__":
  import sys
  gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "DumbAnonymizer" )
  gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )

  r = gdcm.Reader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read(): sys.exit(1)

  obj = MyAnon()

  w = gdcm.Writer()
  ano = gdcm.Anonymizer()
  ano.SetFile( r.GetFile() )
  ano.RemoveGroupLength()
  for tag,rule in tag_rules.items():
    if rule[0] == 'Value':
      print tag,rule
      ano.Replace( gdcm.Tag( tag[0], tag[1] ), rule[1] )
    elif rule[0] == 'Method':
      print tag,rule
      # result = locals()[rule[1]]()
      methodname = rule[1]
      if hasattr(obj, methodname):
        _member = getattr(obj, methodname)
        result = _member()
        ano.Replace( gdcm.Tag( tag[0], tag[1] ), result )
      else:
        print "Problem with: ", methodname

  outfilename = sys.argv[2]
  w.SetFileName( outfilename )
  w.SetFile( ano.GetFile() )
  if not w.Write(): sys.exit(1)
```

14.120 ExtractImageRegion.py

```python


"""

 This small code shows how to use the gdcm.ImageRegionReader API
 In this example we are taking each frame by frame and dump them to
 /tmp/frame.raw.

 Usage:
 $ ExtractImageRegion.py input.dcm

 Example:
 $ ExtractImageRegion.py gdcmData/012345.002.050.dcm
 $ md5sum /tmp/frame.raw
 d594a5e2fde12f32b6633ca859b4d4a6  /tmp/frame.raw
 $ gdcminfo --md5sum gdcmData/012345.002.050.dcm
 [...]
 md5sum: d594a5e2fde12f32b6633ca859b4d4a6
"""

import gdcm

if __name__ == "__main__":
  import sys
  filename = sys.argv[1]

  file_size = gdcm.System.FileSize(filename);

  # instantiate the reader:
  reader = gdcm.ImageRegionReader();
  reader.SetFileName( filename );

  # pull DICOM info:
  if not reader.ReadInformation():
    sys.exit(1)

  # store current offset:
  cur_pos = reader.GetStreamCurrentPosition();

  remaining = file_size - cur_pos;

  print("Remaining bytes to read (Pixel Data): %d" % remaining );

  # Get file infos
  f = reader.GetFile();

  # get some info about image
  dims = gdcm.ImageHelper.GetDimensionsValue(f);
  print(dims)
  pf = gdcm.ImageHelper.GetPixelFormatValue (f);
  pixelsize = pf.GetPixelSize();
  pi = gdcm.ImageHelper.GetPhotometricInterpretationValue(f);
  print( pi );

  # buffer to get the pixels
  buffer = bytearray( dims[0] * dims[1] * pixelsize )

  # define a simple box region.
  box = gdcm.BoxRegion();
  for z in range(0, dims[2]):
    # Define that I want the image 0, full size (dimx x dimy pixels)
    # and do that for each z:
    box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
    #print( box.toString() );
    reader.SetRegion( box );

    # reader will try to load the uncompressed image region into buffer.
    # the call returns an error when buffer.Length is too small. For instance
    # one can call:
    # uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
    # to get the exact size of minimum buffer
    if reader.ReadIntoBuffer(buffer):
      open('/tmp/frame.raw', 'wb').write(buffer)
    else:
      #throw new Exception("can't read pixels error");
      sys.exit(1)
```

14.121 FindAllPatientName.py

```

"""
This example shows how one can use the gdcm.CompositeNetworkFunctions class
for executing a C-FIND query
It will print the list of patient name found

Usage:

 python FindAllPatientName.py

"""

import gdcm

# Patient Name
tag = gdcm.Tag(0x10,0x10)
de = gdcm.DataElement(tag)

# Search all patient name where string match 'F*'
de.SetByteStringValue('F*')

ds = gdcm.DataSet()
ds.Insert(de)

cnf = gdcm.CompositeNetworkFunctions()
theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)

#print theQuery.ValidateQuery()

# prepare the variable for output
ret = gdcm.DataSetArrayType()

# Execute the C-FIND query
cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')

for i in range(0,ret.size()):
  print "Patient #",i
  print ret[i]
```

14.122 FixCommaBug.py

```


"""
Using LC_NUMERIC set to something not compatible with "C" it is possible to write out "," instead of
"." as required by the DICOM standard
Issue is still current (IMHO) with gdcm 2.0.9
"""

import gdcm
import sys

filename = sys.argv[1]
outname = sys.argv[2]

# read
r = gdcm.Reader()
r.SetFileName( filename )
if not r.Read():
  print "not valid"
  sys.exit(1)

file = r.GetFile()
dataset = file.GetDataSet()

ano = gdcm.Anonymizer()
ano.SetFile( file )

tags = [
gdcm.Tag(0x0018,0x1164),
gdcm.Tag(0x0018,0x0088),
gdcm.Tag(0x0018,0x0050),
gdcm.Tag(0x0028,0x0030),
]
```

```

for tag in tags:
  print tag
  if dataset.FindDataElement( tag ):
    pixelspacing = dataset.GetDataElement( tag )
    #print pixelspacing
    bv = pixelspacing.GetByteValue()
    str = bv.GetBuffer()
    #print bv.GetLength()
    #print len(str)
    new_str = str.replace(",",".")
    # Need to explicitly pass bv.GetLength() to remove any trailing garbage
    ano.Replace( tag, new_str, bv.GetLength() )

#print dataset

w = gdcm.Writer()
w.SetFile( file )
w.SetFileName( outname )
if not w.Write():
  print "Cannot write"
  sys.exit(1)

# paranoid:
image_reader = gdcm.ImageReader()
image_reader.SetFileName( outname )
if not image_reader.Read():
  print "there is still a comma"
  sys.exit(1)

print "Success!"
sys.exit(0) # success
```

14.123 GetPortionCSAHeader.py

```


"""
Usage:

 python GetPortionCSAHeader.py input.dcm

Footnote:
  SIEMENS is not publishing any information on the CSA header. So any info extracted
  is at your own risk.
"""

import sys
import gdcm

if __name__ == "__main__":

  file = sys.argv[1]

  r = gdcm.Reader()
  r.SetFileName( file )
  if not r.Read():
    sys.exit(1)

  ds = r.GetFile().GetDataSet()
  csa_t1 = gdcm.CSAHeader()
  csa_t2 = gdcm.CSAHeader()
  #print csa
  t1 = csa_t1.GetCSAImageHeaderInfoTag();
  print t1
  t2 = csa_t2.GetCSASeriesHeaderInfoTag();
  print t2
  # Let's do it for t1:
  if ds.FindDataElement( t1 ):
    csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
    print csa_t1

  # Now let's pretend we are only interested in B_value and DiffusionGradientDirection entries:
  bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case sensitive !
  print bvalues

```

```
  diffgraddir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) # WARNING: it is case sensitive
!
  print diffgraddir

  # repeat for t2 if you like it:
  if ds.FindDataElement( t2 ):
    csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
    # print csa_t2

  gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
  print gdt

  bv = gdt.GetByteValue();
  #print bv
  str = bv.GetPointer()
  print str.split("\\")
```

14.124 HelloWorld.py

```


"""
Hello World !
"""

import gdcm
import sys

if __name__ == "__main__":

  # verbosity:
  #gdcm.Trace.DebugOn()
  #gdcm.Trace.WarningOn()
  #gdcm.Trace.ErrorOn()

  # Get the filename from the command line
  filename = sys.argv[1]

  # Instantiate a gdcm.Reader
  # This is the main class to handle any type of DICOM object
  # You should check for gdcm.ImageReader for reading specifically DICOM Image file
  r = gdcm.Reader()
  r.SetFileName( filename )
  # If the reader fails to read the file, we should stop !
  if not r.Read():
    print "Not a valid DICOM file"
    sys.exit(1)

  # Get the DICOM File structure
  file = r.GetFile()

  # Get the DataSet part of the file
  dataset = file.GetDataSet()

  # Ok let's print it !
  print dataset

  # Use StringFilter to print a particular Tag:
  sf = gdcm.StringFilter()
  sf.SetFile(r.GetFile())

  # Check if Attribute exist
  print dataset.FindDataElement( gdcm.Tag(0x0028,0x0010))

  # Let's print it as string pair:
  print sf.ToStringPair(gdcm.Tag(0x0028,0x0010))
```

14.125 ManipulateFile.py

```


"""
```

```
Usage:

 python ManipulateFile.py input.dcm output.dcm

Footnote:
  GDCM 1.2.x would create incorrect Multiframes MR Image Storage file. Try to recover from
  the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage file.
  e.g:

  python ManipulateFile.py Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  ano = gdcm.Anonymizer()
  ano.SetFile( r.GetFile() )
  ano.RemovePrivateTags()
  ano.Remove( gdcm.Tag(0x0032,0x1030) )
  ano.Remove( gdcm.Tag(0x008,0x14) )
  ano.Remove( gdcm.Tag(0x008,0x1111) )
  ano.Remove( gdcm.Tag(0x008,0x1120) )
  ano.Remove( gdcm.Tag(0x008,0x1140) )
  ano.Remove( gdcm.Tag(0x10,0x21b0) )
  ano.Empty( gdcm.Tag(0x10,0x10) )
  ano.Empty( gdcm.Tag(0x10,0x20) )
  ano.Empty( gdcm.Tag(0x10,0x30) )
  ano.Empty( gdcm.Tag(0x20,0x10) )
  ano.Empty( gdcm.Tag(0x32,0x1032) )
  ano.Empty( gdcm.Tag(0x32,0x1033) )
  ano.Empty( gdcm.Tag(0x40,0x241) )
  ano.Empty( gdcm.Tag(0x40,0x254) )
  ano.Empty( gdcm.Tag(0x40,0x253) )
  ano.Empty( gdcm.Tag(0x40,0x1001) )
  ano.Empty( gdcm.Tag(0x8,0x80) )
  ano.Empty( gdcm.Tag(0x8,0x50) )
  ano.Empty( gdcm.Tag(0x8,0x1030) )
  ano.Empty( gdcm.Tag(0x8,0x103e) )
  ano.Empty( gdcm.Tag(0x18,0x1030) )
  ano.Empty( gdcm.Tag(0x38,0x300) )
  g = gdcm.UIDGenerator()
  ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
  ano.Replace( gdcm.Tag(0x0020,0x00d), g.Generate() )
  ano.Replace( gdcm.Tag(0x0020,0x00e), g.Generate() )
  ano.Replace( gdcm.Tag(0x0020,0x052), g.Generate() )
  #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
  """
  ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
  ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
  ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
  ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
  ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
  ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
  ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
  ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
  ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
  ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing

  ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
  ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
  ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
  ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator

  ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation

  ano.Empty( gdcm.Tag(0x0020,0x0020) )

  ano.Remove( gdcm.Tag(0x7fe0,0x0000) )

  #ano.Empty( gdcm.Tag(0x0028,0x0009) ) # Frame Increment Pointer

```

```
  #ano.Empty( gdcm.Tag(0x0028,0x1052) )  #<entry group="0028" element="1052" vr="DS" vm="1" name="Rescale Intercept"/>
  #ano.Empty( gdcm.Tag(0x0028,0x1053) )  #<entry group="0028" element="1053" vr="DS" vm="1" name="Rescale Slope"/>
  #ano.Replace( gdcm.Tag(0x0028,0x1054), "US" )  #<entry group="0028" element="1054" vr="LO" vm="1" name="Rescale Type"/>

  ano.Replace( gdcm.Tag(0x2050, 0x0020), "IDENTITY")
  """

  w = gdcm.Writer()
  w.SetFile( ano.GetFile() )
  w.SetFileName( file2 )
  if not w.Write():
    sys.exit(1)
```

14.126 ManipulateSequence.py

```


"""
Usage:

 python ManipulateSequence.py input.dcm output.dcm

This was tested using:

 python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm

This is a dummy example on how to modify a value set in a nested-nested dataset

WARNING:
Do not use as-is in production, this is just an example
This example works in an undefined length Item only (you need to explicitly recompute the length otherwise)
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  f = r.GetFile()
  ds = f.GetDataSet()
  tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
  if ds.FindDataElement( tsis ):
    sis = ds.GetDataElement( tsis )
    #sqsis = sis.GetSequenceOfItems()
    # GetValueAsSQ handle more cases
    sqsis = sis.GetValueAsSQ()
    if sqsis.GetNumberOfItems():
      item1 = sqsis.GetItem(1)
      nestedds = item1.GetNestedDataSet()
      tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
      if nestedds.FindDataElement( tprcs ):
        prcs = nestedds.GetDataElement( tprcs )
        sqprcs = prcs.GetSequenceOfItems()
        if sqprcs.GetNumberOfItems():
          item2 = sqprcs.GetItem(1)
          nestedds2 = item2.GetNestedDataSet()
          # (0008,0104) LO [Uncompressed predecessor]               #  24, 1 CodeMeaning
          tcm = gdcm.Tag(0x0008,0x0104)
          if nestedds2.FindDataElement( tcm ):
            cm = nestedds2.GetDataElement( tcm )
            mystr = "GDCM was here"
            cm.SetByteStringValue( mystr )

  w = gdcm.Writer()
```

```
  w.SetFile( f )
  w.SetFileName( file2 )
  if not w.Write():
    sys.exit(1)
```

14.127 MergeFile.py

```

"""
Usage:

 python MergeFile.py input1.dcm input2.dcm

  It will produce a 'merge.dcm' output file, which contains all meta information from input1.dcm
  and copy the Stored Pixel values from input2.dcm
  This script even works when input2.dcm is a Secondary Capture and does not contains information
  such as IOP and IPP...
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r1 = gdcm.ImageReader()
  r1.SetFileName( file1 )
  if not r1.Read():
    sys.exit(1)

  r2 = gdcm.ImageReader()
  r2.SetFileName( file2 )
  if not r2.Read():
    sys.exit(1)

  # Image from r2 could be Secondary Capture and thus would not contains neither IPP nor IOP
  # Instead always prefer to only copy the Raw Data Element.
  # Warning ! Image need to be identical ! Only the value of Stored Pixel can be different.
  r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )

  w = gdcm.ImageWriter()
  w.SetFile( r1.GetFile() )
  #w.SetImage( r2.GetImage() )  # See comment above
  w.SetImage( r1.GetImage() )

  w.SetFileName( "merge.dcm" )
  if not w.Write():
    sys.exit(1)

  sys.exit(0)
```

14.128 NewSequence.py

```

"""
Usage:

  python NewSequence.py input.dcm output.dcm


Thanks to Robert Irie for code
"""

import sys
import gdcm

if __name__ == "__main__":

```

```
file1 = sys.argv[1]
file2 = sys.argv[2]

r = gdcm.Reader()
r.SetFileName( file1 )
if not r.Read():
  sys.exit(1)

f = r.GetFile()
ds = f.GetDataSet()
#tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence

# Create a dataelement
de = gdcm.DataElement(gdcm.Tag(0x0010, 0x2180))
de.SetByteStringValue("Occupation")
de.SetVR(gdcm.VR(gdcm.VR.SH))

# Create an item
it=gdcm.Item()
it.SetVLToUndefined()      # Needed to not popup error message
#it.InsertDataElement(de)
nds=it.GetNestedDataSet()
nds.Insert(de)

# Create a Sequence
sq=gdcm.SequenceOfItems().New()
sq.SetLengthToUndefined()
sq.AddItem(it)

# Insert sequence into data set
des=gdcm.DataElement(gdcm.Tag(0x0400,0x0550))
des.SetVR(gdcm.VR(gdcm.VR.SQ))
des.SetValue(sq.__ref__())
des.SetVLToUndefined()

ds.Insert(des)

w = gdcm.Writer()
w.SetFile( f )
w.SetFileName( file2 )
if not w.Write():
  sys.exit(1)
```

14.129 PhilipsPrivateRescaleInterceptSlope.py

```

"""
Usage:

 python
"""

import gdcm
import sys

filename = sys.argv[1]
tmpfile = "/tmp/philips_rescaled.dcm"


# Need to access some private tags, read the file :
reader = gdcm.Reader()
reader.SetFileName( filename )
if not reader.Read():
  sys.exit(1)

ds = reader.GetFile().GetDataSet()

#print ds
# (2005,1409)     DS      4       0.0
# (2005,140a)     DS      16      1.52283272283272

# (2005,0014)     LO      26      Philips MR Imaging DD 005
tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
print tag1
```

```
print tag2

# make sure to do a copy, we want the private tag to remain
# otherwise gdcm gives us a reference
el1 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
print el1
el2 = gdcm.DataElement( ds.GetDataElement( tag2 ) )
print el2

# (0028,1052) DS [-1000]                                    #   6, 1 RescaleIntercept
# (0028,1053) DS [1]                                        #   2, 1 RescaleSlope

el1.SetTag( gdcm.Tag(0x0028,0x1052) )
el2.SetTag( gdcm.Tag(0x0028,0x1053) )

ds.Insert( el1 )
ds.Insert( el2 )

w = gdcm.Writer()
w.SetCheckFileMetaInformation( False )
w.SetFileName( tmpfile )
w.SetFile( reader.GetFile() )
if not w.Write():
  sys.exit(1)

print "success"
```

14.130 PlaySound.py

```


"""
Usage:

 python PlaySound.py input.dcm
"""

import gdcm
import sys

#filename = "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/audio_from_rafael_sanguinetti.dcm"
filename = sys.argv[1]
print filename

r = gdcm.Reader()
r.SetFileName( filename )
if not r.Read():
  sys.exit(1)

ds = r.GetFile().GetDataSet()

waveformtag = gdcm.Tag(0x5400,0x0100)
waveformsq = ds.GetDataElement( waveformtag )
#print waveformsq

#print dir(waveformsq)

items = waveformsq.GetSequenceOfItems()

if not items.GetNumberOfItems():
  sys.exit(1)

item = items.GetItem(1)
#print item

waveformds = item.GetNestedDataSet()
#print waveformds

waveformdatatag = gdcm.Tag(0x5400,0x1010)
waveformdata = waveformds.GetDataElement( waveformdatatag )

#print waveformdata.GetPointer()
bv = waveformdata.GetByteValue()
print dir(bv)

#print bv.GetPointer()
```

```
print bv.GetLength()
l = 116838

file='test.wav'
myfile = open(file, "wb")
s = bv.GetPointer()
for i in range(0, l):
  myfile.write(s[i])
myfile.close()

# http://mail.python.org/pipermail/python-list/2004-October/288905.html
if sys.platform.startswith('win'):
   from winsound import PlaySound, SND_FILENAME, SND_ASYNC
   PlaySound(file, SND_FILENAME|SND_ASYNC)
elif sys.platform.find('linux')>-1:
   from wave import open as waveOpen
   from ossaudiodev import open as ossOpen
   s = waveOpen(file,'rb')
   (nc,sw,fr,nf,comptype, compname) = s.getparams( )
   dsp = ossOpen('/dev/dsp','w')
   try:
     from ossaudiodev import AFMT_S16_NE
   except ImportError:
     if byteorder == "little":
       AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
     else:
       AFMT_S16_NE = ossaudiodev.AFMT_S16_BE
   dsp.setparameters(AFMT_S16_NE, nc, fr)
   data = s.readframes(nf)
   s.close()
   dsp.write(data)
   dsp.close()
```

14.131 PrivateDict.py

```
"""
"""

import gdcm
import sys,os

if __name__ == "__main__":
  #gdcm.Trace.DebugOn()
  globInst = gdcm.Global.GetInstance()
  # Try to load Part3.xml file
  # This file is too big for being accessible directly at runtime.
  globInst.LoadResourcesFiles()


  # Get a private tag from the runtime dicts. LoadResourcesFiles could
  # have failed but this has no impact on the private dict

  d = globInst.GetDicts()
  print d.GetDictEntry( gdcm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
  pd = d.GetPrivateDict()
  print pd.GetDictEntry( gdcm.PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER") )
```

14.132 ReWriteSCAsMR.py

```
"""
GDCM 1.x would write out MR Image Storage as Secondary Capture Object while still setting Rescale
Slope/Intercept
and saving the Pixel Spacing in (0028,0030)
"""

import gdcm
import sys,os

```

```
def CheckSecondaryCaptureObjectIsMRImageStorage(r):
  ds = r.GetFile().GetDataSet()
  # Check Source Image Sequence
  if ds.FindDataElement( gdcm.Tag(0x0008,0x2112) ):
    sis = ds.GetDataElement( gdcm.Tag(0x0008,0x2112) )
    sqsis = sis.GetSequenceOfItems()
    if sqsis.GetNumberOfItems():
      item1 = sqsis.GetItem(1)
      nestedds = item1.GetNestedDataSet()
      if nestedds.FindDataElement( gdcm.Tag(0x0008,0x1150) ):
        ReferencedSOPClassUID = nestedds.GetDataElement( gdcm.Tag(0x0008,0x1150) )
        raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
        uids = gdcm.UIDs()
        # what is the actual object we are looking at ?
        ms = gdcm.MediaStorage()
        ms.SetFromDataSet(ds)
        msuid = ms.GetString()
        uids.SetFromUID( msuid )
        msuidname = uids.GetName() # real Media Storage Name
        uids.SetFromUID( raw )
        sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
        # If object is SC and Source derivation is MRImageStorage then we can assume 'Pixel Spacing' is correct
        if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary Capture Image Storage' ):
          return True
  # in all other case simply return the currentspacing:
  return False

if __name__ == "__main__":
  r = gdcm.ImageReader()
  filename = sys.argv[1]
  r.SetFileName( filename )
  if not r.Read():
    sys.exit(1)
  f = r.GetFile()

  if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
    # Special handling of the spacing:
    # GDCM 1.2.0 would not rewrite correctly DICOM Object and would always set them as 'Secondary Capture Image Storage'
    # while we would rather have 'MR Image Storage'
    gdcm.ImageHelper.SetForcePixelSpacing( True )
    mrspacing = gdcm.ImageHelper.GetSpacingValue( r.GetFile() )
    # TODO: I cannot do simply the following:
    #image.SetSpacing( mrspacing )
    image.SetSpacing(0, mrspacing[0] )
    image.SetSpacing(1, mrspacing[1] )
    image.SetSpacing(2, mrspacing[2] )
    gdcm.ImageHelper.SetForceRescaleInterceptSlope( True )
    ris = gdcm.ImageHelper.GetRescaleInterceptSlopeValue( r.GetFile() )
    image.SetIntercept( ris[0] )
    image.SetSlope( ris[1] )

  outfilename = sys.argv[2]
  w = gdcm.ImageWriter()
  w.SetFileName( outfilename )
  w.SetFile( r.GetFile() )
  w.SetImage( image )
  if not w.Write():
    sys.exit(1)

  sys.exit(0)
```

14.133 ReadAndDumpDICOMDIR.py

```
import sys
import gdcm

if __name__ == "__main__":
    # Check arguments
    if (len(sys.argv) < 2):
        # No filename passed
```

```
        print "No input filename found"
        quit()

    filename = sys.argv[1]


    # Read file
    reader = gdcm.Reader()
    reader.SetFileName(filename)
    if (not reader.Read()):
        print "Unable to read %s" % (filename)
        quit()

    file = reader.GetFile()

    # Retrieve header information
    fileMetaInformation = file.GetHeader()
    print fileMetaInformation

    # Retrieve data set
    dataSet = file.GetDataSet()
    #print dataSet

    # Check media storage
    mediaStorage = gdcm.MediaStorage()
    mediaStorage.SetFromFile(file)
    if (gdcm.MediaStorage.GetMSType(str(mediaStorage)) != gdcm.MediaStorage.MediaStorageDirectoryStorage):
        # File is not a DICOMDIR
        print "This file is not a DICOMDIR (Media storage type: %s)" % (str(mediaStorage))
        quit()

    # Check Media Storage SOP Class
    if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
        sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0x0002, 0x0002)).GetValue())
        # Check SOP UID
        if (sopClassUid != "1.2.840.10008.1.3.10"):
            # File is not a DICOMDIR
            print "This file is not a DICOMDIR"
    else:
        # Not present
        print "Media Storage SOP Class not present"
        quit()

    # Iterate through the DICOMDIR data set
    iterator = dataSet.GetDES().begin()
    while (not iterator.equal(dataSet.GetDES().end())):
        dataElement = iterator.next()

        # Check the element tag
        if (dataElement.GetTag() == gdcm.Tag(0x004, 0x1220)):
            # The 'Directory Record Sequence' element
            sequence = dataElement.GetValueAsSQ()

            # Loop through the sequence items
            itemNr = 1
            while (itemNr < sequence.GetNumberOfItems()):
                item = sequence.GetItem(itemNr)

                # Check the element tag
                if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
                    # The 'Directory Record Type' element
                    value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())

                    # PATIENT
                    while (value.strip() == "PATIENT"):
                        print value.strip()
                        # Print patient name
                        if (item.FindDataElement(gdcm.Tag(0x0010, 0x0010))):
                            value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0010)).GetValue())
                            print value

                        # Print patient ID
                        if (item.FindDataElement(gdcm.Tag(0x0010, 0x0020))):
                            value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0020)).GetValue())
                            print value

                        # Next
                        itemNr = itemNr + 1
                        item = sequence.GetItem(itemNr)
                        if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
                            value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
```

```

                        # STUDY
                        while (value.strip() == "STUDY"):
                            print value.strip()

                            # Print study UID
                            if (item.FindDataElement(gdcm.Tag(0x0020, 0x000d))):
                                value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000d)).GetValue())
                                print value

                            # Print study date
                            if (item.FindDataElement(gdcm.Tag(0x0008, 0x0020))):
                                value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0020)).GetValue())
                                print value

                            # Print study description
                            if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
                                value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue())
                                print value

                            # Next
                            itemNr = itemNr + 1
                            item = sequence.GetItem(itemNr)
                            if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
                                    value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())

                            # SERIES
                            while (value.strip() == "SERIES"):
                                print value.strip()

                                # Print series UID
                                if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
                                    value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).GetValue())
                                    print value

                                # Print series modality
                                if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
                                    value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060)).GetValue())
                                    print "Modality"
                                    print value

                                # Print series description
                                if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
                                    value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e)).GetValue())
                                    print "Description"
                                    print value

                                # Next
                                itemNr = itemNr + 1
                                item = sequence.GetItem(itemNr)
                                if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
                                    value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())

                                # IMAGE
                                while (value.strip() == "IMAGE"):
                                    print value.strip()

                                    # Print image UID
                                    if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
                                        value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1511)).GetValue())
                                        print value

                                    # Next
                                    if (itemNr < sequence.GetNumberOfItems()):
                                        itemNr = itemNr + 1
                                    else:
                                        break

                                    item = sequence.GetItem(itemNr)
                                    if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
                                            value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())

                    # Next
                    itemNr = itemNr + 1
```

14.134 RemovePrivateTags.py

```
"""
Usage:

 python RemovePrivateTags.py input.dcm output.dcm
"""

import sys
import gdcm


if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  # Instantiate the reader.
  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  # Remove private tags
  ano = gdcm.Anonymizer()
  ano.SetFile( r.GetFile() )
  if not ano.RemovePrivateTags():
    sys.exit(1)

  # Write DICOM file
  w = gdcm.Writer()
  w.SetFile( ano.GetFile() )
  #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
  w.SetFileName( file2 )
  if not w.Write():
    sys.exit(1)

  # It is usually a good idea to exit the script with an error, as gdcm does not remove partial (incorrect) DICOM file
  # (application level)
```

14.135 ScanDirectory.py

```
import gdcm
import sys,os

class ProgressWatcher(gdcm.SimpleSubjectWatcher):
  def ShowProgress(self, sender, event):
    pe = gdcm.ProgressEvent.Cast(event)
    print pe.GetProgress()
  def EndFilter(self):
    print "Yay ! I am done"

if __name__ == "__main__":
  directory = sys.argv[1]

  # Define the set of tags we are interested in
  t1 = gdcm.Tag(0x8,0x8);
  t2 = gdcm.Tag(0x10,0x10);

  # Iterate over directory
  d = gdcm.Directory();
  nfiles = d.Load( directory );
  if(nfiles == 0): sys.exit(1);
  # System.Console.WriteLine( "Files:\n" + d.toString() );

  filenames = d.GetFilenames()

  #  Get rid of any Warning while parsing the DICOM files
  gdcm.Trace.WarningOff()

  # instantiate Scanner:
```

```
  sp = gdcm.Scanner.New();
  s = sp.__ref__()
  w = ProgressWatcher(s, 'Watcher')

  s.AddTag( t1 );
  s.AddTag( t2 );
  b = s.Scan( filenames );
  if(not b): sys.exit(1);

  print "success" ;
  #print s

  pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
  pttv.Start()
  # iterate until the end:
  while( not pttv.IsAtEnd() ):
    # get current value for tag and associated value:
    # if tag was not found, then it was simply not added to the internal std::map
    # Warning value can be None
    tag = pttv.GetCurrentTag()
    value = pttv.GetCurrentValue()
    print tag,"->",value
    # increment iterator
    pttv.Next()

  sys.exit(0)
```

14.136 SortImage.py

```

"""
Usage:

 python SortImage.py dirname
"""

import gdcm
import sys

def PrintProgress(object, event):
  assert event == "ProgressEvent"
  print "Progress:", object.GetProgress()

def MySort(ds1, ds2):
  # compare ds1
  return False

if __name__ == "__main__":

  dirname = sys.argv[1]
  d = gdcm.Directory()
  d.Load( dirname )

  print d

  sorter = gdcm.Sorter()
  sorter.SetSortFunction( MySort )
  #sorter.AddObserver( "ProgressEvent", PrintProgress )
  sorter.Sort( d.GetFilenames() )

  print "Sorter:"
  print sorter
```

14.137 WriteBuffer.py

```

"""
Usage:

http://chuckhahm.com/Ischem/Zurich/XX_0134
```

```

(2005,1132) SQ (Sequence with undefined length #=8)     # u/l, 1 Unknown Tag & Data
  (fffe,e000) na (Item with undefined length #=9)         # u/l, 1 Item
    (2005,0011) LO [Philips MR Imaging DD 002]              #  26, 1 PrivateCreator
    (2005,1137) PN [PDF_CONTROL_GEN_PARS]                   #  20, 1 Unknown Tag & Data
    (2005,1138) PN (no value available)                     #   0, 0 Unknown Tag & Data
    (2005,1139) PN [IEEE_PDF]                               #   8, 1 Unknown Tag & Data
    (2005,1140) PN (no value available)                     #   0, 0 Unknown Tag & Data
    (2005,1141) PN (no value available)                     #   0, 0 Unknown Tag & Data
    (2005,1143) SL 3103                                     #   4, 1 Unknown Tag & Data
    (2005,1144) OW 0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008... # 3104, 1 Unknown Tag & Data
    (2005,1147) CS [Y]                                      #   2, 1 Unknown Tag & Data
  (fffe,e00d) na (ItemDelimitationItem)                   #   0, 0 ItemDelimitationItem
  (fffe,e000) na (Item with undefined length #=9)         # u/l, 1 Item
    (2005,0011) LO [Philips MR Imaging DD 002]              #  26, 1 PrivateCreator
    (2005,1137) PN [PDF_CONTROL_PREP_PARS]                  #  22, 1 Unknown Tag & Data
    (2005,1138) PN (no value available)                     #   0, 0 Unknown Tag & Data
    (2005,1139) PN [IEEE_PDF]                               #   8, 1 Unknown Tag & Data
    (2005,1140) PN (no value available)                     #   0, 0 Unknown Tag & Data
    (2005,1141) PN (no value available)                     #   0, 0 Unknown Tag & Data
    (2005,1143) SL 7934                                     #   4, 1 Unknown Tag & Data
    (2005,1144) OW 19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008... # 7934, 1 Unknown Tag & Data
    (2005,1147) CS [Y]                                      #   2, 1 Unknown Tag & Data
  (fffe,e00d) na (ItemDelimitationItem)                   #   0, 0 ItemDelimitationItem
...
"""

import sys
import gdcm

if __name__ == "__main__":

  file1 = sys.argv[1]
  file2 = sys.argv[2]

  r = gdcm.Reader()
  r.SetFileName( file1 )
  if not r.Read():
    sys.exit(1)

  fg = gdcm.FilenameGenerator()
  f = r.GetFile()
  ds = f.GetDataSet()
  tsis = gdcm.Tag(0x2005,0x1132) #
  if ds.FindDataElement( tsis ):
    sis = ds.GetDataElement( tsis )
    #sqsis = sis.GetSequenceOfItems()
    # GetValueAsSQ handle more cases
    sqsis = sis.GetValueAsSQ()
    if sqsis.GetNumberOfItems():
      nitems = sqsis.GetNumberOfItems();
      fg.SetNumberOfFilenames( nitems )
      fg.SetPrefix( file2 )
      if not fg.Generate():
        print "problem"
        sys.exit(1)
      for i in range(0,nitems):
        item1 = sqsis.GetItem(i+1) # Item start at 1
        nestedds = item1.GetNestedDataSet()
        tprcs = gdcm.Tag(0x2005,0x1144) #
        if nestedds.FindDataElement( tprcs ):
          prcs = nestedds.GetDataElement( tprcs )
          bv = prcs.GetByteValue()
          print bv
          f = open( fg.GetFilename(i) , "w" )
          f.write( bv.WriteBuffer() )
```

14.138 HelloActiviz.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;
using Kitware.VTK;
using System;
using System.Runtime.InteropServices;

/*
 * This example shows how vtkgdcm can be connected to Kitware.VTK Activiz product.
 * Three (3) arguments are required:
 * 1. Input DICOM file                      (SWIG)
 * 2. Temporary PNG (intermediate) file     (Activiz)
 * 3. Final DICOM file                      (SWIG)
 *
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz.exe ~/Creatis/gdcmData/test.acr out.png toto.dcm
 *
 * Footnote:
 * this test originally used vtkBMPWriter / vtkBMPReader combination to store intermediate
 * image file, but BMP file are 24bits by default. Instead use PNG format which supports seems
 * to be closer to what was expected in this simple test.
 */
public class HelloActiviz
{
  // Does not work with ActiViz.NET-5.4.0.455-Linux-x86_64-Personal
/*
  static void ConnectSWIGToActiviz(Kitware.VTK.vtkImageExport imgin, Kitware.VTK.vtkImageImport imgout)
    {
    imgout.SetUpdateInformationCallback(imgin.GetUpdateInformationCallback());
    imgout.SetPipelineModifiedCallback(imgin.GetPipelineModifiedCallback());
    imgout.SetWholeExtentCallback(imgin.GetWholeExtentCallback());
    imgout.SetSpacingCallback(imgin.GetSpacingCallback());
    imgout.SetOriginCallback(imgin.GetOriginCallback());
    imgout.SetScalarTypeCallback(imgin.GetScalarTypeCallback());
    imgout.SetNumberOfComponentsCallback(imgin.GetNumberOfComponentsCallback());
    imgout.SetPropagateUpdateExtentCallback(imgin.GetPropagateUpdateExtentCallback());
    imgout.SetUpdateDataCallback(imgin.GetUpdateDataCallback());
    imgout.SetDataExtentCallback(imgin.GetDataExtentCallback());
    imgout.SetBufferPointerCallback(imgin.GetBufferPointerCallback());
    imgout.SetCallbackUserData(imgin.GetCallbackUserData());
    }
*/

  static Kitware.VTK.vtkImageData ConnectSWIGToActiviz(vtkgdcm.vtkImageData imgin)
    {
    HandleRef rawCppThis = imgin.GetCppThis();
    Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.Handle, false, false);
    return imgout;
    }

  static vtkgdcm.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)
    {
    HandleRef rawCppThis = imgin.GetCppThis();
    vtkgdcm.vtkImageData imgout = new vtkgdcm.vtkImageData( rawCppThis );
    return imgout;
    }


  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];

    // Step 1. Test SWIG -> Activiz
    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    reader.SetFileName( filename );
    //reader.Update(); // DO NOT call Update to check pipeline execution

    Kitware.VTK.vtkImageData imgout = ConnectSWIGToActiviz(reader.GetOutput());

    System.Console.WriteLine( imgout.ToString() ); // not initialized as expected

    vtkPNGWriter writer = new vtkPNGWriter();
    writer.SetInput( imgout );
    writer.SetFileName( outfilename );
```

```
    writer.Write();

    // Step 2. Test Activiz -> SWIG
    vtkPNGReader bmpreader = new vtkPNGReader();
    bmpreader.SetFileName( outfilename );
    //bmpreader.Update(); // DO NOT update to check pipeline execution

    System.Console.WriteLine( bmpreader.GetOutput().ToString() ); // not initialized as expected

    vtkgdcm.vtkImageData imgout2 = ConnectActivizToSWIG(bmpreader.GetOutput());

    System.Console.WriteLine( imgout2.ToString() ); // not initialized as expected


    Kitware.VTK.vtkMedicalImageProperties prop = new Kitware.VTK.vtkMedicalImageProperties();
    prop.SetModality( "MR" );

    string outfilename2 = args[2];
    vtkGDCMImageWriter writer2 = vtkGDCMImageWriter.New();
    writer2.SetMedicalImageProperties( prop.CastToActiviz() );
    writer2.SetFileName( outfilename2 );
    writer2.SetInput( imgout2 );
    writer2.Write();

    return 0;
    }
}
```

14.139 HelloActiviz2.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * Usage:
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz2.exe gdcmData/test.acr bla.png bla2.dcm
 */

/*
 * From the outside view, no-one can detect that object pass to/from
 * vtkGDCMImageWriter/vtkGDCMImageReader are not Activiz object.
 *
 * TODO: Test Command/Observer
 */
public class HelloActiviz2
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    string outfilename = args[1];
    string outfilename2 = args[2];

    vtkGDCMImageReader reader = new Kitware.VTK.GDCM.vtkGDCMImageReader();
    reader.SetFileName( filename );

    // When calling multiple times creation of C# object from the same C++ object it triggers a:
//error: potential refcounting error: Duplicate rawCppThis - weak reference that is still alive. Attempting to
      add '0x00b2dc10' again.
//       Allowing new wrapped object to take over table key...
//       Original object should *not* have been destroyed while we still had it in our table without notifying
      us...
    //reader.GetOutput();
```

```
    //reader.GetOutput();

    System.Console.WriteLine( reader.ToString() ); // Test the ToString compat with Activiz

    vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
    writer.SetInput( reader.GetOutput() );
    writer.SetFileName( outfilename2 );
    writer.Write();

    System.Console.WriteLine( reader.GetOutput().ToString() ); // Test the ToString compat with Activiz

    System.Console.WriteLine( writer.ToString() ); // Test the ToString compat with Activiz

    vtkPNGWriter pngwriter = new vtkPNGWriter();
    pngwriter.SetInput( reader.GetOutput() );
    pngwriter.SetFileName( outfilename );
    pngwriter.Write();

    // at that point the .Write() should have triggered an Update() on the reader:
    if( reader.GetImageFormat() == vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2
      {
      System.Console.WriteLine( "Image is MONOCHROME2" ); //
      }

    vtkPNGReader bmpreader = new vtkPNGReader();
    bmpreader.SetFileName( outfilename );

    vtkMedicalImageProperties prop = new vtkMedicalImageProperties();
    prop.SetModality( "MR" );

    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();

    vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter();
    writer2.SetFileName( outfilename2 );
    writer2.SetDirectionCosines( dircos );
    writer2.SetMedicalImageProperties( prop );
    writer2.SetInput( bmpreader.GetOutput() );
    writer2.Write();

    return 0;
    }
}
```

14.140 HelloActiviz3.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz3.exe ~/Creatis/gdcmData/test.acr
 */
public class HelloActiviz3
{
  public static int Main(string[] args)
    {
    string filename = args[0];

    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);

    reader.SetFileNames(array);
```

```
    reader.Update();

    //System.Console.Write(reader.GetOutput());

    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

    vtkImageViewer2 viewer = vtkImageViewer2.New();
    viewer.SetInput(reader.GetOutput());
    viewer.SetupInteractor(iren);
    viewer.SetSize(600, 600);
    viewer.Render();

    iren.Initialize();
    iren.Start();

    return 0;
    }
}
```

14.141 HelloActiviz4.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz4.exe ~/Creatis/gdcmData/test.acr
 */
public class HelloActiviz4
{
  public static int Main(string[] args)
    {
    string filename = args[0];

    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);

    reader.SetFileNames(array);
    reader.Update();

    //System.Console.Write(reader.GetOutput());

    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

    vtkImageViewer viewer = vtkImageViewer.New();
    viewer.SetInput(reader.GetOutput());
    viewer.SetupInteractor(iren);
    viewer.SetSize(600, 600);
    viewer.Render();

    iren.Initialize();
    iren.Start();

    return 0;
    }
}
```

14.142 HelloActiviz5.cs

```
/*=========================================================================
```

```
  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

// The command line arguments are:
// -I        => run in interactive mode; unless this is used, the program will
//              not allow interaction and exit
// -D <path> => path to the data; the data should be in <path>/Data/

/*
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz5.exe -I
 */
public class HelloActiviz5
{
  public static int Main(string[] args)
    {
    vtkTesting testHelper = vtkTesting.New();
    for ( int cc = 0; cc < args.Length; cc++ )
      {
      //testHelper.AddArguments(argc,const_cast<const char **>(argv));
      //System.Console.Write( "args: " + args[cc] + "\n" );
      testHelper.AddArgument( args[cc] );
      }
    if ( testHelper.IsFlagSpecified("-D") != 0 )
      {
      string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();
      if( VTK_DATA_ROOT != null )
        {
        //System.Console.Write( "VTK_DATA_ROOT: " + VTK_DATA_ROOT  + "\n" );
        testHelper.SetDataRoot(VTK_DATA_ROOT);
        testHelper.AddArgument("-D");
        testHelper.AddArgument(VTK_DATA_ROOT);
        }
      }

    string dataRoot = testHelper.GetDataRoot();
    string filename = dataRoot;
    filename += "/Data/mr.001";

    vtkDirectory dir = vtkDirectory.New();
    if( dir.FileIsDirectory( dataRoot ) == 0 )
      {
      filename = vtkGDCMTesting.GetGDCMDataRoot() + "/test.acr";
      }
    //System.Console.Write( "dataRoot: " + dataRoot + "\n" );
    System.Console.Write( "filename being used is: " + filename + "\n" );

    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    vtkStringArray array = vtkStringArray.New();
    array.InsertNextValue(filename);
    reader.SetFileNames(array);
    reader.Update();

    System.Console.Write(reader.GetOutput());

    vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

    vtkRenderer ren1 = vtkRenderer.New();
    vtkRenderWindow renWin = vtkRenderWindow.New();
    renWin.AddRenderer(ren1);

    vtkImageActor actor = vtkImageActor.New();

    vtkImageMapToWindowLevelColors coronalColors = vtkImageMapToWindowLevelColors.New();
    coronalColors.SetInput(reader.GetOutput());

    actor.SetInput(coronalColors.GetOutput());

    ren1.AddActor(actor);
```

```
    iren.SetRenderWindow(renWin);

    iren.Initialize();

    renWin.Render();

    int retVal = testHelper.IsInteractiveModeSpecified();

    if( retVal != 0 )
      {
      iren.Start();
      }

    return 0;
    }
}
```

14.143 HelloVTKWorld.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld
{
  public static int Main(string[] args)
    {
    string filename = args[0];
    vtkGDCMImageReader reader = vtkGDCMImageReader.New();
    reader.SetFileName( filename );
    reader.Update();

    vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
    System.Console.WriteLine( prop.GetPatientName() ); //

    if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2
      {
      System.Console.WriteLine( "Image is MONOCHROME2" ); //
      }

    // Just for fun, invert the direction cosines, output should reflect that:
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();

    string outfilename = args[1];
    vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
    writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    writer.SetInputConnection( reader.GetOutputPort() );
    writer.Write();

    return 0;
    }
}
```

14.144 HelloVTKWorld2.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using vtkgdcm;

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloVTKWorld2
{
  public static int Main(string[] args)
    {
    string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();

    vtkVolume16Reader reader = vtkVolume16Reader.New();
    reader.SetDataDimensions(64, 64);
    reader.SetDataByteOrderToLittleEndian();
    reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter");
    reader.SetImageRange(1, 93);
    reader.SetDataSpacing(3.2, 3.2, 1.5);

    vtkImageCast cast = vtkImageCast.New();
    cast.SetInputConnection( reader.GetOutputPort() );
    cast.SetOutputScalarTypeToUnsignedChar();

    // By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
    vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
    writer.SetFileName( "headsq.dcm" );
    writer.SetInputConnection( reader.GetOutputPort() );
    // cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
    // writer.SetInputConnection( cast.GetOutputPort() );
    writer.SetFileDimensionality( 3 );
    writer.Write();

    return 0;
    }
}
```

14.145 MetaImageMD5Activiz.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;
using gdcm;

/*
 * $ export MONO_PATH=/usr/lib/cli/ActiViz.NET/:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/MetaImageMD5Activiz.exe gdcmData/012345.002.050.dcm
 */
public class MetaImageMD5Activiz
{
  public static int ProcessOneMHDMD5(string filename)
```

```
  {
  vtkGDCMImageReader reader = vtkGDCMImageReader.New();
  reader.FileLowerLeftOn();
  reader.DebugOff();
  int canread = reader.CanReadFile( filename );
  if( canread == 0 )
    {
    string refms = gdcm.Testing.GetMediaStorageFromFile(filename);
    if( gdcm.MediaStorage.IsImage( gdcm.MediaStorage.GetMSType(refms) ) )
      {
      System.Console.Write( "Problem with file: " + filename + "\n" );
      return 1;
      }
    // not an image
    return 0;
    }

  reader.SetFileName( filename );
  reader.Update();

  // System.Console.Write(reader.GetOutput());

  vtkMetaImageWriter writer = vtkMetaImageWriter.New();
  writer.SetCompression( false );
  writer.SetInput( reader.GetOutput() );
  string subdir = "MetaImageMD5Activiz";
  string tmpdir = gdcm.Testing.GetTempDirectory( subdir );
  if( !gdcm.PosixEmulation.FileIsDirectory( tmpdir ) )
    {
    gdcm.PosixEmulation.MakeDirectory( tmpdir );
    }
  string mhdfile = gdcm.Testing.GetTempFilename( filename, subdir );

  string rawfile = mhdfile;
  mhdfile += ".mhd";
  rawfile += ".raw";
  writer.SetFileName( mhdfile );
  writer.Write();

  string digestmhd = gdcm.Testing.ComputeFileMD5( mhdfile );
  string digestraw = gdcm.Testing.ComputeFileMD5( rawfile );

  string mhdref = vtkGDCMTesting.GetMHDMD5FromFile(filename);
  string rawref = vtkGDCMTesting.GetRAWMD5FromFile(filename);

  if( mhdref != digestmhd )
    {
    System.Console.Write( "Problem with mhd file: " + filename + "\n" );
    System.Console.Write( digestmhd );
    System.Console.Write( "\n" );
    System.Console.Write( mhdref );
    System.Console.Write( "\n" );
    return 1;
    }
  if( rawref != digestraw )
    {
    System.Console.Write( "Problem with raw file: " + filename + "\n" );
    System.Console.Write( digestraw );
    System.Console.Write( "\n" );
    System.Console.Write( rawref );
    System.Console.Write( "\n" );
    return 1;
    }

  return 0;
  }
public static int Main(string[] args)
  {
  if ( args.Length  == 1 )
    {
    string filename = args[0];
    return ProcessOneMHDMD5( filename );
    }
  // Loop over all gdcmData
  gdcm.Trace.DebugOff();
  gdcm.Trace.WarningOff();
  gdcm.Trace.ErrorOff();

  uint n = gdcm.Testing.GetNumberOfFileNames();
  int ret = 0;
  for( uint i = 0; i < n; ++i )
```

```
      {
      string filename = gdcm.Testing.GetFileName( i );
      ret += ProcessOneMHDMD5( filename );
      }
    return ret;
    }
}
```

14.146 RefCounting.cs

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * this is not so much an example but simply a test to make sure cstor / dstor work as expected
 * and call the ::New and ->Delete() of VTK style.
 */
public class RefCounting
{
  public static int Main(string[] args)
    {
    vtkGDCMTesting testing1 = vtkGDCMTesting.New();
    vtkGDCMTesting testing2 = new vtkGDCMTesting(); // just in case people do not read STYLE documentation

    vtkGDCMImageReader reader1 = vtkGDCMImageReader.New();
    vtkGDCMImageReader reader2 = new vtkGDCMImageReader();

    vtkGDCMImageWriter writer1 = vtkGDCMImageWriter.New();
    vtkGDCMImageWriter writer2 = new vtkGDCMImageWriter();

    using (vtkGDCMTesting testing3 = new vtkGDCMTesting())
      {
      System.Console.Write( "GetReferenceCount: " + testing1.GetReferenceCount() + "\n");
      System.Console.Write( "GetReferenceCount: " + testing2.GetReferenceCount() + "\n");
      System.Console.Write( "GetReferenceCount: " + testing3.GetReferenceCount() + "\n");
      }

    using (vtkGDCMImageReader reader3 = new vtkGDCMImageReader())
      {
      System.Console.Write( "GetReferenceCount: " + reader3.GetReferenceCount() + "\n");
      }

    using (vtkGDCMImageWriter writer3 = vtkGDCMImageWriter.New())
      {
      System.Console.Write( "GetReferenceCount: " + writer3.GetReferenceCount() + "\n");
      }

    // C# destructor will call ->Delete on all C++ object as expected.
    return 0;
    }
}
```

14.147 Compute3DSpacing.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
```

```
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader2.h"
#include "vtkImageChangeInformation.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"
#include "gdcmIPPSorter.h"

#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
#endif

/*
 * Simple example to check computation of spacing within vtkGDCMImageReader2
 * This is a direct implementation of:
 *
 *
      http://gdcm.sourceforge.net/wiki/index.php/Using_GDCM_API#Automatic_ordering_of_slices_for_vtkGDCMImageReader.SetFileNames
 *
 * For more advanced information on how 3D spacing is being computed see:
 *
 *  - http://gdcm.sourceforge.net/html/classgdcm_1_1IPPSorter.html
 *
 * Usage:
 *
 * $ Compute3DSpacing SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm \
 *     SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm \
 *     SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm \
 *     SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm
 */

int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;

  std::vector<std::string> filenames;
  for( int  i = 1; i < argc; ++i )
    {
    filenames.push_back( argv[i] );
    }

  gdcm::IPPSorter s;
  s.SetComputeZSpacing( true );
  s.SetZSpacingTolerance( 1e-3 );
  bool b = s.Sort( filenames );
  if( !b )
    {
    std::cerr « "Failed to sort files" « std::endl;
    return 1;
    }
  std::cout « "Sorting succeeded:" « std::endl;
  //s.Print( std::cout );

  std::cout « "Found z-spacing:" « std::endl;
  std::cout « s.GetZSpacing() « std::endl;
  const double ippzspacing = s.GetZSpacing();

  const std::vector<std::string> & sorted = s.GetFilenames();
  vtkGDCMImageReader2 * reader = vtkGDCMImageReader2::New();
  vtkStringArray *files = vtkStringArray::New();
  std::vector< std::string >::const_iterator it = sorted.begin();
  for( ; it != sorted.end(); ++it)
    {
    const std::string &f = *it;
    files->InsertNextValue( f.c_str() );
    }
  reader->SetFileNames( files );
  reader->Update();

  const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();
  vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
  v16->SetInputConnection( reader->GetOutputPort() );
#else
  v16->SetInput( reader->GetOutput() );
```

```
#endif
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();

  v16->GetOutput()->Print( std::cout );

  return 0;
}
```

14.148 Convert16BitsTo8Bits.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"
// The following file is 16/16/15 but the scalar range of the image is [0,192]
// it could be safely stored as 8bits instead:
// gdcmData/012345.002.050.dcm

int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot();
  if(!directory) return 1;
  std::string file = std::string(directory) + "/012345.002.050.dcm";
  std::cout « file « std::endl;

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  vtkImageCast *cast = vtkImageCast::New();
#if (VTK_MAJOR_VERSION >= 6)
  cast->SetInputConnection( reader->GetOutputPort() );
#else
  cast->SetInput( reader->GetOutput() );
#endif
  cast->SetOutputScalarTypeToUnsignedChar();


  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileName( "/tmp/cast.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( cast->GetOutputPort() );
#else
  writer->SetInput( cast->GetOutput() );
#endif
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines() );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();

  reader->Delete();
  cast->Delete();
  writer->Delete();

  return 0;
}
```

14.149 ConvertMultiFrameToSingleFrame.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"
#include "gdcmFilenameGenerator.h"

int main(int argc, char *argv[])
{
  std::string filename;
  if( argc <= 1 )
    {
    const char *directory = gdcm::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/US-PAL-8-10x-echo.dcm";
    filename = file;
    }
  else
    {
    filename = argv[1];
    }
  std::cout « "file: " « filename « std::endl;

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( filename.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  int dims[3];
  reader->GetOutput()->GetDimensions( dims );

  std::ostringstream os;
  os « "singleframe";
  os « "%04d.dcm";
  gdcm::FilenameGenerator fg;
  fg.SetPattern( os.str().c_str() );
  unsigned int nfiles = dims[2];
  fg.SetNumberOfFilenames( nfiles );
  bool b = fg.Generate();
  if( !b )
    {
    std::cerr « "FilenameGenerator::Generate() failed" « std::endl;
    return 1;
    }
  if( !fg.GetNumberOfFilenames() )
    {
    std::cerr « "FilenameGenerator::Generate() failed somehow..." « std::endl;
    return 1;
    }

  // By default write them as Secondary Capture (for portability)
  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  vtkStringArray *filenames = vtkStringArray::New();
  for(unsigned int i = 0; i < fg.GetNumberOfFilenames(); ++i)
    {
    filenames->InsertNextValue( fg.GetFilename(i) );
    }
  assert( filenames->GetNumberOfValues() == (int)fg.GetNumberOfFilenames() );
  writer->SetFileNames( filenames );
  filenames->Delete();
  writer->SetFileDimensionality( 2 );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( reader->GetOutputPort() );
```

```
#else
  writer->SetInput( reader->GetOutput() );
#endif
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->Write();

  reader->Delete();
  writer->Delete();

  return 0;
}
```

14.150 ConvertRGBToLuminance.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageLuminance.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"

// There is no such thing as MR Image Storage + Photometric Interpretation = RGB
// let's rewrite that into a proper single component image:
int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot();
  if(!directory) return 1;
  std::string file = std::string(directory) + "/SIEMENS-MR-RGB-16Bits.dcm";
  std::cout « file « std::endl;

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  vtkImageLuminance *luminance = vtkImageLuminance::New();
#if (VTK_MAJOR_VERSION >= 6)
  luminance->SetInputConnection( reader->GetOutputPort() );
#else
  luminance->SetInput( reader->GetOutput() );
#endif


  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileName( "/tmp/bla.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( luminance->GetOutputPort() );
#else
  writer->SetInput( luminance->GetOutput() );
#endif
  //writer->SetImageFormat( reader->GetImageFormat() ); // Do NOT pass image format
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines() );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();

  // TODO:
  //vtkImageAppendComponents.h

  reader->Delete();
  luminance->Delete();
```

```
  writer->Delete();

  return 0;
}
```

14.151 ConvertSingleBitTo8Bits.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageCast.h"
#include "vtkPointData.h"
#include "vtkBitArray.h"
#include "vtkUnsignedCharArray.h"
#include "vtkVersion.h"

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    return 1;
    }
  const char *filename = argv[1];
  const char *outfilename = argv[2];

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( filename );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  vtkDataArray* array = reader->GetOutput()->GetPointData()->GetScalars();
  vtkBitArray *barray = vtkBitArray::SafeDownCast( array );
  if( !barray ) return false;
  vtkIdType nvalues = array->GetNumberOfTuples();
  vtkUnsignedCharArray *uarray = vtkUnsignedCharArray::New();
  uarray->SetNumberOfTuples( nvalues );
  for(vtkIdType i = 0; i < nvalues; ++i)
    {
    uarray->SetValue( i, (unsigned char)barray->GetValue(i) );
    }

  vtkImageData *copy = vtkImageData::New();
  // http://www.vtk.org/Wiki/VTK/VTK_6_Migration/Changes_to_Scalars_Manipulation_Functions#AllocateScalars.28.29
  copy->SetExtent( reader->GetOutput()->GetExtent() );
#if (VTK_MAJOR_VERSION >= 6)
  copy->AllocateScalars(VTK_UNSIGNED_CHAR, 3);
#else
  copy->SetScalarType( VTK_UNSIGNED_CHAR );
  copy->AllocateScalars();
#endif

  //uarray->Print( std::cout );
  //copy->GetPointData()->GetScalars()->Print( std::cout );
  copy->GetPointData()->SetScalars( uarray );
  uarray->Delete();

  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileName( outfilename );
  //writer->SetInput( cast->GetOutput() );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputData( copy );
#else
  writer->SetInput( copy );
```

```
#endif
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines() );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->SetFileDimensionality( reader->GetFileDimensionality( ) );
  writer->Write();

  reader->Delete();
  copy->Delete();
  writer->Delete();

  return 0;
}
```

14.152 CreateFakePET.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcmTrace.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"
#include "gdcmFilenameGenerator.h"

/*
 * Minimal example to create a fake RTDOSE file. The data contains a sphere
 * just for testing.
 * The vtkMedicalImageProperties is not properly filled, but only contains a
 * single field which is required to set the proper SOP Class
 */
int main(int, char *[])
{
  gdcm::Trace::DebugOn();

  const vtkIdType xSize = 512;
  const vtkIdType ySize = 512;
  const vtkIdType zSize = 512;

  // Create the filenames in advance to supply to the vtkGDCMImageWriter
  std::ostringstream os;
  os « "PT";
  os « "%03d.dcm";
  gdcm::FilenameGenerator fg;
  fg.SetPattern( os.str().c_str() );
  unsigned int nfiles = zSize;
  fg.SetNumberOfFilenames( nfiles );
  bool b = fg.Generate();
  if( !b )
    {
    std::cerr « "FilenameGenerator::Generate() failed" « std::endl;
    return 1;
    }
  if( !fg.GetNumberOfFilenames() )
```

```
    {
    std::cerr « "FilenameGenerator::Generate() failed somehow..." « std::endl;
    return 1;
    }

  vtkStringArray *filenames = vtkStringArray::New();
  for(unsigned int i = 0; i < fg.GetNumberOfFilenames(); ++i)
    {
    filenames->InsertNextValue( fg.GetFilename(i) );
    }

  vtkImageData *image = vtkImageData::New();
  image->SetDimensions(xSize,ySize,zSize);
  image->SetOrigin(-350.684,350.0,890.76);
  image->SetSpacing(5.4688,-5.4688,-3.27);
#if VTK_MAJOR_VERSION <= 5
  image->SetNumberOfScalarComponents(1);
  image->SetScalarTypeToDouble();
#else
  image->AllocateScalars(VTK_DOUBLE ,1);
#endif

  double pt[3];
  for( int z = 0; z < zSize; ++z )
    for( int y = 0; y < ySize; ++y )
      for( int x = 0; x < xSize; ++x )
        {
        pt[0] = x;
        pt[1] = y;
        pt[2] = z;
        pt[0] -= xSize / 2;
        pt[1] -= ySize / 2;
        pt[2] -= zSize / 2;
        pt[0] /= xSize / 2;
        pt[1] /= ySize / 2;
        pt[2] /= zSize / 2;
        const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
        const double inval = unit <= 1. ? (3 * unit + 7) : 0.; // just for fun => max == 10.
        double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
        pixel[0] = inval;
        }


  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileDimensionality( 2 );
  writer->SetFileNames(filenames);
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputData( image );
#else
  writer->SetInput( image );
#endif
  writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
  writer->GetMedicalImageProperties()->SetModality( "PT" );
  writer->SetScale( 0.0042 ); // why not
  writer->Write();

  image->Delete();
  writer->Delete();

  return 0;
}
```

14.153 CreateFakeRTDOSE.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.
```

```
=========================================================================*/
#include "vtkGDCMImageWriter.h"
#include "vtkImageReader.h"
#include "vtkImageCast.h"
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkDataArray.h"
#include "vtkMedicalImageProperties.h"
#include "vtkVersion.h"

#include "gdcmTrace.h"
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"

/*
 * Minimal example to create a fake RTDOSE file. The data contains a sphere
 * just for testing.
 * The vtkMedicalImageProperties is not properly filled, but only contains a
 * single field which is required to set the proper SOP Class
 */
int main(int, char *[])
{
  //gdcm::Trace::DebugOn();

  const vtkIdType xSize = 512;
  const vtkIdType ySize = 512;
  const vtkIdType zSize = 512;

  vtkImageData *image = vtkImageData::New();
  image->SetDimensions(xSize,ySize,zSize);
  image->SetOrigin(-350.684,350.0,890.76);
  image->SetSpacing(5.4688,-5.4688,-3.27);
#if VTK_MAJOR_VERSION <= 5
  image->SetNumberOfScalarComponents(1);
  image->SetScalarTypeToDouble();
#else
  image->AllocateScalars(VTK_DOUBLE ,1);
#endif

  double pt[3];
  for( int z = 0; z < zSize; ++z )
    for( int y = 0; y < ySize; ++y )
      for( int x = 0; x < xSize; ++x )
        {
        pt[0] = x;
        pt[1] = y;
        pt[2] = z;
        pt[0] -= xSize / 2;
        pt[1] -= ySize / 2;
        pt[2] -= zSize / 2;
        pt[0] /= xSize / 2;
        pt[1] /= ySize / 2;
        pt[2] /= zSize / 2;
        const double unit = pt[0] * pt[0] + pt[1] * pt[1] + pt[2] * pt[2];
        const double inval = unit <= 1. ? (3 * unit + 7) : 0.; // just for fun => max == 10.
        double* pixel= static_cast<double*>(image->GetScalarPointer(x,y,z));
        pixel[0] = inval;
        }


  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileDimensionality( 3 );
  writer->SetFileName( "rtdose.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputData( image );
#else
  writer->SetInput( image );
#endif
  writer->GetMedicalImageProperties()->SetSliceThickness("1.5");
  writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Units", "GY");
  writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Summation Type", "PLAN");
  writer->GetMedicalImageProperties()->AddUserDefinedValue( "Dose Type", "PHYSICAL");
  writer->GetMedicalImageProperties()->AddUserDefinedValue( "Frame of Reference UID",
      "1.3.12.2.1107.5.6.1.68100.30270111041215391275000000001");
  writer->GetMedicalImageProperties()->SetModality( "RTDOSE" );
  //writer->GetMedicalImageProperties()->SetModality( "PT" ); // debug
  writer->SetScale( 0.0042 ); // why not
  writer->Write();

  image->Delete();
```

```
  writer->Delete();


  // BEGIN HACK
  // In GDCM version 2.4.3 and before, the following tag was missing which caused issue with some RTDose
      software:

  // Open the DICOM file that was temporarily created. This will allows me to used
  // GDCM to append specific tags that allows the RTDOSE to be associated with the
  // relevant CT images.
  gdcm::Reader reader2;
  reader2.SetFileName("rtdose.dcm" );
  reader2.Read();
  gdcm::File &file = reader2.GetFile();
  gdcm::DataSet &ds = file.GetDataSet();

  // Required by some software and not automagically added by GDCM in old version
  gdcm::Attribute<0x0028,0x0009> framePointer;
  framePointer.SetNumberOfValues(1);
  framePointer.SetValue( gdcm::Tag(0x3004,0x000C) );
  ds.Replace( framePointer.GetAsDataElement() );

  gdcm::Writer writer2;
  writer2.CheckFileMetaInformationOff();
  writer2.SetFileName("rtdose2.dcm");
  writer2.SetFile( file );
  writer2.Write();
  // END HACK

  return 0;
}
```

14.154 GenerateRTSTRUCT.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataWriter.h"
#include "vtkGDCMPolyDataReader.h"
#include "vtkPolyData.h"
#include "vtkPolyDataReader.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRTStructSetProperties.h"
#include "vtkStringArray.h"
#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkImageData.h"
#include "vtkVersion.h"

#include <algorithm> //for std::find

#include "gdcmDirectoryHelper.h"

using namespace gdcm;

//view each organ independently of the others, to make sure that
```

```
//organ names correspond to actual segmentations.
void ShowOrgan(vtkPolyData* inData)
{
  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
  cubeMapper->SetInputData( inData );
#else
  cubeMapper->SetInput( inData );
#endif
  cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();

  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  renWin->AddRenderer(renderer);

  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);

  renderer->AddActor(cubeActor);
  renderer->ResetCamera();
  renderer->SetBackground(1,1,1);

  renWin->SetSize(300,300);

  renWin->Render();
  iren->Start();

  cubeMapper->Delete();
  cubeActor->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();
}

/*
 * Full application which ... RTSTUCT
 */
int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " directory-with-rtstruct-and-ct-images\n";
    return 1;
    }
  std::string theDirName(argv[1]);
  Directory::FilenamesType theRTSeries =
    DirectoryHelper::GetRTStructSeriesUIDs(theDirName);

  gdcm::Directory theDir;
  theDir.Load(argv[1]);

  if (theRTSeries.empty())
    {
    std::cerr « "No RTStructs found for the test, ending." « std::endl;
    return 1;
    }

  for (size_t q = 0; q < theRTSeries.size(); q++)
    {
    Directory::FilenamesType theRTNames =
      DirectoryHelper::GetFilenamesFromSeriesUIDs(theDirName, theRTSeries[q]);

    if (theRTNames.empty()){
      std::cerr « "Unable to load RT Series " « theRTSeries[q] « ", continuing. " « std::endl;
      continue;
    }

    vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
    reader->SetFileName( theRTNames[0].c_str() );
    reader->Update();

    //std::cout « reader->GetMedicalImageProperties()->GetStudyDate() « std::endl;

    vtkGDCMPolyDataWriter * writer = vtkGDCMPolyDataWriter::New();
    int numMasks = reader->GetNumberOfOutputPorts() + 1;//add a blank one in
    writer->SetNumberOfInputPorts( numMasks );
```

```
    std::string thePotentialName = theDirName + "/" + "GDCMTestRTStruct." +  theRTSeries[q] + ".dcm";
    gdcm::Directory::FilenamesType theFileNames = theDir.GetFilenames();
    //keep renaming the output until we get something that doesn't overwrite what was there already
    int count = 0;
    while (std::find(theFileNames.begin(), theFileNames.end(), thePotentialName) != theFileNames.end())
      {
        char buff[255];
        snprintf(buff,sizeof(buff),"%d",count);
        thePotentialName = theDirName + "/" + "GDCMTestRTStruct." + buff + "." + theRTSeries[q] + ".dcm";
      }
    writer->SetFileName( thePotentialName.c_str());
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    //this line is cheating, we won't have the same stuff, and may not have a struct
    //to start with.
    //have to go back to the original data to reconstruct the RTStructureSetProperties
    //writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
    //writer->Write();

    //loop through the outputs in order to write them out as if they had been created and appended
    vtkStringArray* roiNames = vtkStringArray::New();
    vtkStringArray* roiAlgorithms = vtkStringArray::New();
    vtkStringArray* roiTypes = vtkStringArray::New();
    roiNames->SetNumberOfValues(numMasks);
    roiAlgorithms->SetNumberOfValues(numMasks);
    roiTypes->SetNumberOfValues(numMasks);
    vtkAppendPolyData* append = vtkAppendPolyData::New();

    //ok, now we'll add a blank organ
    //the blank organ is to test to ensure that blank organs work; there have been crash reports
    //this code is added at the beginning to ensure that the blank organs are read
    //and preserved as individual organs.
    vtkPolyData* blank = vtkPolyData::New();
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputData(0, blank);
#else
    writer->SetInput(0, blank);
#endif
    roiNames->InsertValue(0, "blank");
    roiAlgorithms->InsertValue(0, "blank");
    roiTypes->InsertValue(0, "ORGAN");

    //note the offsets used to place the blank rtstruct at the beginning of the newly generated RT.
    //the idea is to run the program twice; first to generate an rtstruct with a blank mask (making
    //sure that that functionality works), and then a second time to make sure that everything is
    //being read properly.  Multiple organs with the same name could cause some strangenesses.
    for (int i = 1; i < numMasks; ++i)
      {
#if (VTK_MAJOR_VERSION >= 6)
      writer->SetInputConnection(i, reader->GetOutputPort(i-1));
      append->AddInputConnection(reader->GetOutputPort(i-1));
#else
      writer->SetInput(i, reader->GetOutput(i-1));
      append->AddInput(reader->GetOutput(i-1));
#endif
      std::string theString = reader->GetRTStructSetProperties()->GetStructureSetROIName(i-1);
      roiNames->InsertValue(i, theString);
      theString = reader->GetRTStructSetProperties()->GetStructureSetROIGenerationAlgorithm(i-1);
      roiAlgorithms->InsertValue(i, theString);
      theString = reader->GetRTStructSetProperties()->GetStructureSetRTROIInterpretedType(i-1);
      roiTypes->InsertValue(i, theString);

      ShowOrgan(reader->GetOutput(i-1));
      }

    vtkRTStructSetProperties* theProperties = vtkRTStructSetProperties::New();
    writer->SetRTStructSetProperties(theProperties);
    writer->InitializeRTStructSet(theDirName,
      reader->GetRTStructSetProperties()->GetStructureSetLabel(),
      reader->GetRTStructSetProperties()->GetStructureSetName(),
      roiNames, roiAlgorithms, roiTypes);

    writer->SetRTStructSetProperties(theProperties);
    writer->Write();

    // print reader output:
    reader->Print( std::cout );
    // print first output:
    reader->GetOutput()->Print( std::cout );

    reader->Delete();
    append->Delete();
```

```
    roiNames->Delete();
    roiTypes->Delete();
    theProperties->Delete();
    roiAlgorithms->Delete();
    blank->Delete();

    writer->Delete();
  }
  return 0;
}
```

14.155 MagnifyFile.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageMagnify.h"
#include "vtkImageCast.h"
#include "vtkVersion.h"

#include "gdcmTesting.h"
#include "gdcmSystem.h"

// This is a simple test to magnify an image that is known to give excellent
// compression ratio. This will be our test for those large image
int main(int, char *[])
{
  const char *directory = gdcm::Testing::GetDataRoot();
  if(!directory) return 1;
  std::string file = std::string(directory) + "/test.acr";
  std::cout « file « std::endl;
  if( !gdcm::System::FileExists( file.c_str() ) ) return 1;

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( file.c_str() );
  reader->Update();
  //reader->GetOutput()->Print( std::cout );

  vtkImageCast *cast = vtkImageCast::New();
#if (VTK_MAJOR_VERSION >= 6)
  cast->SetInputConnection( reader->GetOutputPort() );
#else
  cast->SetInput( reader->GetOutput() );
#endif
  cast->SetOutputScalarTypeToUnsignedShort();

  vtkImageMagnify *magnify = vtkImageMagnify::New();
#if (VTK_MAJOR_VERSION >= 6)
  magnify->SetInputConnection( cast->GetOutputPort() );
#else
  magnify->SetInput( cast->GetOutput() );
#endif
  magnify->SetInterpolate( 1 );
  magnify->SetInterpolate( 0 );
  int factor = 100;
  magnify->SetMagnificationFactors (factor, factor, 1);

  vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
  writer->SetFileName( "/tmp/bla.dcm" );
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( magnify->GetOutputPort() );
#else
  writer->SetInput( magnify->GetOutput() );
```

```
#endif
  writer->SetImageFormat( reader->GetImageFormat() );
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetDirectionCosines( reader->GetDirectionCosines() );
  writer->SetShift( reader->GetShift() );
  writer->SetScale( reader->GetScale() );
  writer->Write();

  // TODO:
  //vtkImageAppendComponents.h

  reader->Delete();
  magnify->Delete();
  writer->Delete();

  return 0;
}
```

14.156 gdcmorthoplanes.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/

#include "vtkActor.h"
#include "vtkCamera.h"
#include "vtkMatrix4x4.h"
#include "vtkTransform.h"
#include "vtkAssembly.h"
#include "vtkCellPicker.h"
#include "vtkCommand.h"
#include "vtkImageActor.h"
#include "vtkImageMapToColors.h"
#include "vtkImageOrthoPlanes.h"
#include "vtkImagePlaneWidget.h"
#include "vtkImageReader.h"
#include "vtkInteractorEventRecorder.h"
#include "vtkLookupTable.h"
#include "vtkOutlineFilter.h"
#include "vtkPolyDataMapper.h"
#include "vtkProperty.h"
#include "vtkRenderWindow.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderer.h"
#include "vtkVolume16Reader.h"
#include "vtkImageData.h"
#include "vtkImageChangeInformation.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkAxesActor.h"
#include "vtkCaptionActor2D.h"
#include "vtkTextProperty.h"
#include "vtkPropAssembly.h"

#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkStringArray.h"
#include "vtkVersion.h"

#include "gdcmSystem.h"
#include "gdcmDirectory.h"
#include "gdcmIPPSorter.h"

#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
```

```
#endif

//-----------------------------------------------------------------------------
class vtkOrthoPlanesCallback : public vtkCommand
{
public:
  static vtkOrthoPlanesCallback *New()
  { return new vtkOrthoPlanesCallback; }

  void Execute( vtkObject *caller, unsigned long vtkNotUsed( event ),
                void *callData )
  {
    vtkImagePlaneWidget* self =
      reinterpret_cast< vtkImagePlaneWidget* >( caller );
    if(!self) return;

    double* wl = static_cast<double*>( callData );

    if ( self == this->WidgetX )
      {
      this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
      }
    else if( self == this->WidgetY )
      {
      this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetZ->SetWindowLevel(wl[0],wl[1],1);
      }
    else if (self == this->WidgetZ)
      {
      this->WidgetX->SetWindowLevel(wl[0],wl[1],1);
      this->WidgetY->SetWindowLevel(wl[0],wl[1],1);
      }
  }

  vtkOrthoPlanesCallback():WidgetX( 0 ), WidgetY( 0 ), WidgetZ ( 0 ) {}

  vtkImagePlaneWidget* WidgetX;
  vtkImagePlaneWidget* WidgetY;
  vtkImagePlaneWidget* WidgetZ;
};

int main( int argc, char *argv[] )
{
  //char* fname = vtkTestUtilities::ExpandDataFileName(argc, argv, "Data/headsq/quarter");

  //vtkVolume16Reader* v16 =  vtkVolume16Reader::New();
  //  v16->SetDataDimensions( 64, 64);
  //  v16->SetDataByteOrderToLittleEndian();
  //  v16->SetImageRange( 1, 93);
  //  v16->SetDataSpacing( 3.2, 3.2, 1.5);
  //  v16->SetFilePrefix( fname );
  //  v16->SetDataMask( 0x7fff);
  //  v16->Update();
  std::vector<std::string> filenames;
  if( argc < 2 )
    {
    std::cerr « argv[0] « " filename1.dcm [filename2.dcm ...]\n";
    return 1;
    }
  else
    {
    // Is it a single directory ? If so loop over all files contained in it:
    const char *filename = argv[1];
    if( argc == 2 && gdcm::System::FileIsDirectory( filename ) )
      {
      std::cout « "Loading directory: " « filename « std::endl;
      bool recursive = false;
      gdcm::Directory d;
      d.Load(filename, recursive);
      gdcm::Directory::FilenamesType const &files = d.GetFilenames();
      for( gdcm::Directory::FilenamesType::const_iterator it = files.begin(); it != files.end(); ++it )
        {
        filenames.push_back( it->c_str() );
        }
      }
    else // list of files passed directly on the cmd line:
        // discard non-existing or directory
      {
      for(int i=1; i < argc; ++i)
        {
```

```
        filename = argv[i];
        if( gdcm::System::FileExists( filename ) )
          {
          if( gdcm::System::FileIsDirectory( filename ) )
            {
            std::cerr « "Discarding directory: " « filename « std::endl;
            }
          else
            {
            filenames.push_back( filename );
            }
          }
        else
          {
          std::cerr « "Discarding non existing file: " « filename « std::endl;
          }
        }
      }
    //names->Print( std::cout );
    }

  vtkGDCMImageReader * reader = vtkGDCMImageReader::New();
  double ippzspacing;
  if( filenames.size() > 1 )
    {
    //gdcm::Trace::DebugOn();
    //gdcm::Trace::WarningOn();
    gdcm::IPPSorter s;
    s.SetComputeZSpacing( true );
    s.SetZSpacingTolerance( 1e-3 );
    bool b = s.Sort( filenames );
    if( !b )
      {
      std::cerr « "Failed to sort files" « std::endl;
      return 1;
      }
    std::cout « "Sorting succeeded:" « std::endl;
    s.Print( std::cout );

    std::cout « "Found z-spacing:" « std::endl;
    std::cout « s.GetZSpacing() « std::endl;
    ippzspacing = s.GetZSpacing();

    const std::vector<std::string> & sorted = s.GetFilenames();
    vtkStringArray *files = vtkStringArray::New();
    std::vector< std::string >::const_iterator it = sorted.begin();
    for( ; it != sorted.end(); ++it)
      {
      const std::string &f = *it;
      files->InsertNextValue( f.c_str() );
      }
    reader->SetFileNames( files );
    //reader->SetFileLowerLeft( 1 );
    reader->Update(); // important
    files->Delete();
    }
  else
    {
    reader->SetFileName( argv[1] );
    reader->Update(); // important
    ippzspacing = reader->GetOutput()->GetSpacing()[2];
    ippzspacing = 4;
    }

  //reader->GetOutput()->Print( std::cout );
  //vtkFloatingPointType range[2];
  //reader->GetOutput()->GetScalarRange(range);
  //std::cout « "Range: " « range[0] « " " « range[1] « std::endl;

  const vtkFloatingPointType *spacing = reader->GetOutput()->GetSpacing();

  vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
  v16->SetInputConnection( reader->GetOutputPort() );
#else
  v16->SetInput( reader->GetOutput() );
#endif
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();

#if 0
```

```
    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetInput( v16->GetOutput() );
    writer->SetFileLowerLeft( reader->GetFileLowerLeft() );
    writer->SetDirectionCosines( reader->GetDirectionCosines() );
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->SetFileDimensionality( 3); //reader->GetFileDimensionality() );
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->SetFileName( "out.dcm" );
    writer->Write();
#endif


  vtkOutlineFilter* outline = vtkOutlineFilter::New();
    outline->SetInputConnection(v16->GetOutputPort());

  vtkPolyDataMapper* outlineMapper = vtkPolyDataMapper::New();
    outlineMapper->SetInputConnection(outline->GetOutputPort());

  vtkActor* outlineActor =  vtkActor::New();
    outlineActor->SetMapper( outlineMapper);

  vtkRenderer* ren1 = vtkRenderer::New();
  vtkRenderer* ren2 = vtkRenderer::New();

  vtkRenderWindow* renWin = vtkRenderWindow::New();
    renWin->AddRenderer(ren2);
    renWin->AddRenderer(ren1);

  vtkRenderWindowInteractor* iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

  vtkCellPicker* picker = vtkCellPicker::New();
    picker->SetTolerance(0.005);

  vtkProperty* ipwProp = vtkProperty::New();
   //assign default props to the ipw's texture plane actor

  vtkImagePlaneWidget* planeWidgetX = vtkImagePlaneWidget::New();
    planeWidgetX->SetInteractor( iren);
    planeWidgetX->SetKeyPressActivationValue('x');
    planeWidgetX->SetPicker(picker);
    planeWidgetX->RestrictPlaneToVolumeOn();
    planeWidgetX->GetPlaneProperty()->SetColor(1,0,0);
    planeWidgetX->SetTexturePlaneProperty(ipwProp);
    planeWidgetX->TextureInterpolateOff();
    planeWidgetX->SetResliceInterpolateToNearestNeighbour();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetX->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetX->SetInput(v16->GetOutput());
#endif
    planeWidgetX->SetPlaneOrientationToXAxes();
    //planeWidgetX->SetSliceIndex(32);
    planeWidgetX->DisplayTextOn();
    planeWidgetX->On();
    planeWidgetX->InteractionOff();
    planeWidgetX->InteractionOn();

  vtkImagePlaneWidget* planeWidgetY = vtkImagePlaneWidget::New();
    planeWidgetY->SetInteractor( iren);
    planeWidgetY->SetKeyPressActivationValue('y');
    planeWidgetY->SetPicker(picker);
    planeWidgetY->GetPlaneProperty()->SetColor(1,1,0);
    planeWidgetY->SetTexturePlaneProperty(ipwProp);
    planeWidgetY->TextureInterpolateOn();
    planeWidgetY->SetResliceInterpolateToLinear();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetY->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetY->SetInput(v16->GetOutput());
#endif
    planeWidgetY->SetPlaneOrientationToYAxes();
    //planeWidgetY->SetSlicePosition(102.4);
    planeWidgetY->SetLookupTable( planeWidgetX->GetLookupTable());
    planeWidgetY->DisplayTextOn();
    planeWidgetY->UpdatePlacement();
    planeWidgetY->On();

  vtkImagePlaneWidget* planeWidgetZ = vtkImagePlaneWidget::New();
```

```
    planeWidgetZ->SetInteractor( iren);
    planeWidgetZ->SetKeyPressActivationValue('z');
    planeWidgetZ->SetPicker(picker);
    planeWidgetZ->GetPlaneProperty()->SetColor(0,0,1);
    planeWidgetZ->SetTexturePlaneProperty(ipwProp);
    planeWidgetZ->TextureInterpolateOn();
    planeWidgetZ->SetResliceInterpolateToCubic();
#if (VTK_MAJOR_VERSION >= 6)
    planeWidgetZ->SetInputConnection(v16->GetOutputPort());
#else
    planeWidgetZ->SetInput(v16->GetOutput());
#endif
    planeWidgetZ->SetPlaneOrientationToZAxes();
    //planeWidgetZ->SetSliceIndex(25);
    planeWidgetZ->SetLookupTable( planeWidgetX->GetLookupTable());
    planeWidgetZ->DisplayTextOn();
    planeWidgetZ->On();

  vtkImageOrthoPlanes *orthoPlanes = vtkImageOrthoPlanes::New();
    orthoPlanes->SetPlane(0, planeWidgetX);
    orthoPlanes->SetPlane(1, planeWidgetY);
    orthoPlanes->SetPlane(2, planeWidgetZ);
    orthoPlanes->ResetPlanes();

   vtkOrthoPlanesCallback* cbk = vtkOrthoPlanesCallback::New();
   cbk->WidgetX = planeWidgetX;
   cbk->WidgetY = planeWidgetY;
   cbk->WidgetZ = planeWidgetZ;
   planeWidgetX->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
   planeWidgetY->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
   planeWidgetZ->AddObserver( vtkCommand::EndWindowLevelEvent, cbk );
   cbk->Delete();

  double wl[2];
  planeWidgetZ->GetWindowLevel(wl);

  // Add a 2D image to test the GetReslice method
  //
  vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
    colorMap->PassAlphaToOutputOff();
    colorMap->SetActiveComponent(0);
    colorMap->SetOutputFormatToLuminance();
#if (VTK_MAJOR_VERSION >= 6)
    colorMap->SetInputData(planeWidgetZ->GetResliceOutput());
#else
    colorMap->SetInput(planeWidgetZ->GetResliceOutput());
#endif
    colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

  vtkImageActor* imageActor = vtkImageActor::New();
    imageActor->PickableOff();
#if (VTK_MAJOR_VERSION >= 6)
    imageActor->SetInputData(colorMap->GetOutput());
#else
    imageActor->SetInput(colorMap->GetOutput());
#endif

  // Add the actors
  //
  ren1->AddActor( outlineActor);
  ren2->AddActor( imageActor);

  ren1->SetBackground( 0.1, 0.1, 0.2);
  ren2->SetBackground( 0.2, 0.1, 0.2);

  renWin->SetSize( 600, 350);

  ren1->SetViewport(0,0,0.58333,1);
  ren2->SetViewport(0.58333,0,1,1);

  // Set the actors' positions
  //
  renWin->Render();
  //iren->SetEventPosition( 175,175);
  //iren->SetKeyCode('r');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);
  //iren->SetEventPosition( 475,175);
  //iren->SetKeyCode('r');
  //iren->InvokeEvent(vtkCommand::CharEvent,NULL);
```

```
//renWin->Render();

//ren1->GetActiveCamera()->Elevation(110);
//ren1->GetActiveCamera()->SetViewUp(0, 0, -1);
//ren1->GetActiveCamera()->Azimuth(45);
//ren1->GetActiveCamera()->Dolly(1.15);
ren1->ResetCameraClippingRange();

vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText ( "R" );
cube->SetXMinusFaceText( "L" );
cube->SetYPlusFaceText ( "A" );
cube->SetYMinusFaceText( "P" );
cube->SetZPlusFaceText ( "H" );
cube->SetZMinusFaceText( "F" );
cube->SetFaceTextScale( 0.666667 );

vtkAxesActor* axes2 = vtkAxesActor::New();

vtkMatrix4x4 *invert = vtkMatrix4x4::New();
invert->DeepCopy( reader->GetDirectionCosines() );
invert->Invert();

// simulate a left-handed coordinate system
//
vtkTransform *transform = vtkTransform::New();
transform->Identity();
//transform->RotateY(90);
transform->Concatenate(invert);
axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
cube->GetAssembly()->SetUserTransform( transform );

axes2->SetTotalLength( 1.5, 1.5, 1.5 );
axes2->SetCylinderRadius( 0.500 * axes2->GetCylinderRadius() );
axes2->SetConeRadius    ( 1.025 * axes2->GetConeRadius() );
axes2->SetSphereRadius  ( 1.500 * axes2->GetSphereRadius() );

vtkTextProperty* tprop = axes2->GetXAxisCaptionActor2D()->
  GetCaptionTextProperty();
tprop->ItalicOn();
tprop->ShadowOn();
tprop->SetFontFamilyToTimes();

axes2->GetYAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );
axes2->GetZAxisCaptionActor2D()->GetCaptionTextProperty()->ShallowCopy( tprop );

vtkPropAssembly* assembly = vtkPropAssembly::New();
assembly->AddPart( axes2 );
assembly->AddPart( cube );

vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
widget->SetOrientationMarker( assembly );
widget->SetInteractor( iren );
widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
widget->SetEnabled( 1 );
widget->InteractiveOff();
widget->InteractiveOn();

// Playback recorded events
//
//vtkInteractorEventRecorder *recorder = vtkInteractorEventRecorder::New();
//recorder->SetInteractor(iren);
//recorder->ReadFromInputStringOn();
//recorder->SetInputString(IOPeventLog);

// Interact with data
// Render the image
//
iren->Initialize();
renWin->Render();

// Test SetKeyPressActivationValue for one of the widgets
//
//iren->SetKeyCode('z');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);
//iren->SetKeyCode('z');
//iren->InvokeEvent(vtkCommand::CharEvent,NULL);

//int retVal = vtkRegressionTestImage( renWin );
```

```
  //
  //if ( retVal == vtkRegressionTester::DO_INTERACTOR)
    {
    iren->Start();
    }

  // Clean up
  //
  //recorder->Off();
  //recorder->Delete();

  ipwProp->Delete();
  orthoPlanes->Delete();
  planeWidgetX->Delete();
  planeWidgetY->Delete();
  planeWidgetZ->Delete();
  colorMap->Delete();
  imageActor->Delete();
  picker->Delete();
  outlineActor->Delete();
  outlineMapper->Delete();
  outline->Delete();
  iren->Delete();
  renWin->Delete();
  ren1->Delete();
  ren2->Delete();
  v16->Delete();
  reader->Delete();

  return 0;
}
```

14.157 gdcmreslice.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkImageFlip.h"
#include "vtkImageReslice.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"
#include "vtkVersion.h"

int main( int argc, char *argv[] )
{
  if( argc < 2 ) return 1;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( argv[1] );
  //reader->FileLowerLeftOn();
```

```
  reader->Update();

  vtkImageFlip *flip = vtkImageFlip::New();
#if (VTK_MAJOR_VERSION >= 6)
  flip->SetInputConnection(reader->GetOutputPort());
#else
  flip->SetInput(reader->GetOutput());
#endif
  flip->SetFilteredAxis(0);
  flip->Update();

  vtkImageReslice *reslice = vtkImageReslice::New();
  //reslice->SetInput(reader->GetOutput());
#if (VTK_MAJOR_VERSION >= 6)
  reslice->SetInputConnection(flip->GetOutputPort());
#else
  reslice->SetInput(flip->GetOutput());
#endif
  //reslice->SetResliceAxesDirectionCosines()
  reader->GetDirectionCosines()->Print(std::cout);
  vtkMatrix4x4 *invert = vtkMatrix4x4::New();
  invert->DeepCopy( reader->GetDirectionCosines() );
  invert->Invert();

  //reslice->SetResliceAxes( reader->GetDirectionCosines() );
  reslice->SetResliceAxes( invert );
  reslice->Update();
  vtkImageData* ima = reslice->GetOutput();

  vtkLookupTable* table = vtkLookupTable::New();
  table->SetNumberOfColors(1000);
  table->SetTableRange(0,1000);
  table->SetSaturationRange(0,0);
  table->SetHueRange(0,1);
  table->SetValueRange(0,1);
  table->SetAlphaRange(1,1);
  table->Build();

  // Texture
  vtkTexture* texture = vtkTexture::New();
#if (VTK_MAJOR_VERSION >= 6)
  texture->SetInputData(ima);
#else
  texture->SetInput(ima);
#endif
  texture->InterpolateOn();
  texture->SetLookupTable(table);

  // PlaneSource
  vtkPlaneSource* plane = vtkPlaneSource::New();

  // PolyDataMapper
  vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
  planeMapper->SetInputConnection(plane->GetOutputPort());
#else
  planeMapper->SetInput(plane->GetOutput());
#endif

  // Actor
  vtkActor* planeActor = vtkActor::New();
  planeActor->SetTexture(texture);
  planeActor->SetMapper(planeMapper);
  planeActor->PickableOn();

  // Final rendering with simple interactor:
  vtkRenderer        *ren = vtkRenderer::New();
  vtkRenderWindow *renwin = vtkRenderWindow::New();
  renwin->AddRenderer(ren);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renwin);
  ren->AddActor(planeActor);
  ren->SetBackground(0,0,0.5);

  // DICOM is RAH:
  vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
  cube->SetXPlusFaceText ( "R" );
  cube->SetXMinusFaceText( "L" );
  cube->SetYPlusFaceText ( "A" );
  cube->SetYMinusFaceText( "P" );
  cube->SetZPlusFaceText ( "H" );
```

```
  cube->SetZMinusFaceText( "F" );

  vtkAxesActor* axes2 = vtkAxesActor::New();

  vtkTransform *transform = vtkTransform::New();
  transform->Identity();
  //reader->GetDirectionCosines()->Print(std::cout);
  transform->Concatenate(invert);
  //axes2->SetShaftTypeToCylinder();
  axes2->SetUserTransform( transform );
  cube->GetAssembly()->SetUserTransform( transform ); // can't get it to work

  vtkPropAssembly* assembly = vtkPropAssembly::New();
  assembly->AddPart( axes2 );
  assembly->AddPart( cube );

  vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
  widget->SetOrientationMarker( assembly );
  widget->SetInteractor( iren );
  widget->SetEnabled( 1 );
  widget->InteractiveOff();
  widget->InteractiveOn();

  renwin->Render();
  iren->Start();

  // Clean up:
  reader->Delete();
  table->Delete();
  texture->Delete();
  plane->Delete();
  planeMapper->Delete();
  planeActor->Delete();
  ren->Delete();
  renwin->Delete();
  iren->Delete();

  return 0;
}
```

14.158 gdcmrtionplan.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkXMLPolyDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>
#include "vtkVersion.h"

#include "gdcmReader.h"
#include "gdcmAttribute.h"

/*
 This example is just for fun. We found a RT Ion Plan Storage and simply extracted the viz stuff for VTK
```

```
    RTIonPlanStorage, // 1.2.840.10008.5.1.4.1.1.481.8
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " filename.dcm outfile.vti\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];
  const char * outfilename2 = argv[3];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::MediaStorage ms;
  ms.SetFromFile( reader.GetFile() );
  if( ms != gdcm::MediaStorage::RTIonPlanStorage )
    {
    return 1;
    }

/*
(300a,03a2) SQ                                                  # u/l,1 Ion Beam Sequence
  (fffe,e000) na (Item with undefined length)
    (0008,1040) LO [Test]                                       # 4,1 Institutional Department Name
    (300a,00b2) SH (no value)                                   # 0,1 Treatment Machine Name
    (300a,00b3) CS [MU]                                         # 2,1 Primary Dosimeter Unit
    (300a,00c0) IS [1 ]                                         # 2,1 Beam Number
    (300a,00c2) LO [1 ]                                         # 2,1 Beam Name
    (300a,00c4) CS [STATIC]                                     # 6,1 Beam Type
    (300a,00c6) CS [PROTON]                                     # 6,1 Radiation Type
    (300a,00ce) CS [TREATMENT ]                                 # 10,1 Treatment Delivery Type
    (300a,00d0) IS [0 ]                                         # 2,1 Number of Wedges
    (300a,00e0) IS [1 ]                                         # 2,1 Number of Compensators
    (300a,00ed) IS [0 ]                                         # 2,1 Number of Boli
    (300a,00f0) IS [1 ]                                         # 2,1 Number of Blocks
    (300a,0110) IS [2 ]                                         # 2,1 Number of Control Points
    (300a,02ea) SQ                                              # u/l,1 Ion Range Compensator Sequence
      (fffe,e000) na (Item with undefined length)
        (300a,00e1) SH [lucite]                                 # 6,1 Material ID
        (300a,00e4) IS [1 ]                                     # 2,1 Compensator Number
        (300a,00e5) SH [75hdhe5 ]                               # 8,1 Compensator ID
        (300a,00e7) IS [35]                                     # 2,1 Compensator Rows
        (300a,00e8) IS [37]                                     # 2,1 Compensator Columns
        (300a,00e9) DS [3.679991\4.249288 ]                     # 18,2 Compensator Pixel Spacing
        (300a,00ea) DS [-76.00\62.50]                           # 12,2 Compensator Position
        (300a,00ec) DS
      [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.
      # 7618,1-n Compensator Thickness Data
        (300a,02e0) CS [ABSENT]                                 # 6,1 Compensator Divergence
        (300a,02e1) CS [SOURCE_SIDE ]                           # 12,1 Compensator Mounting Position
        (300a,02e4) FL 39.2                                     # 4,1 Isocenter to Compensator Tray Distance
        (300a,02e5) FL 2.12                                     # 4,1 Compensator Column Offset
        (300a,02e8) FL 4.76                                     # 4,1 Compensator Milling Tool Diameter
      (fffe,e00d)
*/
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  gdcm::Tag tbeamsq(0x300a,0x03a2);
  if( !ds.FindDataElement( tbeamsq ) )
    {
    return 1;
    }
  const gdcm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
  //std::cout « beamsq « std::endl;
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = beamsq.GetValueAsSQ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return 1;
    }

  //for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
  //  {
    //const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
    const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
```

```
    const gdcm::DataSet& nestedds = item.GetNestedDataSet();
    //std::cout « nestedds « std::endl;
    gdcm::Tag tcompensatorsq(0x300a,0x02ea);
    if( !nestedds.FindDataElement( tcompensatorsq ) )
      {
      return 1;
      }
    const gdcm::DataElement &compensatorsq = nestedds.GetDataElement( tcompensatorsq );
    //std::cout « compensatorsq « std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> ssqi = compensatorsq.GetValueAsSQ();
    const gdcm::Item & item2 = ssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
    //std::cout « nestedds2 « std::endl;
    gdcm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
    if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &compensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
    //  std::cout « compensatorthicknessdata « std::endl;
    gdcm::Attribute<0x300a,0x00ec> at;
    at.SetFromDataElement( compensatorthicknessdata );
    const double* pts = at.GetValues();
    //        (300a,00e7) IS [35]                                     # 2,1 Compensator Rows
    gdcm::Attribute<0x300a,0x00e7> at1;
    const gdcm::DataElement &compensatorrows = nestedds2.GetDataElement( at1.GetTag() );
    at1.SetFromDataElement( compensatorrows );
    std::cout « at1.GetValue() « std::endl;
    //        (300a,00e8) IS [37]                                     # 2,1 Compensator Columns
    gdcm::Attribute<0x300a,0x00e8> at2;
    const gdcm::DataElement &compensatorcols = nestedds2.GetDataElement( at2.GetTag() );
    at2.SetFromDataElement( compensatorcols );
    std::cout « at2.GetValue() « std::endl;

        // (300a,00e9) DS [3.679991\4.249288 ]                       # 18,2 Compensator Pixel Spacing
    gdcm::Attribute<0x300a,0x00e9> at3;
    const gdcm::DataElement &compensatorpixelspacing = nestedds2.GetDataElement( at3.GetTag() );
    at3.SetFromDataElement( compensatorpixelspacing );
    std::cout « at3.GetValue(0) « std::endl;
        // (300a,00ea) DS [-76.00\62.50]                             # 12,2 Compensator Position
    gdcm::Attribute<0x300a,0x00ea> at4;
    const gdcm::DataElement &compensatorposition = nestedds2.GetDataElement( at4.GetTag() );
    at4.SetFromDataElement( compensatorposition );
    std::cout « at4.GetValue(0) « std::endl;

    vtkDoubleArray *d = vtkDoubleArray::New();
    d->SetArray( const_cast<double*>(pts) , at1.GetValue() * at2.GetValue() , 0 );

    vtkImageData *img = vtkImageData::New();
    img->Initialize();
    img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
    //imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetScalarTypeToDouble();
#endif
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetNumberOfScalarComponents(1);
#endif
    img->GetPointData()->SetScalars(d);

#if (VTK_MAJOR_VERSION >= 6)
#else
    img->Update();
#endif
    img->Print(std::cout);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
```

```
    (300a,03a6) SQ                                          # u/l,1 Ion Block Sequence
      (fffe,e000) na (Item with undefined length)
        (300a,00e1) SH [brass ]                             # 6,1 Material ID
        (300a,00f7) FL 95.03                                # 4,1 Isocenter to Block Tray Distance
        (300a,00f8) CS [APERTURE]                           # 8,1 Block Type
        (300a,00fa) CS [ABSENT]                             # 6,1 Block Divergence
        (300a,00fb) CS [SOURCE_SIDE ]                       # 12,1 Block Mounting Position
        (300a,00fc) IS [1 ]                                 # 2,1 Block Number
        (300a,0100) DS [50.00 ]                             # 6,1 Block Thickness
        (300a,0104) IS [179 ]                               # 4,1 Block Number of Points
        (300a,0106) DS
      [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2
2\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
      (fffe,e00d)
    (fffe,e0dd)

*/
    gdcm::Tag tblocksq(0x300a,0x03a6);
    if( !nestedds.FindDataElement( tblocksq ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
    //std::cout « blocksq « std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> sssqi = blocksq.GetValueAsSQ();
    const gdcm::Item & item3 = sssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds3 = item3.GetNestedDataSet();

    gdcm::Tag tblockdata(0x300a,0x0106);
    if( !nestedds3.FindDataElement( tblockdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &blockdata = nestedds3.GetDataElement( tblockdata );
    //  std::cout « blockdata « std::endl;
    gdcm::Attribute<0x300a,0x0106> at_;
    at_.SetFromDataElement( blockdata );

    vtkDoubleArray *scalars = vtkDoubleArray::New();
    scalars->SetNumberOfComponents(3);

    gdcm::Attribute<0x300a,0x0104> bnpts; //  IS [179 ]                                    # 4,1 Block Number
      of Points
    if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocknpts = nestedds3.GetDataElement( bnpts.GetTag() );
    bnpts.SetFromDataElement(  blocknpts );
    //std::cout « bnpts.GetValue() « std::endl;

    vtkPolyData *output = vtkPolyData::New();
    vtkPoints *newPts = vtkPoints::New();
    vtkCellArray *polys = vtkCellArray::New();
    const double *ptr = at_.GetValues();
    //unsigned int npts = bnpts.GetNumberOfValues() / 2;
    unsigned int npts = bnpts.GetValue();
    vtkIdType *ptIds =  new vtkIdType[npts];
    for(unsigned int i = 0; i < npts; ++i)
      {
      float x[3] = {};
      x[0] = (float)ptr[2*i+0];
      x[1] = (float)ptr[2*i+1];
      //x[2] = pts[i+2];
      vtkIdType ptId = newPts->InsertNextPoint( x );
      //std::cout « x[0] « "," « x[1] « "," « x[2] « std::endl;
      ptIds[i ] = ptId;
      }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
#if (VTK_MAJOR_VERSION >= 6)
#else
    output->Update();
```

```
#endif
    output->Print( std::cout );



  //  }

   vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

   vtkImageColorViewer *viewer = vtkImageColorViewer::New();
#if (VTK_MAJOR_VERSION >= 6)
   viewer->SetInputData(img);
#else
   viewer->SetInput(img);
#endif
   viewer->SetupInteractor(iren);
   viewer->SetSize(600, 600);
   viewer->GetRenderer()->ResetCameraClippingRange();
   viewer->Render();
   viewer->GetRenderer()->ResetCameraClippingRange();

  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
#if (VTK_MAJOR_VERSION >= 6)
      cubeMapper->SetInputData( output );
#else
      cubeMapper->SetInput( output );
#endif
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();

viewer->GetRenderer()->AddActor( cubeActor );

    vtkXMLPolyDataWriter *writec= vtkXMLPolyDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writec->SetInputData( output );
#else
    writec->SetInput( output );
#endif
    writec->SetFileName( outfilename2 );
    writec->Write( );

   iren->Initialize();
   iren->Start();


  return 0;
}
```

14.159 gdcmrtplan.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
```

```
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>
#include "vtkVersion.h"

#include "gdcmReader.h"
#include "gdcmAttribute.h"

/*
 This example is just for fun. We found a fake RT Ion Plan Storage and simply extracted the viz stuff for VTK
 but this is rather a RT Plan storage
*/
int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " filename.dcm outfile.vti\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];

  gdcm::Reader reader;
  reader.SetFileName( filename );
  if( !reader.Read() )
    {
    return 1;
    }

  gdcm::MediaStorage ms;
  ms.SetFromFile( reader.GetFile() );
  if( ms != gdcm::MediaStorage::RTIonPlanStorage )
    {
    return 1;
    }

/*
(300a,00b0) SQ                                                  # u/l,1 Beam Sequence
  (fffe,e000) na (Item with undefined length)
    (300a,00b2) SH (no value)                                   # 0,1 Treatment Machine Name
    (300a,00c0) IS [1 ]                                         # 2,1 Beam Number
    (300a,00c2) LO [1 ]                                         # 2,1 Beam Name
    (300a,00c4) CS [STATIC]                                     # 6,1 Beam Type
    (300a,00c6) CS [PROTON]                                     # 6,1 Radiation Type
    (300a,00ce) CS [TREATMENT ]                                 # 10,1 Treatment Delivery Type
    (300a,00e0) IS [1 ]                                         # 2,1 Number of Compensators
    (300a,00e3) SQ                                              # u/l,1 Compensator Sequence
      (fffe,e000) na (Item with undefined length)
        (300a,00e1) SH [lucite]                                 # 6,1 Material ID
        (300a,00e4) IS [1 ]                                     # 2,1 Compensator Number
        (300a,00e5) SH [75hdhe5 ]                               # 8,1 Compensator ID
        (300a,00e7) IS [35]                                     # 2,1 Compensator Rows
        (300a,00e8) IS [37]                                     # 2,1 Compensator Columns
        (300a,00e9) DS [3.679991\4.249288 ]                     # 18,2 Compensator Pixel Spacing
        (300a,00ea) DS [-76.00\62.50]                           # 12,2 Compensator Position
        (300a,00ec) DS
      [52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\33.28\35.43\35.43\34.54\34.54\34.71\36.
      # 7618,1-n Compensator Thickness Data
        (300a,02e0) CS [ABSENT]                                 # 6,1 Compensator Divergence
        (300a,02e1) CS [SOURCE_SIDE ]                           # 12,1 Compensator Mounting Position
      (fffe,e00d)
      (fffe,e000) na (Item with undefined length)
      (fffe,e00d)
    (fffe,e0dd)
*/
  const gdcm::DataSet& ds = reader.GetFile().GetDataSet();
  gdcm::Tag tbeamsq(0x300a,0x00b0);
  if( !ds.FindDataElement( tbeamsq ) )
    {
    return 1;
    }
  const gdcm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
  //std::cout « beamsq « std::endl;
  gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = beamsq.GetValueAsSQ();
  if( !sqi || !sqi->GetNumberOfItems() )
    {
    return 1;
```

```
    }

  //for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
  //  {
    //const gdcm::Item & item = sqi->GetItem(1); // Item start at #1
    const gdcm::Item & item = sqi->GetItem(2); // Item start at #1
    const gdcm::DataSet& nestedds = item.GetNestedDataSet();
    //std::cout « nestedds « std::endl;
    gdcm::Tag tcompensatorsq(0x300a,0x00e3);
    if( !nestedds.FindDataElement( tcompensatorsq ) )
      {
      return 1;
      }
    const gdcm::DataElement &compensatorsq = nestedds.GetDataElement( tcompensatorsq );
    //std::cout « compensatorsq « std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> ssqi = compensatorsq.GetValueAsSQ();
    const gdcm::Item & item2 = ssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds2 = item2.GetNestedDataSet();
    //std::cout « nestedds2 « std::endl;
    gdcm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
    if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &compensatorthicknessdata = nestedds2.GetDataElement( tcompensatorthicknessdata );
    //  std::cout « compensatorthicknessdata « std::endl;
    gdcm::Attribute<0x300a,0x00ec> at;
    at.SetFromDataElement( compensatorthicknessdata );
    const double* pts = at.GetValues();
    //        (300a,00e7) IS [35]                                     # 2,1 Compensator Rows
    gdcm::Attribute<0x300a,0x00e7> at1;
    const gdcm::DataElement &compensatorrows = nestedds2.GetDataElement( at1.GetTag() );
    at1.SetFromDataElement( compensatorrows );
    std::cout « at1.GetValue() « std::endl;
    //        (300a,00e8) IS [37]                                     # 2,1 Compensator Columns
    gdcm::Attribute<0x300a,0x00e8> at2;
    const gdcm::DataElement &compensatorcols = nestedds2.GetDataElement( at2.GetTag() );
    at2.SetFromDataElement( compensatorcols );
    std::cout « at2.GetValue() « std::endl;

        // (300a,00e9) DS [3.679991\4.249288 ]                       # 18,2 Compensator Pixel Spacing
    gdcm::Attribute<0x300a,0x00e9> at3;
    const gdcm::DataElement &compensatorpixelspacing = nestedds2.GetDataElement( at3.GetTag() );
    at3.SetFromDataElement( compensatorpixelspacing );
    std::cout « at3.GetValue(0) « std::endl;
        // (300a,00ea) DS [-76.00\62.50]                             # 12,2 Compensator Position
    gdcm::Attribute<0x300a,0x00ea> at4;
    const gdcm::DataElement &compensatorposition = nestedds2.GetDataElement( at4.GetTag() );
    at4.SetFromDataElement( compensatorposition );
    std::cout « at4.GetValue(0) « std::endl;

    vtkDoubleArray *d = vtkDoubleArray::New();
    d->SetArray( const_cast<double*>(pts) , at1.GetValue() * at2.GetValue() , 0 );

    vtkImageData *img = vtkImageData::New();
    img->Initialize();
    img->SetDimensions( at2.GetValue(), at1.GetValue(), 1 );
    //imgb->SetExtent(1, xdim, 1, ydim, 1, zdim);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetScalarTypeToDouble();
#endif
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
#if (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    img->SetNumberOfScalarComponents(1);
#endif
    img->GetPointData()->SetScalars(d);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
    writeb->SetInputData( img );
#else
    writeb->SetInput( img );
#endif
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
```

```
 (300a,00f4) SQ                                          # u/l,1 Block Sequence
      (fffe,e000) na (Item with undefined length)
        (300a,00e1) SH [brass ]                             # 6,1 Material ID
        (300a,00f8) CS [APERTURE]                           # 8,1 Block Type
        (300a,00fa) CS [ABSENT]                             # 6,1 Block Divergence
        (300a,00fb) CS [SOURCE_SIDE ]                       # 12,1 Block Mounting Position
        (300a,00fc) IS [1 ]                                 # 2,1 Block Number
        (300a,0100) DS [50.00 ]                             # 6,1 Block Thickness
        (300a,0104) IS [179 ]                               # 4,1 Block Number of Points
        (300a,0106) DS
      [1.7\50.0\14.3\50.0\16.7\49.4\18.7\48.2\19.4\47.7\20.1\47.1\21.0\47.0\22.3\47.0\23.7\46.8\25.7\46.2\27.0\45.6\27.2\45.4\28.2
      # 1934,2-2n Block Data
      (fffe,e00d)
      (fffe,e000) na (Item with undefined length)
      (fffe,e00d)
    (fffe,e0dd)
*/
    gdcm::Tag tblocksq(0x300a,0x00f4);
    if( !nestedds.FindDataElement( tblocksq ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
    //std::cout « blocksq « std::endl;
    gdcm::SmartPointer<gdcm::SequenceOfItems> sssqi = blocksq.GetValueAsSQ();
    const gdcm::Item & item3 = sssqi->GetItem(1); // Item start at #1
    const gdcm::DataSet& nestedds3 = item3.GetNestedDataSet();

    gdcm::Tag tblockdata(0x300a,0x0106);
    if( !nestedds3.FindDataElement( tblockdata ) )
      {
      return 1;
      }
    const gdcm::DataElement &blockdata = nestedds3.GetDataElement( tblockdata );
    //  std::cout « blockdata « std::endl;
    gdcm::Attribute<0x300a,0x0106> at_;
    at_.SetFromDataElement( blockdata );

    vtkDoubleArray *scalars = vtkDoubleArray::New();
    scalars->SetNumberOfComponents(3);

    gdcm::Attribute<0x300a,0x0104> bnpts; //  IS [179 ] # 4,1 Block Number of Points
    if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
      {
      return 1;
      }
    const gdcm::DataElement &blocknpts = nestedds3.GetDataElement( bnpts.GetTag() );
    bnpts.SetFromDataElement(  blocknpts );
    std::cout « bnpts.GetValue() « std::endl;

    vtkPolyData *output = vtkPolyData::New();
    vtkPoints *newPts = vtkPoints::New();
    vtkCellArray *polys = vtkCellArray::New();
    const double *ptr = at_.GetValues();
    //unsigned int npts = bnpts.GetNumberOfValues() / 2;
    unsigned int npts = bnpts.GetValue();
    vtkIdType *ptIds =  new vtkIdType[npts];
    for(unsigned int i = 0; i < npts; ++i)
      {
      float x[3] = {};
      x[0] = (float)ptr[2*i+0];
      x[1] = (float)ptr[2*i+1];
      //x[2] = pts[i+2];
      vtkIdType ptId = newPts->InsertNextPoint( x );
      //std::cout « x[0] « "," « x[1] « "," « x[2] « std::endl;
      ptIds[i ] = ptId;
      }
    vtkIdType cellId = polys->InsertNextCell(npts , ptIds);
    (void)cellId;
    delete[] ptIds;

    output->SetPoints(newPts);
    newPts->Delete();
    output->SetPolys(polys);
    polys->Delete();
    //output->GetCellData()->SetScalars(scalars);
    //scalars->Delete();
#if (VTK_MAJOR_VERSION >= 6)
#else
    output->Update();
#endif
```

```
    output->Print( std::cout );



  //  }

   vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

   vtkImageColorViewer *viewer = vtkImageColorViewer::New();
#if (VTK_MAJOR_VERSION >= 6)
   viewer->SetInputData(img);
#else
   viewer->SetInput(img);
#endif
   viewer->SetupInteractor(iren);
   viewer->SetSize(600, 600);
   viewer->Render();

  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
#if (VTK_MAJOR_VERSION >= 6)
      cubeMapper->SetInputData( output );
#else
      cubeMapper->SetInput( output );
#endif
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();

  viewer->GetRenderer()->AddActor( cubeActor );

   iren->Initialize();
   iren->Start();


  return 0;
}
```

14.160 gdcmscene.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataReader.h"
//#include "vtkGDCMPolyDataWriter.h"

#include "vtkAppendPolyData.h"
#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkVersion.h"


// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
```

```
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " filename1.dcm\n";
    return 1;
    }
  const char * filename = argv[1];

  vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
  reader->SetFileName( filename );
  reader->Update();

//  vtkGDCMPolyDataWriter * writer2 = vtkGDCMPolyDataWriter::New();
//  for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
//    writer2->SetInput( num, reader->GetOutput(num) );
//  writer2->SetFileName( "rtstruct.dcm" );
//  writer2->Write();

  // print reader output:
  reader->Print( std::cout );
  // print first output:
  reader->GetOutput()->Print( std::cout );

  vtkAppendPolyData *append = vtkAppendPolyData::New();
  int n = reader->GetNumberOfOutputPorts();
  for(int i = 0; i < n; ++i)
    {
#if (VTK_MAJOR_VERSION >= 6)
    append->AddInputConnection( reader->GetOutputPort(i) );
#else
    append->AddInput( reader->GetOutput(i) );
#endif
    }

  vtkPolyDataWriter * writer = vtkPolyDataWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputConnection( reader->GetOutputPort() );
#else
  writer->SetInput( reader->GetOutput() );
#endif
  writer->SetFileName( "rtstruct.vtk" );
  //writer->Write();

  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
  //vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
      //cubeMapper->SetInput( reader->GetOutput() );
#if (VTK_MAJOR_VERSION >= 6)
      cubeMapper->SetInputConnection( append->GetOutputPort() );
#else
      cubeMapper->SetInput( append->GetOutput() );
#endif
      cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  //vtkActor2D* cubeActor = vtkActor2D::New();
      cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();
  //cubeActor->GetProperty()->SetColor(1, 0, 0);


  // The usual rendering stuff.
//  vtkCamera *camera = vtkCamera::New();
//      camera->SetPosition(1,1,1);
//      camera->SetFocalPoint(0,0,0);

  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
    renWin->AddRenderer(renderer);

  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
    iren->SetRenderWindow(renWin);

  renderer->AddActor(cubeActor);
  //renderer->AddActor2D(cubeActor);
      //renderer->SetActiveCamera(camera);
      renderer->ResetCamera();
```

```
      renderer->SetBackground(1,1,1);

  renWin->SetSize(300,300);

  // interact with data
  renWin->Render();
  iren->Start();



  reader->Delete();
  append->Delete();
  cubeMapper->Delete();
  cubeActor->Delete();
//  camera->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();

  writer->Delete();

  return 0;
}
```

14.161 gdcmtexture.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"
#include "vtkVersion.h"

int main( int argc, char *argv[] )
{
  if( argc < 2 ) return 1;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( argv[1] );

  reader->Update();
  vtkImageData* ima = reader->GetOutput();

  vtkLookupTable* table = vtkLookupTable::New();
  table->SetNumberOfColors(1000);
  table->SetTableRange(0,1000);
  table->SetSaturationRange(0,0);
  table->SetHueRange(0,1);
  table->SetValueRange(0,1);
  table->SetAlphaRange(1,1);
```

```
  table->Build();

  // Texture
  vtkTexture* texture = vtkTexture::New();
#if (VTK_MAJOR_VERSION >= 6)
  texture->SetInputData(ima);
#else
  texture->SetInput(ima);
#endif
  texture->InterpolateOn();
  texture->SetLookupTable(table);

  // PlaneSource
  vtkPlaneSource* plane = vtkPlaneSource::New();
  plane->SetOrigin( -0.5, -0.5, 0.0);
  plane->SetPoint1(  0.5, -0.5, 0.0);
  plane->SetPoint2( -0.5,  0.5, 0.0);

  // PolyDataMapper
  vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
  planeMapper->SetInputConnection(plane->GetOutputPort());
#else
  planeMapper->SetInput(plane->GetOutput());
#endif

  // Actor
  vtkActor* planeActor = vtkActor::New();
  planeActor->SetTexture(texture);
  planeActor->SetMapper(planeMapper);
  planeActor->PickableOn();

  // Final rendering with simple interactor:
  vtkRenderer        *ren = vtkRenderer::New();
  vtkRenderWindow *renwin = vtkRenderWindow::New();
  renwin->AddRenderer(ren);
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renwin);
  ren->AddActor(planeActor);
  ren->SetBackground(0,0,0.5);

  vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
  cube->SetXPlusFaceText ( "L" );
  cube->SetXMinusFaceText( "R" );
  cube->SetYPlusFaceText ( "A" );
  cube->SetYMinusFaceText( "P" );
  cube->SetZPlusFaceText ( "H" );
  cube->SetZMinusFaceText( "F" );

  vtkAxesActor* axes2 = vtkAxesActor::New();
  // simulate a left-handed coordinate system
  //
  vtkTransform *transform = vtkTransform::New();
  transform->Identity();
  //transform->RotateY(180);
  reader->GetDirectionCosines()->Print(std::cout);
  transform->Concatenate(reader->GetDirectionCosines());
  //axes2->SetShaftTypeToCylinder();
  axes2->SetUserTransform( transform );
  //cube->SetUserTransform( transform ); // can't get it to work
  cube->GetAssembly()->SetUserTransform( transform ); // can't get it to work

  vtkPropAssembly* assembly = vtkPropAssembly::New();
  assembly->AddPart( axes2 );
  assembly->AddPart( cube );

  vtkOrientationMarkerWidget* widget = vtkOrientationMarkerWidget::New();
  //widget->SetOutlineColor( 0.9300, 0.5700, 0.1300 );
  widget->SetOrientationMarker( assembly );
  widget->SetInteractor( iren );
  //widget->SetViewport( 0.0, 0.0, 0.4, 0.4 );
  widget->SetEnabled( 1 );
  widget->InteractiveOff();
  widget->InteractiveOn();

  renwin->Render();
  iren->Start();

  // Clean up:
  reader->Delete();
  table->Delete();
```

```
  texture->Delete();
  plane->Delete();
  planeMapper->Delete();
  planeActor->Delete();
  ren->Delete();
  renwin->Delete();
  iren->Delete();

  return 0;
}
```

14.162 gdcmvolume.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkVersion.h"
#include "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#if VTK_MAJOR_VERSION < 7
#include "vtkVolumeTextureMapper3D.h"
#endif
#include "vtkFixedPointVolumeRayCastMapper.h"
#include "vtkInteractorStyleTrackballCamera.h"
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkImageClip.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkVersion.h"


// gdcmvolume gdcmData/GE_DLX-8-MONO2-Multiframe-Jpeg_Lossless.dcm
int main(int argc, char *argv[])
{
  if( argc < 2 ) return 1;
  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( argv[1] );
  reader->Update();

  // Create the renderers, render window, and interactor
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
  iren->SetRenderWindow(renWin);
  vtkRenderer *ren = vtkRenderer::New();
  renWin->AddRenderer(ren);

  // Create a transfer function mapping scalar value to opacity
  vtkPiecewiseFunction *oTFun = vtkPiecewiseFunction::New();
  //oTFun->AddSegment(0, 1.0, 256, 0.1);
  oTFun->AddSegment(0, 1.0, 240, 0.1);

  vtkColorTransferFunction *cTFun = vtkColorTransferFunction::New();
  cTFun->AddRGBPoint(   0, 1.0, 1.0, 1.0 );
  //cTFun->AddRGBPoint( 255, 1.0, 1.0, 1.0 );
  cTFun->AddRGBPoint( 240, 1.0, 1.0, 1.0 );

  // Need to crop to actually see minimum intensity
  vtkImageClip *clip = vtkImageClip::New();
  clip->SetInputConnection( reader->GetOutputPort() );
  clip->SetOutputWholeExtent(0,66,0,66,30,37);
  clip->ClipDataOn();

  vtkVolumeProperty *property = vtkVolumeProperty::New();
```

```
  property->SetScalarOpacity(oTFun);
  property->SetColor(cTFun);
  property->SetInterpolationTypeToLinear();

  vtkFixedPointVolumeRayCastMapper *mapper = vtkFixedPointVolumeRayCastMapper::New();
  mapper->SetBlendModeToMinimumIntensity();
  mapper->SetInputConnection( reader->GetOutputPort() );

  vtkVolume *volume = vtkVolume::New();
  volume->SetMapper(mapper);
  volume->SetProperty(property);


  ren->AddViewProp(volume);

  renWin->Render();
    {
    iren->Start();
    }

  volume->Delete();
  mapper->Delete();
  property->Delete();
  clip->Delete();
  cTFun->Delete();
  oTFun->Delete();
  reader->Delete();
  renWin->Delete();
  iren->Delete();
  ren->Delete();


 return 0;
}
```

14.163 offscreenimage.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMImageReader.h"
#include "vtkRenderWindow.h"
#include "vtkRenderer.h"
#include "vtkImageMapToWindowLevelColors.h"
#include "vtkImageActor.h"
#include "vtkPNGWriter.h"
#include "vtkWindowToImageFilter.h"
#include "vtkMedicalImageProperties.h"
#include "vtkVersion.h"

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    return 1;
    }
  const char *filename = argv[1];

  vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
  reader->SetFileName( filename );
  reader->Update(); // important to read the window/level info

  vtkMedicalImageProperties *prop = reader->GetMedicalImageProperties();

  vtkRenderWindow *renWin = vtkRenderWindow::New();
```

```
  renWin->OffScreenRenderingOn();

  vtkRenderer *renderer = vtkRenderer::New();
  renWin->AddRenderer(renderer);

  vtkImageMapToWindowLevelColors *windowlevel = vtkImageMapToWindowLevelColors::New();
#if (VTK_MAJOR_VERSION >= 6)
  windowlevel->SetInputConnection( reader->GetOutputPort() );
#else
  windowlevel->SetInput( reader->GetOutput() );
#endif
  unsigned int n = prop->GetNumberOfWindowLevelPresets();
  if( n )
    {
    // Take the first one by default:
    const double *wl = prop->GetNthWindowLevelPreset(0);
    windowlevel->SetWindow( wl[0] );
    windowlevel->SetLevel( wl[1] );
    }

  vtkImageActor *actor = vtkImageActor::New();
#if (VTK_MAJOR_VERSION >= 6)
  actor->SetInputData( windowlevel->GetOutput() );
#else
  actor->SetInput( windowlevel->GetOutput() );
#endif

  renderer->AddActor( actor );

  renWin->Render();

  vtkWindowToImageFilter *w2if = vtkWindowToImageFilter::New();
  w2if->SetInput ( renWin );

  vtkPNGWriter *wr = vtkPNGWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
  wr->SetInputConnection( w2if->GetOutputPort() );
#else
  wr->SetInput( w2if->GetOutput() );
#endif
  wr->SetFileName ( "offscreenimage.png" );
  wr->Write();

  reader->Delete();
  renWin->Delete();
  renderer->Delete();
  windowlevel->Delete();
  actor->Delete();
  w2if->Delete();
  wr->Delete();

  return 0;
}
```

14.164 reslicesphere.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
//
// Load a DICOM series.
// Position a sphere within the volume.
// Allow the user to change between Axial, Sagittal, Coronal, and
//     Oblique view of the images and move through the slices.
// The display should show the resliced image and the cross section
// of the sphere intersecting that plane.
```

```
//


/*
from  Scott Johnson /Scott Johnson neuwave com/
to  VTK /vtkusers vtk.org/
date  Tue, May 11, 2010 at 7:01 PM
*/
#include <string>

#include <vtkDICOMImageReader.h>
#include <vtkStringArray.h>
#include <vtkDirectory.h>
#include <vtkImageThreshold.h>
#include <vtkImageShiftScale.h>
#include <vtkImageReslice.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageViewer2.h>
#include <vtkSphereSource.h>
#include <vtkPolyDataMapper.h>
#include <vtkPlane.h>
#include <vtkCutter.h>
#include <vtkActor.h>
#include <vtkCommand.h>
#include <vtkSmartPointer.h>
#include <vtkMatrix4x4.h>
#include <vtkInteractorObserver.h>
#include <vtkProperty.h>
#include <vtkRenderer.h>
#include <vtkImageData.h>
#include <vtkImageActor.h>
#include "vtkTransformPolyDataFilter.h"
#include <vtkCamera.h>
#include <vtkMath.h>
#include <vtkTransform.h>
#include <vtkTextActor.h>
#include <vtkActor2D.h>
#include <vtkPolyDataMapper2D.h>
#include <vtkProperty2D.h>
#include <vtkGDCMImageReader.h>
#include <vtkImageChangeInformation.h>
#include <vtkVersion.h>

#include "gdcmDirectory.h"
#include "gdcmTesting.h"
#include "gdcmIPPSorter.h"

// Change to match the path to find Raw_0.vti or provide
// the parameter when starting ResliceSphere.

const double sphereCenter[3]={74, 219, 70};

// Angles (0, 0, 0)
const double AxialMatrix[]    = { 1.0,  0.0,  0.0,  0.0,
                                  0.0,  1.0,  0.0,  0.0,
                                  0.0,  0.0,  1.0,  0.0,
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (0, 90, 0)
const double SagittalMatrix[] = { 0.0,  0.0,  1.0,  0.0,
                                  0.0,  1.0,  0.0,  0.0,
                                 -1.0,  0.0,  0.0,  0.0,
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (-90, 0, 0)
const double CoronalMatrix[]  = { 1.0,  0.0,  0.0,  0.0,
                                  0.0,  0.0,  1.0,  0.0,
                                  0.0, -1.0,  0.0,  0.0,
                                  0.0,  0.0,  0.0,  1.0 };
// Angles (0, 90, 31)
const double ObliqueMatrix[] =  { 0.0, -0.515038, 0.857167, 0.0,
                                  0.0,  0.857167, 0.515038, 0.0,
                                 -1.0,  0.0,      0.0,      0.0,
                                  0.0,  0.0,      0.0,      1.0 };

class ResliceRender;

// Class to handle key press events.
class KeyCallback : public vtkCommand
{
public:
    static KeyCallback* New()
    {
```

```
        return new KeyCallback();
    }

    void Execute(vtkObject* caller, unsigned long eventId, void *calldata);
    void SetCallbackData(ResliceRender* reslice);

protected:
    ResliceRender* _reslice;
};

class ResliceRender
{
public:
    typedef enum _ORIENTATION
    {
        AXIAL = 0,
        SAGITTAL = 1,
        CORONAL = 2,
        OBLIQUE = 3
    } ORIENTATION;

    ResliceRender()
    {
        _orientation=AXIAL;
    }

    ~ResliceRender()
    {
        _transform->Delete();
        _reader->Delete();
        _reslice->Delete();
        _interactor->Delete();
        _imageViewer->Delete();

        _sphere->Delete();
        _sphereMapper->Delete();
        _sphereActor->Delete();

        _plane->Delete();
        _cutter->Delete();
        _polyTransform->Delete();
        _ROIMapper->Delete();
        _ROIActor->Delete();

        _annotation->Delete();
    }

    void CreatePipeline(const char* fileName)
    {
        vtkProperty2D* props;

        //_reader=vtkXMLImageDataReader::New();
        //_reader->SetFileName(fileName);
        //_reader->Update();

        //_reader=qzDICOMImageReader::New();
        _reader=vtkGDCMImageReader::New();

      //vtkDirectory *d = vtkDirectory::New();
      //d->Open(fileName);
      //d->Print( std::cout );
      gdcm::Directory d;
      d.Load(fileName);
      gdcm::Directory::FilenamesType const &files = d.GetFilenames();

  gdcm::IPPSorter s;
  s.SetComputeZSpacing( true );
  s.SetZSpacingTolerance( 1e-3 );
  bool b = s.Sort( files );
  if( !b )
    {
    std::cerr « "Failed to sort:" « fileName « std::endl;
    //return ;
    }
  //std::cout « "Sorting succeeded:" « std::endl;
  //s.Print( std::cout );

  //std::cout « "Found z-spacing:" « std::endl;
  //std::cout « s.GetZSpacing() « std::endl;
```

```
  double ippzspacing = s.GetZSpacing();

  const std::vector<std::string> & sorted = s.GetFilenames();
  vtkStringArray *vtkfiles = vtkStringArray::New();
  std::vector< std::string >::const_iterator it = sorted.begin();
  for( ; it != sorted.end(); ++it)
    {
    const std::string &f = *it;
    vtkfiles->InsertNextValue( f.c_str() );
    }

        //_reader->SetDirectoryName(fileName);
        //_reader->SetFileNames( d->GetFiles() );
        _reader->SetFileNames( vtkfiles );
        _reader->Update();

#ifndef vtkFloatingPointType
#define vtkFloatingPointType double
#endif
  const vtkFloatingPointType *spacing = _reader->GetOutput()->GetSpacing();

  vtkImageChangeInformation *v16 = vtkImageChangeInformation::New();
#if (VTK_MAJOR_VERSION >= 6)
  v16->SetInputConnection( _reader->GetOutputPort() );
#else
  v16->SetInput( _reader->GetOutput() );
#endif
  v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
  v16->Update();


        _threshold=vtkImageThreshold::New();
        _threshold->ThresholdByUpper(-3024.0);
        _threshold->ReplaceOutOn();
        _threshold->SetOutValue(0.0);
        _threshold->SetInputConnection(v16->GetOutputPort());

        _shift=vtkImageShiftScale::New();
        _shift->SetShift(0);
        _shift->SetScale(1);
        _shift->SetInputConnection(_threshold->GetOutputPort());

        // Initialize the reslice with an axial orientation.
        vtkSmartPointer<vtkMatrix4x4> matrix =
            vtkSmartPointer<vtkMatrix4x4>::New();
        matrix->Identity();

        _transform = vtkTransform::New();
        _transform->SetMatrix(matrix);

        _reslice = vtkImageReslice::New();
        _reslice->SetOutputDimensionality(3);

        // PROBLEM:
        // The original intent was to connect the same transform
        // to the vtkImageReslice and vtkTransformPolyDataFilter,
        // but the resulting reslices appear different using the
        // vtkTransform as opposed to explicitly setting the
        // reslice axes via SetResliceAxes.  Also, if the vtkTransform
        // is connected and orientated other than axial, the extents
        // don't seem to update resulting in VTK believing the slice
        // is out of range.

        //_reslice->SetResliceTransform(_transform);
        _reslice->SetResliceAxes(matrix);
        //_reslice->SetInputConnection(_reader->GetOutputPort());
        _reslice->SetInputConnection(_shift->GetOutputPort());

        // Create the sphere target shape.
        _sphere=vtkSphereSource::New();
        _sphere->SetRadius(7.0);
        _sphere->SetThetaResolution(16);
        _sphere->SetPhiResolution(16);
        _sphere->SetCenter(sphereCenter[0], sphereCenter[1], sphereCenter[2]);

        _sphereMapper=vtkPolyDataMapper::New();
        _sphereMapper->SetInputConnection(_sphere->GetOutputPort());

        _sphereActor=vtkActor::New();
        _sphereActor->SetMapper(_sphereMapper);
        _sphereActor->PickableOff();
```

```
        _sphereActor->GetProperty()->SetColor(1.0, 0.0, 0.0);
        _sphereActor->GetProperty()->SetEdgeColor(1.0, 0.0, 0.0);
        _sphereActor->GetProperty()->SetDiffuseColor(1.0, 0.0, 0.0);
        _sphereActor->SetVisibility(true);

        // Create the cutting pipeline.
        // This plane will be positioned in the original image coordinate system.
        _plane = vtkPlane::New();
        _plane->SetNormal(0.0, 0.0, 1.0);

        _cutter = vtkCutter::New();
        _cutter->SetInputConnection(_sphere->GetOutputPort());
        _cutter->SetCutFunction(_plane);
        _cutter->GenerateCutScalarsOn();
        _cutter->SetValue(0, 0.5);

        // The transform attached to _polyTransform should move the cut
        // ROI into the resliced coordinate system, which should be the
        // same as the coordinate system of the resliced images.
        // PROBLEM:  It doesn't.
        _polyTransform = vtkTransformPolyDataFilter::New();
        _polyTransform->SetTransform(_transform);
        _polyTransform->SetInputConnection(_cutter->GetOutputPort());

        _ROIMapper = vtkPolyDataMapper2D::New();
        _ROIMapper->SetInputConnection(_polyTransform->GetOutputPort());

    vtkCoordinate* coordinate = vtkCoordinate::New();
    coordinate->SetCoordinateSystemToWorld();
    _ROIMapper->SetTransformCoordinate(coordinate);

        _ROIActor = vtkActor2D::New();
        _ROIActor->SetMapper(_ROIMapper);

        // Make sure the cut can be seen, especially the edges.
        props=_ROIActor->GetProperty();
        props->SetLineWidth(2);
        props->SetOpacity(1.0);
//        props->EdgeVisibilityOn();
//        props->SetDiffuse(0.8);
//        props->SetSpecular(0.3);
//        props->SetSpecularPower(20);
//        props->SetRepresentationToSurface();
//        props->SetDiffuseColor(1.0, 0.0, 0.0);
//        props->SetEdgeColor(1.0, 0.0, 0.0);
        props->SetColor(1.0, 0.0, 0.0);

        _interactor = vtkRenderWindowInteractor::New();

        // Create the image viewer and add the actor with the cut ROI.
        _imageViewer = vtkImageViewer2::New();
        _imageViewer->SetupInteractor(_interactor);
        _imageViewer->SetSize(400, 400);
        _imageViewer->SetColorWindow(1024);
        _imageViewer->SetColorLevel(800);
        _imageViewer->SetInputConnection(_reslice->GetOutputPort());
        _imageViewer->GetImageActor()->SetOpacity(0.5);

        _annotation = vtkTextActor::New();
        _annotation->SetTextScaleModeToViewport();
        _imageViewer->GetRenderer()->AddActor(_annotation);

        // Add the cut shape actor to the renderer.
        _imageViewer->GetRenderer()->AddActor(_ROIActor);

        // Set up the key handler.
        vtkSmartPointer<KeyCallback> callback = vtkSmartPointer<KeyCallback>::New();
        callback->SetCallbackData(this);
        _interactor->AddObserver(vtkCommand::KeyPressEvent, callback);

        _interactor->Initialize();
    }

    void Start()
    {
        _interactor->Start();
    }

    void ResetOrientation()
    {
        vtkSmartPointer<vtkMatrix4x4> matrix =
```

```
            vtkSmartPointer<vtkMatrix4x4>::New();
        matrix->Identity();

        SetOrientation(matrix);
    }

    // Make sure the orientation of the vtkImageReslice and
    // vtkTransform are in sync.
    void SetOrientation(vtkMatrix4x4* matrix)
    {
        _reslice->SetResliceAxes(matrix);
        _reslice->Update();

    vtkMatrix4x4* inverse = vtkMatrix4x4::New();
    vtkMatrix4x4::Invert(matrix, inverse);

        _transform->SetMatrix(inverse);
        _transform->Update();
    }

    // Set the current slice of the current view.
    void SetSlice(int slice)
    {
        std::stringstream posString;

        double    center[3];
        double    spacing[3];
        double    origin[3];
        double    point[4];
        double    newPoint[4];

        vtkImageData* imageData;
        int newSlice;

        // Try to make sure the extents of the reslice are updated.
        // PROBLEM:  It doesn't seem to work when changing the orientation.
        imageData=vtkImageData::SafeDownCast(_reslice->GetOutput());
#if (VTK_MAJOR_VERSION >= 6)
        assert(0);
#else
        imageData->UpdateInformation();
#endif

        // Let vtkImageViewer2 handle the slice limits.
        _imageViewer->SetSlice(slice);
        newSlice=GetSlice();

        imageData->GetCenter(center);
        imageData->GetSpacing(spacing);
        imageData->GetOrigin(origin);

        // Compute the position of the center of the slice based on the
        // spacing of the slices.  The resliced axis will always
        // be the "Z" axis.
        point[0]=center[0];
        point[1]=center[1];
        point[2]=(newSlice * spacing[2]) + origin[2];
        point[3]=1.0;

        // Convert the coordinate from the reslice coordinate system to the
        // original image coordinate system.
        // PROBLEM:  Logically this seems like it should have been multiplied
        // by the inverse to translate from the resliced coordinate system to
        // the original coordinate system.  However, multiplying by the inverse
        // sticks the plane in the wrong place completely.  Using the original
        // matrix at least gets the Z coordinate right.
        vtkMatrix4x4* matrix=_reslice->GetResliceAxes();
        vtkSmartPointer<vtkMatrix4x4> inverse =
            vtkSmartPointer<vtkMatrix4x4>::New();
        vtkMatrix4x4::Invert(matrix, inverse);

        matrix->MultiplyPoint(point, newPoint);
        _plane->SetOrigin(newPoint[0], newPoint[1], newPoint[2]);

        // Annotate the image.
        posString « "Position: (" « newPoint[0] « ", " « newPoint[1]
                  « ", " « newPoint[2] « ")  Slice: " « newSlice;
        _annotation->SetInput(posString.str().c_str());

        _imageViewer->Render();
```

```
    }

    int GetSlice()
    {
        return _imageViewer->GetSlice();
    }

    // Set the orientation of the view.
    void SetOrientation(ResliceRender::ORIENTATION orientation)
    {
        vtkCamera* camera=_imageViewer->GetRenderer()->GetActiveCamera();

        double spacing[3];
        double origin[3];
        double point[4];
        double newPoint[4];
        double initialPosition;
        double xDirCosine[3];
        double yDirCosine[3];
        double zDirCosine[3];
        double normal[3];

        vtkImageData* imageData;

        vtkSmartPointer<vtkMatrix4x4> matrix =
            vtkSmartPointer<vtkMatrix4x4>::New();

        _orientation=orientation;

        // Reset ViewUp
        camera->SetViewUp(0.0, 1.0, 0.0);

        // Compute the cut plane position to the input coordinate system.
        imageData=vtkImageData::SafeDownCast(_reslice->GetInput());
#if (VTK_MAJOR_VERSION >= 6)
        assert(0);
#else
        imageData->UpdateInformation();
#endif
        imageData->GetSpacing(spacing);
        imageData->GetOrigin(origin);

        point[0]=origin[0];
        point[1]=origin[1];
        point[2]=origin[2];
        point[3]=1.0;

        switch (_orientation)
        {
        case AXIAL:
            matrix->DeepCopy(AxialMatrix);
            initialPosition=sphereCenter[2];
            break;

        case CORONAL:
            matrix->DeepCopy(CoronalMatrix);
            initialPosition=sphereCenter[1];
            break;

        case SAGITTAL:
            matrix->DeepCopy(SagittalMatrix);
            initialPosition=sphereCenter[0];
            break;

        case OBLIQUE:
            matrix->DeepCopy(ObliqueMatrix);
            initialPosition=sphereCenter[2];
            break;
        }

        // Move the origin from the original image coordinate system to the
        // resliced image coordinate system.
        matrix->MultiplyPoint(point, newPoint);
        matrix->SetElement(0, 3, newPoint[0]);
        matrix->SetElement(1, 3, newPoint[1]);
        matrix->SetElement(2, 3, newPoint[2]);

        ResetOrientation();
        SetOrientation(matrix);

        // Compute the cutting plane normal and set it.
```

```
        // PROBLEM:  If the transformation is connected rather than
        // using SetResliceAxes, the Direction Cosines do not reflect
        // the orientation of the vtkImageReslice.
        _reslice->GetResliceAxesDirectionCosines(xDirCosine, yDirCosine,
                                                 zDirCosine);
        vtkMath::Cross(xDirCosine, yDirCosine, normal);
        _plane->SetNormal(normal);

        // Set the extents and spacing of the reslice to account for
        // all of the data.
        _reslice->SetOutputExtentToDefault();
        _reslice->SetOutputSpacing(spacing[0], spacing[0], spacing[0]);

        // Force the vtkImageViewer2 to update.
        // PROBLEM:  The whole extent does not seem to be set in time
        // for the first render.  This results in an error because the
        // slice is positioned outside the old bounds.
#if (VTK_MAJOR_VERSION >= 6)
        _imageViewer->SetInputData(NULL);
#else
        _imageViewer->SetInput(NULL);
#endif
        _imageViewer->SetInputConnection(_reslice->GetOutputPort());

        _imageViewer->GetRenderer()->ResetCameraClippingRange();
        _imageViewer->GetRenderer()->ResetCamera();

        // Set the initial slice to be at the center of the sphere.
        // Divide by the spacing because this will be undone in SetSlice.
        SetSlice( (int)(initialPosition / spacing[0]));
    }

    vtkRenderWindowInteractor* GetInteractor()
    {
        return _interactor;
    }

protected:
    ORIENTATION                 _orientation;

    //qzDICOMImageReader*        _reader;
    vtkGDCMImageReader*        _reader;
    vtkImageThreshold*          _threshold;
    vtkImageShiftScale*         _shift;
    vtkImageReslice*            _reslice;
    vtkRenderWindowInteractor*  _interactor;
    vtkImageViewer2*            _imageViewer;

    vtkSphereSource*            _sphere;
    vtkPolyDataMapper*          _sphereMapper;
    vtkActor*                   _sphereActor;

    vtkPlane*                   _plane;
    vtkCutter*                  _cutter;
    vtkTransform*               _transform;
    vtkTransformPolyDataFilter* _polyTransform;
    vtkPolyDataMapper2D*          _ROIMapper;
    vtkActor2D*                   _ROIActor;

    vtkTextActor*               _annotation;
};


// Catch KeyPress events.
// Up Arrow   - increases the slice
// Down Arrow - decreases the slice
// 'A'        - sets the view to Axial
// 'S'        - sets the view to Sagittal
// 'C'        - sets the view to Coronal
// 'O'        - set the view to Oblique

void KeyCallback::Execute(vtkObject* caller, unsigned long eventId, void *calldata)
{
  (void)caller;
  (void)eventId;
  (void)calldata;
    std::string sym=_reslice->GetInteractor()->GetKeySym();

    if (!sym.compare("Up"))
    {
        _reslice->SetSlice(_reslice->GetSlice() + 1);
```

```
    }
    else if (!sym.compare("Down"))
    {
        _reslice->SetSlice(_reslice->GetSlice() - 1);
    }
    else if ((!sym.compare("A")) || (!sym.compare("a")))
    {
        _reslice->SetOrientation(ResliceRender::AXIAL);
    }
    else if ((!sym.compare("C")) || (!sym.compare("c")))
    {
        _reslice->SetOrientation(ResliceRender::CORONAL);
    }
    else if ((!sym.compare("S")) || (!sym.compare("s")))
    {
        _reslice->SetOrientation(ResliceRender::SAGITTAL);
    }
    else if ((!sym.compare("O")) || (!sym.compare("o")))
    {
        _reslice->SetOrientation(ResliceRender::OBLIQUE);
    }
}

void KeyCallback::SetCallbackData(ResliceRender* reslice)
{
    _reslice=reslice;
}

// Usage:  ResliceSphere [fileName]
int main(int argc, char *argv[])
{
    ResliceRender render;

    if (argc == 1)
      {
      const char *root = gdcm::Testing::GetDataExtraRoot();
      std::string dir3 = root;
      dir3 += "/gdcmSampleData/ForSeriesTesting/Dentist/images/";
      render.CreatePipeline(dir3.c_str());
      }
    else
      {
      render.CreatePipeline(argv[1]);
      }

    render.SetOrientation(ResliceRender::AXIAL);
    render.Start();

    return EXIT_SUCCESS;
}
```

14.165 rtstructapp.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "vtkGDCMPolyDataReader.h"
#include "vtkGDCMPolyDataWriter.h"

#include "vtkPolyDataWriter.h"
#include "vtkPolyDataMapper.h"
#include "vtkPolyDataMapper2D.h"
#include "vtkActor2D.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkMedicalImageProperties.h"
#include "vtkRenderWindow.h"
```

```
#include "vtkRenderer.h"
#include "vtkCamera.h"
#include "vtkProperty.h"
#include "vtkProperty2D.h"
#include "vtkAppendPolyData.h"
#include "vtkImageData.h"
#include "vtkVersion.h"

/*
 * Small example to read in a RTSTUCT and write it out (displays it too).
 */

// gdcmDataExtra/gdcmNonImageData/exRT_Structure_Set_Storage.dcm
// gdcmDataExtra/gdcmNonImageData/RTSTRUCT_1.3.6.1.4.1.22213.1.1396.2.dcm
// gdcmDataExtra/gdcmNonImageData/RT/RTStruct.dcm

int main(int argc, char *argv[])
{
  if( argc < 3 )
    {
    std::cerr « argv[0] « " input.dcm output.dcm\n";
    return 1;
    }
  const char * filename = argv[1];
  const char * outfilename = argv[2];
  vtkGDCMPolyDataReader * reader = vtkGDCMPolyDataReader::New();
  reader->SetFileName( filename );
  reader->Update();

  //std::cout « reader->GetMedicalImageProperties()->GetStudyDate() « std::endl;

  vtkGDCMPolyDataWriter * writer = vtkGDCMPolyDataWriter::New();
  writer->SetNumberOfInputPorts( reader->GetNumberOfOutputPorts() );
  writer->SetFileName( outfilename );
  for(int num = 0; num < reader->GetNumberOfOutputPorts(); ++num )
#if (VTK_MAJOR_VERSION >= 6)
    writer->SetInputConnection( num, reader->GetOutputPort(num) );
#else
    writer->SetInput( num, reader->GetOutput(num) );
#endif
  //doesn't look like the medical properties are actually written out
  writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
  writer->SetRTStructSetProperties( reader->GetRTStructSetProperties() );
  writer->Write();

  // print reader output:
  reader->Print( std::cout );
  // print first output:
  reader->GetOutput()->Print( std::cout );

 vtkAppendPolyData *append = vtkAppendPolyData::New();

 int n = reader->GetNumberOfOutputPorts();
 for(int i = 0; i < n; ++i)
   {
#if (VTK_MAJOR_VERSION >= 6)
   append->AddInputConnection( reader->GetOutputPort(i) );
#else
   append->AddInput( reader->GetOutput(i) );
#endif
   }

  // Now we'll look at it.
  vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
#if (VTK_MAJOR_VERSION >= 6)
  cubeMapper->SetInputConnection( append->GetOutputPort());
#else
  cubeMapper->SetInput( append->GetOutput());
#endif
  cubeMapper->SetScalarRange(0,7);
  vtkActor *cubeActor = vtkActor::New();
  cubeActor->SetMapper(cubeMapper);
  vtkProperty * property = cubeActor->GetProperty();
  property->SetRepresentationToWireframe();

  vtkRenderer *renderer = vtkRenderer::New();
  vtkRenderWindow *renWin = vtkRenderWindow::New();
  renWin->AddRenderer(renderer);

  vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
```

```
  iren->SetRenderWindow(renWin);

  renderer->AddActor(cubeActor);
  renderer->ResetCamera();
  renderer->SetBackground(1,1,1);

  renWin->SetSize(300,300);

  renWin->Render();
  iren->Start();

  reader->Delete();
  append->Delete();
  cubeMapper->Delete();
  cubeActor->Delete();
  renderer->Delete();
  renWin->Delete();
  iren->Delete();
  writer->Delete();

  return 0;
}
```

14.166 threadgdcm.cxx

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
#include "gdcmReader.h"
#include "gdcmImageReader.h"
#include "gdcmDirectory.h"
#include "gdcmSystem.h"

#include "vtkImageData.h"
#include "vtkStructuredPointsWriter.h"
#include "vtkVersion.h"

#include <pthread.h>

struct threadparams
{
  const char **filenames;
  size_t nfiles;
  char *scalarpointer;
// TODO I should also pass in the dim of the reference image just in case
};

void *ReadFilesThread(void *voidparams)
{
  const threadparams *params = static_cast<const threadparams *> (voidparams);

  const size_t nfiles = params->nfiles;
  for(unsigned int file = 0; file < nfiles; ++file)
    {
    /*
    // TODO: update progress
    pthread_mutex_lock(&params->lock);
    //section critique
    ReadingProgress+=params->stepProgress;
    pthread_mutex_unlock(&params->lock);
    */
    const char *filename = params->filenames[file];
    //std::cerr « filename « std::endl;

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
```

```
    try
      {
      if( !reader.Read() )
        {
        std::cerr « "Failed to read: " « filename « std::endl;
        break;
        }
      }
    catch( ... )
      {
      std::cerr « "Failed to read: " « filename « std::endl;
      break;
      }

    const gdcm::Image &image = reader.GetImage();
    unsigned long len = image.GetBufferLength();
    char * pointer = params->scalarpointer;
#if 0
    char *tempimage = new char[len];
    image.GetBuffer(tempimage);

    memcpy(pointer + file*len, tempimage, len);
    delete[] tempimage;
#else
    char *tempimage = pointer + file * len;
    image.GetBuffer(tempimage);
#endif
    }

  return voidparams;
}

void ShowFilenames(const threadparams &params)
{
  std::cout « "start" « std::endl;
  for(unsigned int i = 0; i < params.nfiles; ++i)
    {
    const char *filename = params.filenames[i];
    std::cout « filename « std::endl;
    }
  std::cout « "end" « std::endl;
}

void ReadFiles(size_t nfiles, const char *filenames[])
{
  // \precondition: nfiles > 0
  assert( nfiles > 0 );
  const char *reference= filenames[0]; // take the first image as reference

  gdcm::ImageReader reader;
  reader.SetFileName( reference );
  if( !reader.Read() )
    {
    // That would be very bad...
    assert(0);
    }

  const gdcm::Image &image = reader.GetImage();
  gdcm::PixelFormat pixeltype = image.GetPixelFormat();
  unsigned long len = image.GetBufferLength();
  const unsigned int *dims = image.GetDimensions();
  unsigned short pixelsize = pixeltype.GetPixelSize();
  (void)pixelsize;
  assert( image.GetNumberOfDimensions() == 2 );

  vtkImageData *output = vtkImageData::New();
  output->SetDimensions(dims[0], dims[1], (int)nfiles);

#if (VTK_MAJOR_VERSION >= 6)
  int numscal = pixeltype.GetSamplesPerPixel();
  switch( pixeltype )
    {
  case gdcm::PixelFormat::INT8:
    output->AllocateScalars( VTK_SIGNED_CHAR, numscal );
    break;
  case gdcm::PixelFormat::UINT8:
    output->AllocateScalars( VTK_UNSIGNED_CHAR, numscal );
    break;
  case gdcm::PixelFormat::INT16:
    output->AllocateScalars( VTK_SHORT, numscal );
    break;
```

```
  case gdcm::PixelFormat::UINT16:
    output->AllocateScalars( VTK_UNSIGNED_SHORT, numscal );
    break;
  case gdcm::PixelFormat::INT32:
    output->AllocateScalars( VTK_INT, numscal );
    break;
  case gdcm::PixelFormat::UINT32:
    output->AllocateScalars( VTK_UNSIGNED_INT, numscal );
    break;
  default:
    assert(0);
    }
#else
  switch( pixeltype )
    {
  case gdcm::PixelFormat::INT8:
#if (VTK_MAJOR_VERSION >= 5) || ( VTK_MAJOR_VERSION == 4 && VTK_MINOR_VERSION > 5 )
    output->SetScalarType ( VTK_SIGNED_CHAR );
#else
    output->SetScalarType ( VTK_CHAR );
#endif
    break;
  case gdcm::PixelFormat::UINT8:
    output->SetScalarType ( VTK_UNSIGNED_CHAR );
    break;
  case gdcm::PixelFormat::INT16:
    output->SetScalarType ( VTK_SHORT );
    break;
  case gdcm::PixelFormat::UINT16:
    output->SetScalarType ( VTK_UNSIGNED_SHORT );
    break;
  case gdcm::PixelFormat::INT32:
    output->SetScalarType ( VTK_INT );
    break;
  case gdcm::PixelFormat::UINT32:
    output->SetScalarType ( VTK_UNSIGNED_INT );
    break;
  default:
    assert(0);
    }
  output->SetNumberOfScalarComponents ( pixeltype.GetSamplesPerPixel() );
  output->AllocateScalars();
#endif
  char * scalarpointer = static_cast<char*>(output->GetScalarPointer());

  const unsigned int nthreads = 4;
  threadparams params[nthreads];

  //pthread_mutex_t lock;
  //pthread_mutex_init(&lock, NULL);

  pthread_t *pthread = new pthread_t[nthreads];

  // There is nfiles, and nThreads
  assert( nfiles > nthreads );
  const size_t partition = nfiles / nthreads;
  for (unsigned int thread=0; thread < nthreads; ++thread)
    {
    params[thread].filenames = filenames + thread * partition;
    params[thread].nfiles = partition;
    if( thread == nthreads - 1 )
      {
      // There is slightly more files to process in this thread:
      params[thread].nfiles += nfiles % nthreads;
      }
    assert( thread * partition < nfiles );
    params[thread].scalarpointer = scalarpointer + thread * partition * len;
    //assert( params[thread].scalarpointer < scalarpointer + 2 * dims[0] * dims[1] * dims[2] );
    // start thread:
    int res = pthread_create( &pthread[thread], NULL, ReadFilesThread, &params[thread]);
    if( res )
      {
      std::cerr « "Unable to start a new thread, pthread returned: " « res « std::endl;
      assert(0);
      }
    //ShowFilenames(params[thread]);
    }
// DEBUG
  size_t total = 0;
  for (unsigned int thread=0; thread < nthreads; ++thread)
    {
```

```
    total += params[thread].nfiles;
    }
  assert( total == nfiles );
// END DEBUG

  for (unsigned int thread=0;thread<nthreads;thread++)
    {
    pthread_join( pthread[thread], NULL);
    }
  delete[] pthread;

  //pthread_mutex_destroy(&lock);

  // For some reason writing down the file is painfully slow...
  vtkStructuredPointsWriter *writer = vtkStructuredPointsWriter::New();
#if (VTK_MAJOR_VERSION >= 6)
  writer->SetInputData( output );
#else
  writer->SetInput( output );
#endif
  writer->SetFileName( "/tmp/threadgdcm.vtk" );
  writer->SetFileTypeToBinary();
  //writer->Write();
  writer->Delete();

  //output->Print( std::cout );
  output->Delete();
}

int main(int argc, char *argv[])
{
  if( argc < 2 )
    {
    std::cerr « argv[0] « " [directory|list of filenames]\n";
    return 1;
    }

  // Check if user pass in a single directory
  if( argc == 2 && gdcm::System::FileIsDirectory( argv[1] ) )
    {
    gdcm::Directory d;
    d.Load( argv[1] );
    gdcm::Directory::FilenamesType l = d.GetFilenames();
    const size_t nfiles = l.size();
    const char **filenames = new const char* [ nfiles ];
    for(unsigned int i = 0; i < nfiles; ++i)
      {
      filenames[i] = l[i].c_str();
      }
    ReadFiles(nfiles, filenames);
    delete[] filenames;
    }
  else
    {
    // Simply copy all filenames into the vector:
    const char **filenames = const_cast<const char**>(argv+1);
    const size_t nfiles = argc - 1;
    ReadFiles(nfiles, filenames);
    }


  return 0;
}
```

14.167 AWTMedical3.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
```

```
     PURPOSE.  See the above copyright notice for more information.

==========================================================================*/
package examples;

import vtk.*;
//import gdcm.*;

import vtk.util.VtkPanelContainer;
import vtk.util.VtkPanelUtil;
import vtk.util.VtkUtil;

import java.util.ArrayList;


import javax.swing.*;
import java.awt.*;
import java.io.File;

public class AWTMedical3  extends JComponent implements VtkPanelContainer {

  private vtkPanel renWin;

    vtkImageData ReadDataFile(File inSelectedFile){

        vtkImageData outImageData = null;
        Directory theDir = new Directory();

        String theInputDirectory = inSelectedFile.getPath();
        theDir.Load(theInputDirectory);

        Scanner theScanner = new Scanner();
        Tag theStudyTag = new Tag(0x0020,0x000d);
        Tag theSeriesTag = new Tag(0x0020,0x000e);
        theScanner.AddTag(theStudyTag);//get studies,
        theScanner.AddTag(theSeriesTag);//get studies,
        theScanner.Scan(theDir.GetFilenames());

        FilenamesType theStudyValues = theScanner.GetOrderedValues(theStudyTag);
        long theNumStudies = theStudyValues.size();
        //for now, take the first study, and nothing else.
        //and the return is actually not FilenamesType, just a
        //vector of strings
        if (theNumStudies != 1)
            return outImageData;
        String theStudyVal = theStudyValues.get(0);
        //now, get all the values from the scanner that are in that
        //study, then from that get their different series
        FilenamesType theFilenames =
                theScanner.GetAllFilenamesFromTagToValue(theStudyTag, theStudyVal);

        //from that set of filenames, isolate individual series
        //conclude that singleton series = RT struct (can do further
        //checking for things like MIPs and the like)
        //and multiple series entries = volumetric data
        theScanner.Scan(theFilenames);
        FilenamesType theSeriesValues = theScanner.GetOrderedValues(theSeriesTag);
        String studyUID = theScanner.GetValue(theScanner.GetFilenames().get(0), theStudyTag);
        long theNumSeries = theSeriesValues.size();
        for (int i = 0; i < theNumSeries; i++) {
            FilenamesType theSeriesFiles =
                theScanner.GetAllFilenamesFromTagToValue(theSeriesTag, theSeriesValues.get(i));
            long theNumFilesInSeries = theSeriesFiles.size();
            if (theNumFilesInSeries > 1) {//assume it's CT or volumetric data
                //for now, assume a single volume
                //could have multiples, like PET and CT

                IPPSorter sorter = new IPPSorter();
                sorter.SetComputeZSpacing(true);
                sorter.SetZSpacingTolerance(0.001);
                Boolean sorted = sorter.Sort(theSeriesFiles);
                if (!sorted){
                    //need some better way to handle failures here
                    return outImageData;
                }

                FilenamesType sortedFT = sorter.GetFilenames();
                long theSize = sortedFT.size();
                vtkStringArray sa = new vtkStringArray();
```

```
                ArrayList<String> theStrings = new ArrayList<String>();

                vtkGDCMImageReader gdcmReader = new vtkGDCMImageReader();
                for (int j = 0; j < theSize; j++) {
                    String theFileName = sortedFT.get(j);
                    if (gdcmReader.CanReadFile(theFileName) > 0){
                        theStrings.add(theFileName);
                        sa.InsertNextValue(theFileName);
                    } else {
                        //this is a busted series
                        //need some more appropriate error here
                        return outImageData;
                    }
                }

                gdcmReader.SetFileNames(sa);

                gdcmReader.Update();

                outImageData = gdcmReader.GetOutput();//the zeroth output should be the image
            }
        }
        String theImageInfo = "";
        if (outImageData != null){
            theImageInfo = outImageData.Print();
        }
        return outImageData;
    }

    //this function is a rewrite of Medical3 to see if data can
    //be loaded via gdcm easily
  public AWTMedical3(File inFile) {
    // Create the buttons.
    renWin = new vtkPanel();

    vtkImageData theImageData = ReadDataFile(inFile);

    // An isosurface, or contour value of 500 is known to correspond to the
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter skinExtractor = new vtkContourFilter();
    skinExtractor.SetInput(theImageData);
    skinExtractor.SetValue(0, 500);
    vtkPolyDataNormals skinNormals = new vtkPolyDataNormals();
    skinNormals.SetInput(skinExtractor.GetOutput());
    skinNormals.SetFeatureAngle(60.0);
//        vtkStripper skinStripper = new vtkStripper();
//        skinStripper.SetInput(skinNormals.GetOutput());
    vtkPolyDataMapper skinMapper = new vtkPolyDataMapper();
    skinMapper.SetInput(skinNormals.GetOutput());
    skinMapper.ScalarVisibilityOff();
    vtkActor skin = new vtkActor();
    skin.SetMapper(skinMapper);
    skin.GetProperty().SetDiffuseColor(1, .49, .25);
    skin.GetProperty().SetSpecular(.3);
    skin.GetProperty().SetSpecularPower(20);

    // An isosurface, or contour value of 1150 is known to correspond to the
    // skin of the patient. Once generated, a vtkPolyDataNormals filter is
    // is used to create normals for smooth surface shading during rendering.
    // The triangle stripper is used to create triangle strips from the
    // isosurface these render much faster on some systems.
    vtkContourFilter boneExtractor = new vtkContourFilter();
    boneExtractor.SetInput(theImageData);
    boneExtractor.SetValue(0, 1150);
    vtkPolyDataNormals boneNormals = new vtkPolyDataNormals();
    boneNormals.SetInput(boneExtractor.GetOutput());
    boneNormals.SetFeatureAngle(60.0);
    vtkStripper boneStripper = new vtkStripper();
    boneStripper.SetInput(boneNormals.GetOutput());
    vtkPolyDataMapper boneMapper = new vtkPolyDataMapper();
    boneMapper.SetInput(boneStripper.GetOutput());
    boneMapper.ScalarVisibilityOff();
    vtkActor bone = new vtkActor();
    bone.SetMapper(boneMapper);
    bone.GetProperty().SetDiffuseColor(1, 1, .9412);

    // An outline provides context around the data.
    vtkOutlineFilter outlineData = new vtkOutlineFilter();
```

```
outlineData.SetInput(theImageData);
vtkPolyDataMapper mapOutline = new vtkPolyDataMapper();
mapOutline.SetInput(outlineData.GetOutput());
vtkActor outline = new vtkActor();
outline.SetMapper(mapOutline);
outline.GetProperty().SetColor(0, 0, 0);

// Now we are creating three orthogonal planes passing through the
// volume. Each plane uses a different texture map and therefore has
// different coloration.

// Start by creating a black/white lookup table.
vtkLookupTable bwLut = new vtkLookupTable();
bwLut.SetTableRange(0, 2000);
bwLut.SetSaturationRange(0, 0);
bwLut.SetHueRange(0, 0);
bwLut.SetValueRange(0, 1);
bwLut.Build();

// Now create a lookup table that consists of the full hue circle (from
// HSV);.
vtkLookupTable hueLut = new vtkLookupTable();
hueLut.SetTableRange(0, 2000);
hueLut.SetHueRange(0, 1);
hueLut.SetSaturationRange(1, 1);
hueLut.SetValueRange(1, 1);
hueLut.Build();

// Finally, create a lookup table with a single hue but having a range
// in the saturation of the hue.
vtkLookupTable satLut = new vtkLookupTable();
satLut.SetTableRange(0, 2000);
satLut.SetHueRange(.6, .6);
satLut.SetSaturationRange(0, 1);
satLut.SetValueRange(1, 1);
satLut.Build();

// Create the first of the three planes. The filter vtkImageMapToColors
// maps the data through the corresponding lookup table created above.
// The vtkImageActor is a type of vtkProp and conveniently displays an
// image on a single quadrilateral plane. It does this using texture
// mapping and as a result is quite fast. (Note: the input image has to
// be unsigned char values, which the vtkImageMapToColors produces.);
// Note also that by specifying the DisplayExtent, the pipeline
// requests data of this extent and the vtkImageMapToColors only
// processes a slice of data.
vtkImageMapToColors sagittalColors = new vtkImageMapToColors();
sagittalColors.SetInput(theImageData);
sagittalColors.SetLookupTable(bwLut);
vtkImageActor sagittal = new vtkImageActor();
sagittal.SetInput(sagittalColors.GetOutput());
sagittal.SetDisplayExtent(32, 32, 0, 63, 0, 92);

// Create the second (axial); plane of the three planes. We use the same
// approach as before except that the extent differs.
vtkImageMapToColors axialColors = new vtkImageMapToColors();
axialColors.SetInput(theImageData);
axialColors.SetLookupTable(hueLut);
vtkImageActor axial = new vtkImageActor();
axial.SetInput(axialColors.GetOutput());
axial.SetDisplayExtent(0, 63, 0, 63, 46, 46);

// Create the third (coronal); plane of the three planes. We use the same
// approach as before except that the extent differs.
vtkImageMapToColors coronalColors = new vtkImageMapToColors();
coronalColors.SetInput(theImageData);
coronalColors.SetLookupTable(satLut);
vtkImageActor coronal = new vtkImageActor();
coronal.SetInput(coronalColors.GetOutput());
coronal.SetDisplayExtent(0, 63, 32, 32, 0, 92);

// It is convenient to create an initial view of the data. The FocalPoint
// and Position form a vector direction. Later on (ResetCamera() method)
// this vector is used to position the camera to look at the data in
// this direction.
vtkCamera aCamera = new vtkCamera();
aCamera.SetViewUp(0, 0, -1);
aCamera.SetPosition(0, 1, 0);
aCamera.SetFocalPoint(0, 0, 0);
```

```
    aCamera.ComputeViewPlaneNormal();

    // Actors are added to the renderer. An initial camera view is created.
    // The Dolly() method moves the camera towards the FocalPoint,
    // thereby enlarging the image.
    renWin.GetRenderer().AddActor(sagittal);
    renWin.GetRenderer().AddActor(axial);
    renWin.GetRenderer().AddActor(coronal);
    renWin.GetRenderer().AddActor(outline);
    renWin.GetRenderer().AddActor(skin);
    renWin.GetRenderer().AddActor(bone);

    // Turn off bone for this example.
    bone.VisibilityOff();

    // Set skin to semi-transparent.
    skin.GetProperty().SetOpacity(0.5);

    // An initial camera view is created.  The Dolly() method moves
    // the camera towards the FocalPoint, thereby enlarging the image.
    renWin.GetRenderer().SetActiveCamera(aCamera);
    renWin.GetRenderer().ResetCamera();
    aCamera.Dolly(1.5);

    // Set a background color for the renderer and set the size of the
    // render window (expressed in pixels).
    renWin.GetRenderer().SetBackground(1, 1, 1);
    VtkPanelUtil.setSize(renWin, 640, 480);

    // Note that when camera movement occurs (as it does in the Dolly()
    // method), the clipping planes often need adjusting. Clipping planes
    // consist of two planes: near and far along the view direction. The
    // near plane clips out objects in front of the plane the far plane
    // clips out objects behind the plane. This way only what is drawn
    // between the planes is actually rendered.
    renWin.GetRenderer().ResetCameraClippingRange();

    // Setup panel
    setLayout(new BorderLayout());
    add(renWin, BorderLayout.CENTER);
  }



  public vtkPanel getRenWin() {
    return renWin;
  }


  public static void main(String s[]) {
    if (s.length == 0){
      return; //need a filename here
    }
    File theFile = new File(s[0]);
    //File theFile = new
      File("/Users/mmroden/Documents/MVSDownloadDirectory/Documents/1.2.840.113704.1.111.3384.1271766367.5/");
    AWTMedical3 panel = new AWTMedical3(theFile);

    JFrame frame = new JFrame("AWTMedical3");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add("Center", panel);
    frame.pack();
    frame.setVisible(true);
  }

}
```

14.168 HelloVTKWorld.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
```

```
     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// We are required to call the package 'vtk' even though I (MM) would have preferred
// an import statement along the line of:
// import vtkgdcm.*;
import vtk.*;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac HelloVTKWorld.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
      CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java HelloVTKWorld gdcmData/012345.002.050.dcm
      bla.dcm
 *
 */
public class HelloVTKWorld
{
  static {
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkgdcmJava");
    try {
      System.loadLibrary("vtkRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkHybridJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkVolumeRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkVolumeRendering, skipping...");
    }
  }

  public static void main(String[] args)
    {
    String filename = args[0];
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileName( filename );
    reader.Update();

    vtkMedicalImageProperties prop = reader.GetMedicalImageProperties();
    System.out.println( prop.GetPatientName() ); //

//    if( reader.GetImageFormat() == vtkgdcm.vtkgdcm.VTK_LUMINANCE ) // MONOCHROME2
//      {
//      System.out.println( "Image is MONOCHROME2" ); //
//      }

    // Just for fun, invert the direction cosines, output should reflect that:
    vtkMatrix4x4 dircos = reader.GetDirectionCosines();
    dircos.Invert();

    // We need to maintain in sync information stored in vtkMedicalImageProperties:
    double[] cosines = new double[6];
    cosines[0] = dircos.GetElement(0,0);
    cosines[1] = dircos.GetElement(1,0);
    cosines[2] = dircos.GetElement(2,0);
    cosines[3] = dircos.GetElement(0,1);
    cosines[4] = dircos.GetElement(1,1);
    cosines[5] = dircos.GetElement(2,1);
    reader.GetMedicalImageProperties().SetDirectionCosine( cosines );

    String outfilename = args[1];
    vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
    writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
    writer.SetDirectionCosines( dircos );
    writer.SetShift( reader.GetShift() );
```

```
    writer.SetScale( reader.GetScale() );
    writer.SetImageFormat( reader.GetImageFormat() );
    writer.SetFileName( outfilename );
    writer.SetInputConnection( reader.GetOutputPort() ); // new
    //writer.SetInput( reader.GetOutput() ); // old
    writer.Write();

    System.out.println("Success reading: " + filename );
    }
}
```

14.169 MIPViewer.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;
import java.awt.Canvas;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MIPViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
      CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MIPViewer BRAINX
 *
 */
public class MIPViewer extends Canvas
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    System.loadLibrary("vtkVolumeRenderingJava"); // vtkSmartVolumeMapper
    System.loadLibrary("vtkWidgetsJava"); // vtkBoxWidget
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }

  static FilenamesType fns = new FilenamesType();

  protected native int Lock();

  protected native int UnLock();

  public static void process(String path)
    {
    fns.add( path );
    }

  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
```

```
        }
      }
    else
      {
      process(dir.getPath());
      }
    }

  public static void main(String[] args) throws Exception
    {
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
      {
      return;
      }

    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
      {
      throw new Exception("Could not scan");
      }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
      String f = sorted.get(i);
      files.InsertNextValue( f );
      }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

    // Create our volume and mapper
    vtkVolume volume = new vtkVolume();
    vtkSmartVolumeMapper mapper = new vtkSmartVolumeMapper();

    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();

    // Add a box widget if the clip option was selected
    vtkBoxWidget box = new vtkBoxWidget();
    box.SetInteractor(iren);
    box.SetPlaceFactor(1.01);
    box.SetInputConnection(change.GetOutputPort());

    //box.SetDefaultRenderer(renderer);
    box.InsideOutOn();
    box.PlaceWidget();
    //vtkBoxWidgetCallback callback = vtkBoxWidgetCallback::New();
    //callback.SetMapper(mapper);
    //box.AddObserver(vtkCommand::InteractionEvent, callback);
    //callback.Delete();
//    Lock();
//    box.EnabledOn();
//    UnLock();
    box.GetSelectedFaceProperty().SetOpacity(0.0);

    mapper.SetInputConnection( change.GetOutputPort() );

    // Create our transfer function
    vtkColorTransferFunction colorFun = new vtkColorTransferFunction();
    vtkPiecewiseFunction opacityFun = new vtkPiecewiseFunction();

    // Create the property and attach the transfer functions
    vtkVolumeProperty property = new vtkVolumeProperty();
    property.IndependentComponentsOn();
    property.SetColor( colorFun );
```

```
    property.SetScalarOpacity( opacityFun );
    property.SetInterpolationTypeToLinear();

    // connect up the volume to the property and the mapper
    volume.SetProperty( property );
    volume.SetMapper( mapper );

    vtkMedicalImageProperties medprop = reader.GetMedicalImageProperties();
    int n = medprop.GetNumberOfWindowLevelPresets();
    double opacityWindow = 4096;
    double opacityLevel = 2048;

    // Override default with value from DICOM files:
    for( int i = 0; i < n; ++i )
      {
      double wl[] = medprop.GetNthWindowLevelPreset(i);
      //System.out.println( "W/L: " + wl[0] + " " + wl[1] );
      opacityWindow = wl[0];
      opacityLevel  = wl[1];
      }

    colorFun.AddRGBSegment(0.0, 1.0, 1.0, 1.0, 255.0, 1.0, 1.0, 1.0 );
    opacityFun.AddSegment( opacityLevel - 0.5*opacityWindow, 0.0,
      opacityLevel + 0.5*opacityWindow, 1.0 );
    mapper.SetBlendModeToMaximumIntensity();

    // Create the RenderWindow, Renderer
    vtkRenderer ren1 = new vtkRenderer();
    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);

    // Set the default window size
    renWin.SetSize(600,600);

    // Add the volume to the scene
    ren1.AddVolume( volume );
    ren1.ResetCamera();

    iren.SetRenderWindow( renWin );

    // interact with data
    renWin.Render();

    iren.Start();
    }
}
```

14.170 MPRViewer.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
      CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MPRViewer BRAINX
 *
 */
public class MPRViewer
```

```
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }

  static FilenamesType fns = new FilenamesType();

  public static void process(String path)
    {
    fns.add( path );
    }

  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }

  public static void main(String[] args) throws Exception
    {
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
      {
      return;
      }

    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
      {
      throw new Exception("Could not scan");
      }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
      String f = sorted.get(i);
      files.InsertNextValue( f );
      }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

    // A simple vtkInteractorStyleImage example for
    // 3D image viewing with the vtkImageResliceMapper.
    //
    // Drag Left mouse button to window/level
    // Shift-Left drag to rotate (oblique slice)
```

```
    // Shift-Middle drag to slice through image
    // OR Ctrl-Right drag to slice through image

    // Create the RenderWindow, Renderer
    vtkRenderer ren1 = new vtkRenderer();
    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);

    vtkImageResliceMapper im = new vtkImageResliceMapper();
    im.SetInputConnection(change.GetOutputPort());
    im.SliceFacesCameraOn();
    im.SliceAtFocalPointOn();
    im.BorderOff();

    vtkImageProperty ip = new vtkImageProperty();
    ip.SetColorWindow(2000);
    ip.SetColorLevel(1000);
    ip.SetAmbient(0.0);
    ip.SetDiffuse(1.0);
    ip.SetOpacity(1.0);
    ip.SetInterpolationTypeToLinear();

    vtkImageSlice ia = new vtkImageSlice();
    ia.SetMapper(im);
    ia.SetProperty(ip);

    ren1.AddViewProp(ia);
    ren1.SetBackground(0.1,0.2,0.4);
    renWin.SetSize(300,300);

    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
    vtkInteractorStyleImage style = new vtkInteractorStyleImage();
    style.SetInteractionModeToImage3D();
    iren.SetInteractorStyle(style);
    renWin.SetInteractor(iren);

    // render the image
    renWin.Render();
    vtkCamera cam1 = ren1.GetActiveCamera();
    cam1.ParallelProjectionOn();
    ren1.ResetCameraClippingRange();
    renWin.Render();

    iren.Start();
    }
}
```

14.171 MPRViewer2.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer2.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:.
      CLASSPATH=/usr/share/java/vtk.jar:vtkgdcm.jar:gdcm.jar:. java MPRViewer2 BRAINX
 *
 */
public class MPRViewer2
```

```
{
  static {
    // VTK
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    System.loadLibrary("vtkHybridJava");
    System.loadLibrary("vtkWidgetsJava");
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
  }

  static FilenamesType fns = new FilenamesType();

  public static void process(String path)
    {
    fns.add( path );
    }

  // Process only files under dir
  public static void visitAllFiles(File dir)
    {
    if (dir.isDirectory())
      {
      String[] children = dir.list();
      for (int i=0; i<children.length; i++)
        {
        visitAllFiles(new File(dir, children[i]));
        }
      }
    else
      {
      process(dir.getPath());
      }
    }

  public void dointer(vtkImagePlaneWidget  current_widget)
    {
    int cstat = current_widget.GetCursorDataStatus();
    double[] v = current_widget.GetCurrentCursorPosition();
    //System.out.println( cstat );
    //System.out.println( v[0] );
    //System.out.println( v[1] );
    //System.out.println( v[2] );
    planeWidgetX.SetSliceIndex( (int)v[0] );
    planeWidgetY.SetSliceIndex( (int)v[1] );
    planeWidgetZ.SetSliceIndex( (int)v[2] );
    planeWidgetX.GetCurrentRenderer().ResetCameraClippingRange();
    planeWidgetY.GetCurrentRenderer().ResetCameraClippingRange();
    planeWidgetZ.GetCurrentRenderer().ResetCameraClippingRange();
    }
  public void startinterX()
    {
    dointer( planeWidgetX );
    }
  public void interX()
    {
    dointer( planeWidgetX );
    }
  public void endinterX()
    {
    }
  public void startinterY()
    {
    dointer( planeWidgetY );
    }
  public void interY()
    {
    dointer( planeWidgetY );
    }
  public void endinterY()
    {
    }
  public void startinterZ()
    {
    dointer( planeWidgetZ );
    }
  public void interZ()
```

```
    {
    dointer( planeWidgetZ );
    }
  public void endinterZ()
    {
    //System.out.println( "endinter" );
    }

  public static void AlignCamera(int slice_number, vtkImagePlaneWidget current_widget)
    {
    vtkImageData image = (vtkImageData)current_widget.GetInput();
    vtkRenderer ren = current_widget.GetCurrentRenderer();
    double[] origin = image.GetOrigin();
    double ox = origin[0];
    double oy = origin[1];
    double oz = origin[2];

    int dims[] = image.GetDimensions();
    int xMin = 0;
    int xMax = 1;
    int yMin = 2;
    int yMax = dims[0]-1;
    int zMin = dims[1]-1;
    int zMax = dims[2]-1;

    double[] spacing = image.GetSpacing();
    double sx = spacing[0];
    double sy = spacing[1];
    double sz = spacing[2];

    double cx = ox+(0.5*(xMax-xMin))*sx;
    double cy = oy+(0.5*(yMax-yMin))*sy;
    double cz = oy+(0.5*(zMax-zMin))*sz;
    double vx = 0, vy = 0, vz = 0;
    double nx = 0, ny = 0, nz = 0;
    int iaxis = current_widget.GetPlaneOrientation();
    if ( iaxis == 0 ) {
      vz = -1;
      nx = ox + xMax*sx;
      cx = ox + slice_number*sx;
      }
    else if ( iaxis == 1 ) {
      vz = -1;
      ny = oy+yMax*sy;
      cy = oy+slice_number*sy;
      }
    else {
      vy = 1;
      nz = oz+zMax*sz;
      cz = oz+slice_number*sz;
    }
    double px = cx+nx*2;
    double py = cy+ny*2;
    double pz = cz+nz*3;

    vtkCamera camera = ren.GetActiveCamera();
    camera.SetViewUp(vx, vy, vz);
    camera.SetFocalPoint(cx, cy, cz);
    camera.SetPosition(px, py, pz);
    camera.OrthogonalizeViewUp();
    ren.ResetCameraClippingRange();
    }

  private vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
  private vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
  private vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();

  public void config()
    {
    //System.out.println( "config" );
    planeWidgetX.GetCurrentRenderer().ResetCamera();
    planeWidgetY.GetCurrentRenderer().ResetCamera();
    planeWidgetZ.GetCurrentRenderer().ResetCamera();
    }

  public void Run(String dirname)
    {
    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
```

```
ipp.SetComputeZSpacing( true );
ipp.SetZSpacingTolerance( 1e-3 );
boolean b = ipp.Sort( fns );
if(!b)
  {
  //throw new Exception("Could not scan");
  }
double ippzspacing = ipp.GetZSpacing();

FilenamesType sorted = ipp.GetFilenames();
vtkStringArray files = new vtkStringArray();
long nfiles = sorted.size();
//for( String f : sorted )
for (int i = 0; i < nfiles; i++) {
  String f = sorted.get(i);
  files.InsertNextValue( f );
  }
vtkGDCMImageReader reader = new vtkGDCMImageReader();
reader.SetFileNames( files );
reader.Update(); // get spacing value

double[] spacing = reader.GetOutput().GetSpacing();

vtkImageChangeInformation change = new vtkImageChangeInformation();
change.SetInputConnection( reader.GetOutputPort() );
change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
change.Update();

System.out.println( change.GetOutput().toString() );

vtkRenderer ren1 = new vtkRenderer();
ren1.SetViewport(0., 0., 0.333, 1);
ren1.SetBackground(0.1,0.2,0.4);
vtkRenderer ren2 = new vtkRenderer();
ren2.SetViewport(0.333, 0., 0.667, 1);
ren2.SetBackground(0.1,0.2,0.4);
vtkRenderer ren3 = new vtkRenderer();
ren3.SetViewport(0.667, 0., 1., 1.);
ren3.SetBackground(0.1,0.2,0.4);

vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);
renWin.AddRenderer(ren2);
renWin.AddRenderer(ren3);

vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
iren.SetRenderWindow(renWin);

vtkInteractorStyleImage style = new vtkInteractorStyleImage();
iren.SetInteractorStyle( style );

vtkCellPicker picker = new vtkCellPicker();
picker.SetTolerance(0.005);

vtkProperty ipwProp = new vtkProperty();

//vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
planeWidgetX.SetInteractor(iren);
planeWidgetX.SetCurrentRenderer(ren1);
planeWidgetX.SetDefaultRenderer(ren1);
planeWidgetX.RestrictPlaneToVolumeOn();
planeWidgetX.SetTexturePlaneProperty(ipwProp);
//planeWidgetX.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetX.TextureInterpolateOff();
//planeWidgetX.SetResliceInterpolateToNearestNeighbour();
planeWidgetX.SetInputConnection(change.GetOutputPort());
planeWidgetX.SetPlaneOrientationToXAxes();
planeWidgetX.SetSliceIndex(62);
planeWidgetX.SetPicker(picker);
planeWidgetX.SetKeyPressActivationValue('x');
planeWidgetX.On();
planeWidgetX.InteractionOn();

//vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
planeWidgetY.SetInteractor(iren);
planeWidgetY.SetCurrentRenderer(ren2);
planeWidgetY.SetDefaultRenderer(ren2);
planeWidgetY.RestrictPlaneToVolumeOn();
planeWidgetY.SetTexturePlaneProperty(ipwProp);
//planeWidgetY.GetPlaneProperty().SetColor(1,0,0);
//planeWidgetY.TextureInterpolateOff();
```

```
    //planeWidgetY.SetResliceInterpolateToNearestNeighbour();
    planeWidgetY.SetInputConnection(change.GetOutputPort());
    planeWidgetY.SetLookupTable( planeWidgetX.GetLookupTable() );
    planeWidgetY.SetPlaneOrientationToYAxes();
    planeWidgetY.SetSliceIndex(32);
    planeWidgetY.SetPicker(picker);
    planeWidgetY.SetKeyPressActivationValue('y');
    planeWidgetY.On();


    //vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();
    planeWidgetZ.SetInteractor(iren);
    planeWidgetZ.SetCurrentRenderer(ren3);
    planeWidgetZ.SetDefaultRenderer(ren3);
    planeWidgetZ.RestrictPlaneToVolumeOn();
    planeWidgetZ.SetTexturePlaneProperty(ipwProp);
    //planeWidgetZ.GetPlaneProperty().SetColor(1,0,0);
    //planeWidgetZ.TextureInterpolateOff();
    //planeWidgetZ.SetResliceInterpolateToNearestNeighbour();
    planeWidgetZ.SetInputConnection(change.GetOutputPort());
    planeWidgetZ.SetLookupTable( planeWidgetX.GetLookupTable() );
    planeWidgetZ.SetPlaneOrientationToZAxes();
    planeWidgetZ.SetSliceIndex(32);
    planeWidgetZ.SetPicker(picker);
    planeWidgetZ.SetKeyPressActivationValue('z');
    planeWidgetZ.On();

    iren.Initialize();

    renWin.Render();
    AlignCamera(52, planeWidgetX);
    AlignCamera(32, planeWidgetY);
    AlignCamera(32, planeWidgetZ);

    planeWidgetX.GetCurrentRenderer().ResetCamera();
    planeWidgetY.GetCurrentRenderer().ResetCamera();
    planeWidgetZ.GetCurrentRenderer().ResetCamera();

    renWin.Render();

    planeWidgetX.AddObserver("StartInteractionEvent", this,"startinterX");
    planeWidgetX.AddObserver("InteractionEvent", this,"interX");
    planeWidgetX.AddObserver("EndInteractionEvent", this,"endinterX");
    planeWidgetY.AddObserver("StartInteractionEvent", this,"startinterY");
    planeWidgetY.AddObserver("InteractionEvent", this,"interY");
    planeWidgetY.AddObserver("EndInteractionEvent", this,"endinterY");
    planeWidgetZ.AddObserver("StartInteractionEvent", this,"startinterZ");
    planeWidgetZ.AddObserver("InteractionEvent", this,"interZ");
    planeWidgetZ.AddObserver("EndInteractionEvent", this,"endinterZ");

    iren.AddObserver("ConfigureEvent", this,"config");

    iren.Start();
    }

  public static void main(String[] args) throws Exception
    {
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
      {
      return;
      }

    MPRViewer2 me = new MPRViewer2();
    me.Run( dirname );
    }
}
```

14.172 ReadSeriesIntoVTK.java

```
/*=========================================================================

  Program: GDCM (Grassroots DICOM). A DICOM library

  Copyright (c) 2006-2011 Mathieu Malaterre
  All rights reserved.
```

```
  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

     This software is distributed WITHOUT ANY WARRANTY; without even
     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
     PURPOSE.  See the above copyright notice for more information.

=========================================================================*/
// We are required to call the package 'vtk' even though I (MM) would have preferred
// an import statement along the line of:
// import vtkgdcm.*;
import vtk.*;

/*
 * Usage:
 * export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
 * java -classpath `pwd`/vtkgdcm.jar:/usr/share/java/vtk.jar:. ReadSeriesIntoVTK
 */
public class ReadSeriesIntoVTK
{
  static {
    System.loadLibrary("vtkCommonJava");
    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkgdcmJava");
    try {
      System.loadLibrary("vtkRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkHybridJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkHybrid, skipping...");
    }
    try {
      System.loadLibrary("vtkVolumeRenderingJava");
    } catch (Throwable e) {
      System.out.println("cannot load vtkVolumeRendering, skipping...");
    }
  }

  public static void main(String[] args)
    {
    vtkFileOutputWindow outWin = new vtkFileOutputWindow();
    outWin.SetInstance(outWin);
    outWin.SetFileName("MVSVTKViewer.log");

    // See: http://review.source.kitware.com/#change,888
    // vtkWrapJava does not handle static keyword
    // String directory = vtkGDCMTesting.GetGDCMDataRoot();
    vtkGDCMTesting t = new vtkGDCMTesting();
    String directory = t.GetGDCMDataRoot();
    String file0 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm";
    String file1 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm";
    String file2 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm";
    String file3 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm";

    vtkStringArray s = new vtkStringArray();
    System.out.println("adding : " + file0 );
    s.InsertNextValue( file0 );
    s.InsertNextValue( file1 );
    s.InsertNextValue( file2 );
    s.InsertNextValue( file3 );

    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( s );
    reader.Update();

    System.out.println("Success reading: " + file0 );

    vtkMetaImageWriter writer = new vtkMetaImageWriter();
    writer.DebugOn();
    writer.SetCompression( false );
    writer.SetInputConnection( reader.GetOutputPort() );
    writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
    writer.Write();

    System.out.println("Success writing: " + writer.GetFileName() );
    }
```

}

14.173 CastConvertPhilips.py

```

"""
Usage:

 python --public /path/to/directory/
or
 python --private /path/to/directory/

 python --public --extension bak /path/to/directory/

rename -f 's/\.bak$//' *.bak

TODO:
http://docs.python.org/library/optparse.html#module-optparse
"""

import vtkgdcm
import vtk
import sys
import gdcm

def ProcessOneFilePublic(filename, outfilename, tmpfile):
  gdcm.ImageHelper.SetForceRescaleInterceptSlope(True)
  vtkreader = vtkgdcm.vtkGDCMImageReader()
  vtkreader.SetFileName( filename )
  vtkreader.Update()

  cast = vtk.vtkImageCast()
  cast.SetInput( vtkreader.GetOutput() )
  cast.SetOutputScalarTypeToUnsignedShort()

  # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
  # Some operation will actually be discarded (we simply need a temp storage)
  vtkwriter = vtkgdcm.vtkGDCMImageWriter()
  vtkwriter.SetFileName( tmpfile )
  vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
  vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
  print "Format:",vtkreader.GetImageFormat()
  vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
  vtkwriter.SetInput( cast.GetOutput() )
  #vtkwriter.Update()
  vtkwriter.Write()

  # ok now rewrite the exact same file as the original (keep all info)
  # but use the Pixel Data Element from the written file
  tmpreader = gdcm.ImageReader()
  tmpreader.SetFileName( tmpfile )
  if not tmpreader.Read():
    sys.exit(1)

  reader = gdcm.Reader()
  reader.SetFileName( filename )
  if not reader.Read():
    sys.exit(1)

  # Make sure to remove Slope/Rescale to avoid re-execution
  ds = reader.GetFile().GetDataSet()
  tags = [
    gdcm.Tag(0x0028,0x1052),
    gdcm.Tag(0x0028,0x1053),
    gdcm.Tag(0x0028,0x1053),
  ]
  for tag in tags:
    ds.Remove( tag )

  writer = gdcm.ImageWriter()
  writer.SetFileName( outfilename )
  # Pass image from vtk written file
  writer.SetImage( tmpreader.GetImage() )
  # pass dataset from initial 'reader'
  writer.SetFile( reader.GetFile() )
```

```
  if not writer.Write():
    sys.exit(1)

def ProcessOneFilePrivate(filename, outfilename, tmpfile):
  vtkreader = vtkgdcm.vtkGDCMImageReader()
  vtkreader.SetFileName( filename )
  vtkreader.Update()


  # (2005,1409)     DS      4       0.0
  # (2005,140a)     DS      16      1.52283272283272

  # (2005,0014)     LO      26      Philips MR Imaging DD 005
  tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
  tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")



  # Need to access some private tags, reread the file (for now):
  reader = gdcm.Reader()
  reader.SetFileName( filename )
  if not reader.Read():
    sys.exit(1)

  ds = reader.GetFile().GetDataSet()

  el1 = ds.GetDataElement( tag1 )
  el2 = ds.GetDataElement( tag2 )


  #pf = gdcm.PythonFilter()
  #pf.SetFile( reader.GetFile() )
  #print el1.GetTag()

  print el1.GetByteValue()
  v1 = eval(el1.GetByteValue().GetBuffer())
  print el2.GetByteValue()
  v2 = eval(el2.GetByteValue().GetBuffer())

  print v1
  shift = v1
  print v2
  scale = v2

  ss = vtk.vtkImageShiftScale()
  ss.SetInput( vtkreader.GetOutput() )
  # because VTK image shift / scale convention is inverted from DICOM make sure shift is 0
  assert shift == 0
  ss.SetShift( shift )
  ss.SetScale( scale )
  ss.SetOutputScalarTypeToUnsignedShort ()
  ss.Update()

  # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
  # Some operation will actually be discarded (we simply need a temp storage)
  vtkwriter = vtkgdcm.vtkGDCMImageWriter()
  vtkwriter.SetFileName( tmpfile )
  vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
  vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
  vtkwriter.SetImageFormat( reader.GetImageFormat() )
  # do not pass shift/scale again
  vtkwriter.SetInput( ss.GetOutput() )
  #vtkwriter.Update()
  vtkwriter.Write()

  # ok now rewrite the exact same file as the original (keep all info)
  # but use the Pixel Data Element from the written file
  tmpreader = gdcm.ImageReader()
  tmpreader.SetFileName( tmpfile )
  if not tmpreader.Read():
    sys.exit(1)

  writer = gdcm.ImageWriter()
  writer.SetFileName( outfilename )
  # Pass image from vtk written file
  writer.SetImage( tmpreader.GetImage() )
  # pass dataset from initial 'reader'
  writer.SetFile( reader.GetFile() )
  if not writer.Write():
    sys.exit(1)

```

```
if __name__ == "__main__":

  gdcm.Trace.DebugOff()
  gdcm.Trace.WarningOff()
  #filename = sys.argv[1]
  #outfilename = sys.argv[2]
  tmpfile = "/tmp/philips_rescaled.dcm"
  #ProcessOneFile( filename, outfilename, tmpfile )
  rescaletype = sys.argv[1]
  assert rescaletype == "--public" or rescaletype == "--private"
  dirname = sys.argv[2]
  d = gdcm.Directory()
  d.Load( dirname )

  for f in d.GetFilenames():
    #print f
    ProcessOneFilePublic( f, f + ".bak", tmpfile )


print "success"
```

14.174 headsq2dcm.py

```


"""
Usage:
 python headsq2dcm.py -D /path/to/VTKData
"""

import vtk
import vtkgdcm
from vtk.util.misc import vtkGetDataRoot
VTK_DATA_ROOT = vtkGetDataRoot()

reader = vtk.vtkVolume16Reader()
reader.SetDataDimensions(64, 64)
reader.SetDataByteOrderToLittleEndian()
reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter")
reader.SetImageRange(1, 93)
reader.SetDataSpacing(3.2, 3.2, 1.5)

cast = vtk.vtkImageCast()
cast.SetInput( reader.GetOutput() )
cast.SetOutputScalarTypeToUnsignedChar()

# By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
writer = vtkgdcm.vtkGDCMImageWriter()
writer.SetFileName( "headsq.dcm" )
writer.SetInput( reader.GetOutput() )
# cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
#writer.SetInput( cast.GetOutput() )
writer.SetFileDimensionality( 3 )
writer.Write()
```

Index

~ASN1
gdcm::ASN1, 156
~AnonymizeEvent
gdcm::AnonymizeEvent, 133
~Anonymizer
gdcm::Anonymizer, 138
~Attribute
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 190
~AudioCodec
gdcm::AudioCodec, 221
~BaseCompositeMessage
gdcm::network::BaseCompositeMessage, 227
~BaseNormalizedMessage
gdcm::network::BaseNormalizedMessage, 229
~BasePDU
gdcm::network::BasePDU, 231
~BaseQuery
gdcm::BaseQuery, 234
~BaseRootQuery
gdcm::BaseRootQuery, 239
~Bitmap
gdcm::Bitmap, 253
~BitmapToBitmapFilter
gdcm::BitmapToBitmapFilter, 266
~BoxRegion
gdcm::BoxRegion, 269
~ByteSwapFilter
gdcm::ByteSwapFilter, 275
~ByteValue
gdcm::ByteValue, 279
~CAPICryptographicMessageSyntax
gdcm::CAPICryptographicMessageSyntax, 288
~CSAHeader
gdcm::CSAHeader, 351
~Cleaner
gdcm::Cleaner, 301
~Coder
gdcm::Coder, 311
~Command
gdcm::Command, 318
~CommandDataSet
gdcm::CommandDataSet, 322
~CryptoFactory
gdcm::CryptoFactory, 337
~CryptographicMessageSyntax
gdcm::CryptographicMessageSyntax, 338
~Curve
gdcm::Curve, 366
~DICOMDIRGenerator
gdcm::DICOMDIRGenerator, 418
~DPath
gdcm::DPath, 452
~DataEvent
gdcm::DataEvent, 386
~DataSetEvent
gdcm::DataSetEvent, 403
~Decoder
gdcm::Decoder, 406
~Defs
gdcm::Defs, 409
~DeltaEncodingCodec
gdcm::DeltaEncodingCodec, 415
~DictConverter
gdcm::DictConverter, 425
~DictPrinter
gdcm::DictPrinter, 434
~Dicts
gdcm::Dicts, 437
~DirectionCosines
gdcm::DirectionCosines, 442
~Directory
gdcm::Directory, 446
~Dumper
gdcm::Dumper, 456
~Element
gdcm::Element< TVR, VM::VM1_n >, 470
~EmptyMaskGenerator
gdcm::EmptyMaskGenerator, 522
~Event
gdcm::Event, 534
~Exception
gdcm::Exception, 537
~File
gdcm::File, 550
~FileAnonymizer
gdcm::FileAnonymizer, 555
~FileChangeTransferSyntax
gdcm::FileChangeTransferSyntax, 559
~FileDecompressLookupTable

gdcm::FileDecompressLookupTable, 563
~FileDerivation
gdcm::FileDerivation, 565
~FileExplicitFilter
gdcm::FileExplicitFilter, 569
~FileMetaInformation
gdcm::FileMetaInformation, 575
~FileNameEvent
gdcm::FileNameEvent, 586
~FileStreamer
gdcm::FileStreamer, 596
~FilenameGenerator
gdcm::FilenameGenerator, 589
~Global
gdcm::Global, 616
~GroupDict
gdcm::GroupDict, 619
~IconImageFilter
gdcm::IconImageFilter, 622
~IconImageGenerator
gdcm::IconImageGenerator, 625
~Image
gdcm::Image, 633
~ImageApplyLookupTable
gdcm::ImageApplyLookupTable, 639
~ImageChangePhotometricInterpretation
gdcm::ImageChangePhotometricInterpretation, 643
~ImageChangePlanarConfiguration
gdcm::ImageChangePlanarConfiguration, 648
~ImageChangeTransferSyntax
gdcm::ImageChangeTransferSyntax, 652
~ImageCodec
gdcm::ImageCodec, 658
~ImageConverter
gdcm::ImageConverter, 668
~ImageFragmentSplitter
gdcm::ImageFragmentSplitter, 671
~ImageReader
gdcm::ImageReader, 682
~ImageRegionReader
gdcm::ImageRegionReader, 687
~ImageToImageFilter
gdcm::ImageToImageFilter, 691
~ImageWriter
gdcm::ImageWriter, 695
~JPEG12Codec
gdcm::JPEG12Codec, 731
~JPEG16Codec
gdcm::JPEG16Codec, 736
~JPEG2000Codec
gdcm::JPEG2000Codec, 740
~JPEG8Codec
gdcm::JPEG8Codec, 749
~JPEGCodec
gdcm::JPEGCodec, 754
~JPEGLSCodec
gdcm::JPEGLSCodec, 762
~JSON
gdcm::JSON, 767
~KAKADUCodec
gdcm::KAKADUCodec, 771
~LookupTable
gdcm::LookupTable, 778
~MemberCommand
gdcm::MemberCommand< T >, 808
~MeshPrimitive
gdcm::MeshPrimitive, 814
~ModuleEntry
gdcm::ModuleEntry, 830
~MrProtocol
gdcm::MrProtocol, 843
~Object
gdcm::Object, 872
~OpenSSLCryptographicMessageSyntax
gdcm::OpenSSLCryptographicMessageSyntax, 878
~OpenSSLP7CryptographicMessageSyntax
gdcm::OpenSSLP7CryptographicMessageSyntax, 883
~Orientation
gdcm::Orientation, 886
~Overlay
gdcm::Overlay, 891
~PDBHeader
gdcm::PDBHeader, 912
~PDFCodec
gdcm::PDFCodec, 915
~PGXCodec
gdcm::PGXCodec, 925
~PNMCodec
gdcm::PNMCodec, 961
~PVRGCodec
gdcm::PVRGCodec, 1002
~ParseException
gdcm::ParseException, 899
~Parser
gdcm::Parser, 902
~Pixmap
gdcm::Pixmap, 943
~PixmapReader
gdcm::PixmapReader, 949
~PixmapToPixmapFilter
gdcm::PixmapToPixmapFilter, 952
~PixmapWriter
gdcm::PixmapWriter, 956
~Preamble
gdcm::Preamble, 964
~Printer
gdcm::Printer, 985

~PrivateDict
gdcm::PrivateDict, 988
~ProgressEvent
gdcm::ProgressEvent, 997
~PythonFilter
gdcm::PythonFilter, 1004
~QueryBase
gdcm::QueryBase, 1006
~RAWCodec
gdcm::RAWCodec, 1022
~RLECodec
gdcm::RLECodec, 1042
~Reader
gdcm::Reader, 1027
~Region
gdcm::Region, 1033
~Rescaler
gdcm::Rescaler, 1036
~SHA1
gdcm::SHA1, 1123
~Scanner
gdcm::Scanner, 1052
~Scanner2
gdcm::Scanner2, 1062
~Segment
gdcm::Segment, 1071
~SegmentReader
gdcm::SegmentReader, 1083
~SegmentWriter
gdcm::SegmentWriter, 1088
~SegmentedPaletteColorLookupTable
gdcm::SegmentedPaletteColorLookupTable, 1079
~SerieHelper
gdcm::SerieHelper, 1109
~ServiceClassUser
gdcm::ServiceClassUser, 1117
~SimpleMemberCommand
gdcm::SimpleMemberCommand< T >, 1127
~SimpleSubjectWatcher
gdcm::SimpleSubjectWatcher, 1130
~SmartPointer
gdcm::SmartPointer< ObjectType >, 1136
~Sorter
gdcm::Sorter, 1143
~Spacing
gdcm::Spacing, 1147
~SplitMosaicFilter
gdcm::SplitMosaicFilter, 1150
~StreamImageReader
gdcm::StreamImageReader, 1156
~StreamImageWriter
gdcm::StreamImageWriter, 1161
~StrictScanner
gdcm::StrictScanner, 1170
~StrictScanner2
gdcm::StrictScanner2, 1180
~StringFilter
gdcm::StringFilter, 1191
~Subject
gdcm::Subject, 1196
~Surface
gdcm::Surface, 1202
~SurfaceReader
gdcm::SurfaceReader, 1218
~SurfaceWriter
gdcm::SurfaceWriter, 1223
~Table
gdcm::Table, 1237
~TableEntry
gdcm::TableEntry, 1239
~TableReader
gdcm::TableReader, 1240
~TableRow
gdcm::network::TableRow, 1244
~TagPath
gdcm::TagPath, 1255
~Testing
gdcm::Testing, 1258
~Trace
gdcm::Trace, 1265
~Transition
gdcm::network::Transition, 1277
~ULAction
gdcm::network::ULAction, 1304
~ULBasicCallback
gdcm::network::ULBasicCallback, 1345
~ULConnection
gdcm::network::ULConnection, 1347
~ULConnectionCallback
gdcm::network::ULConnectionCallback, 1352
~ULConnectionManager
gdcm::network::ULConnectionManager, 1358
~ULEvent
gdcm::network::ULEvent, 1363
~ULWritingCallback
gdcm::network::ULWritingCallback, 1367
~UserInformation
gdcm::network::UserInformation, 1382
~Validate
gdcm::Validate, 1385
~Value
gdcm::Value, 1388
~Version
gdcm::Version, 1392
~Writer
gdcm::Writer, 1536
~XMLDictReader
gdcm::XMLDictReader, 1540

~XMLPrinter
gdcm::XMLPrinter, 1543
~XMLPrivateDictReader
gdcm::XMLPrivateDictReader, 1547
~vtkGDCMImageReader
vtkGDCMImageReader, 1421
~vtkGDCMImageReader2
vtkGDCMImageReader2, 1436
~vtkGDCMImageWriter
vtkGDCMImageWriter, 1450
~vtkGDCMMedicalImageProperties
vtkGDCMMedicalImageProperties, 1457
~vtkGDCMPolyDataReader
vtkGDCMPolyDataReader, 1461
~vtkGDCMPolyDataWriter
vtkGDCMPolyDataWriter, 1466
~vtkGDCMTesting
vtkGDCMTesting, 1470
~vtkGDCMThreadedImageReader
vtkGDCMThreadedImageReader, 1475
~vtkGDCMThreadedImageReader2
vtkGDCMThreadedImageReader2, 1479
~vtkImageColorViewer
vtkImageColorViewer, 1488
~vtkImageMapToColors16
vtkImageMapToColors16, 1499
~vtkImageMapToWindowLevelColors2
vtkImageMapToWindowLevelColors2, 1505
~vtkImagePlanarComponentsToComponents
vtkImagePlanarComponentsToComponents, 1508
~vtkImageRGBToYBR
vtkImageRGBToYBR, 1511
~vtkImageYBRToRGB
vtkImageYBRToRGB, 1513
~vtkLookupTable16
vtkLookupTable16, 1515
~vtkRTStructSetProperties
vtkRTStructSetProperties, 1520

AAbortPDU
gdcm::network::AAbortPDU, 114
AAssociateACPDU
gdcm::network::AAssociateACPDU, 117
gdcm::network::AAssociateRQPDU, 127
AAssociateRJPDU
gdcm::network::AAssociateRJPDU, 121
AAssociateRQPDU
gdcm::network::AAssociateACPDU, 119
gdcm::network::AAssociateRQPDU, 124
AbstractSyntax
gdcm::network::AbstractSyntax, 129
gdcm::PresentationContext, 970
ActiveComponent
vtkImageMapToColors16, 1502
Add
gdcm::GroupDict, 620
add1
gdcm, 91
AddAcceptedPresentationContext
gdcm::network::ULConnection, 1347
AddContourReferencedFrameOfReference
vtkRTStructSetProperties, 1520
AddCSAHeaderDictEntry
gdcm::CSAHeaderDict, 356
AddDerivationDescription
gdcm::FileDerivation, 566
AddDictEntry
gdcm::Dict, 422
gdcm::PrivateDict, 988
AddFile
gdcm::FileSet, 592
gdcm::SerieHelper, 1109
AddFileName
gdcm::SerieHelper, 1109
AddFragment
gdcm::SequenceOfFragments, 1093
AddFromFile
gdcm::PresentationContextGenerator, 974
AddGroupLength
gdcm::DictConverter, 426
AddImageDirectoryRecord
gdcm::DICOMDIRGenerator, 418
AddInput
vtkImageColorViewer, 1488
AddInputConnection
vtkImageColorViewer, 1488
AddIOD
gdcm::IODs, 713
AddIODEntry
gdcm::IOD, 707
AddItem
gdcm::SequenceOfItems, 1101
AddMacro
gdcm::Macros, 789
gdcm::Module, 826
AddMacroEntry
gdcm::Macro, 787
AddModule
gdcm::Modules, 834
AddModuleEntry
gdcm::Module, 826
gdcm::NestedModuleEntries, 856
AddNewUndefinedLengthItem
gdcm::SequenceOfItems, 1101
AddObserver
gdcm::Subject, 1197
AddPatientDirectoryRecord
gdcm::DICOMDIRGenerator, 418

AddPresentationContext
gdcm::network::AAssociateRQPDU, 124
gdcm::PresentationContextGenerator, 974
AddPresentationContextAC
gdcm::network::AAssociateACPDU, 118
AddPresentationDataValue
gdcm::network::PDataTFPDU, 906
AddPrimitiveData
gdcm::MeshPrimitive, 814
AddPrivateTag
gdcm::Scanner, 1052
gdcm::Scanner2, 1062
gdcm::StrictScanner, 1170
gdcm::StrictScanner2, 1180
AddPublicTag
gdcm::Scanner2, 1062
gdcm::StrictScanner2, 1180
AddPurposeOfReferenceCodeSequence
gdcm::FileDerivation, 566
AddQueryDataSet
gdcm::BaseQuery, 234
AddReference
gdcm::FileDerivation, 566
AddReferencedFrameOfReference
vtkRTStructSetProperties, 1520
AddRestriction
gdcm::SerieHelper, 1109
AddRoleSelectionSub
gdcm::network::UserInformation, 1382
AddSegment
gdcm::SegmentWriter, 1088
AddSelect
gdcm::Sorter, 1143
AddSeriesDirectoryRecord
gdcm::DICOMDIRGenerator, 418
AddSkipTag
gdcm::Scanner, 1052
gdcm::Scanner2, 1062
gdcm::StrictScanner, 1170
gdcm::StrictScanner2, 1180
AddSOPClassExtendedNegociationSub
gdcm::network::UserInformation, 1382
AddSourceImageSequence
gdcm::FileDerivation, 566
AddStructureSetROI
vtkRTStructSetProperties, 1520
AddStructureSetROIObservation
vtkRTStructSetProperties, 1521
AddStudyDirectoryRecord
gdcm::DICOMDIRGenerator, 418
AddSurface
gdcm::Segment, 1071
AddTag
gdcm::Scanner, 1052
gdcm::StrictScanner, 1171
AddTransferSyntax
gdcm::network::PresentationContextRQ, 977
gdcm::PresentationContext, 969
AE
gdcm::VR, 1405
AEComp
gdcm, 86
AES128_CIPHER
gdcm::CryptographicMessageSyntax, 338
AES192_CIPHER
gdcm::CryptographicMessageSyntax, 338
AES256_CIPHER
gdcm::CryptographicMessageSyntax, 338
AffectedSOPClassUID
gdcm::network::CEchoRQ, 292
AGFA
gdcm::EquipmentManufacturer, 532
ALGOType
gdcm::Segment, 1071
ALGOType_END
gdcm::Segment, 1071
Allocate
gdcm::LookupTable, 779
AmbulatoryECGWaveformStorage
gdcm::MediaStorage, 798
AnatomicRegion
gdcm::Segment, 1075
AnatomicRegionModifiers
gdcm::Segment, 1075
AnonymizeEvent
gdcm::AnonymizeEvent, 133
Anonymizer
gdcm::Anonymizer, 138
Append
gdcm::ByteValue, 280
gdcm::Global, 616
AppendFrameEncode
gdcm::ImageCodec, 658
gdcm::JPEG2000Codec, 740
gdcm::JPEGCodec, 754
gdcm::JPEGLSCodec, 763
gdcm::RLECodec, 1043
AppendImplementationClassUID
gdcm::FileMetaInformation, 575
AppendRowEncode
gdcm::ImageCodec, 658
gdcm::JPEG2000Codec, 740
gdcm::JPEGCodec, 754
gdcm::JPEGLSCodec, 763
gdcm::RLECodec, 1043
AppendToDataElement
gdcm::FileStreamer, 596
AppendToGroupDataElement

gdcm::FileStreamer, 596
ApplicationContext
gdcm::network::ApplicationContext, 146
Apply
gdcm::ImageApplyLookupTable, 640
ApplyInverseVideo
vtkGDCMImageReader, 1430
vtkGDCMImageReader2, 1445
ApplyLookupTable
vtkGDCMImageReader, 1430
vtkGDCMImageReader2, 1445
ApplyPlanarConfiguration
vtkGDCMImageReader, 1430
vtkGDCMImageReader2, 1445
ApplyShiftScale
vtkGDCMImageReader, 1430
vtkGDCMImageReader2, 1445
ApplyYBRToRGB
vtkGDCMImageReader, 1430
vtkGDCMImageReader2, 1445
Area
gdcm::BoxRegion, 269
gdcm::Region, 1033
AReleaseRPPDU
gdcm::network::AReleaseRPPDU, 150
AReleaseRQPDU
gdcm::network::AReleaseRQPDU, 153
AreOverlaysInPixelData
gdcm::Bitmap, 253
gdcm::Pixmap, 943
ARGB
gdcm::PhotometricInterpretation, 928
ArrayIncludeMacrosType
gdcm::Macro, 786
gdcm::Module, 826
ArrayType
gdcm::Attribute< Group, Element, TVR, TVM >, 160
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 170
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 178
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 183
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 189
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 198
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 204
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 210
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 216
ARTIMTimer
gdcm::network::ARTIMTimer, 154
AS
gdcm::VR, 1405
ASComp
gdcm, 86
ASN1
gdcm::ASN1, 156
AsynchronousOperationsWindowSub
gdcm::network::AsynchronousOperationsWindowSub, 157
AT
gdcm::VR, 1405
Attribute
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 190
gdcm::terminal, 110
Audio
gdcm::MediaStorage, 801
AudioCodec
gdcm::AudioCodec, 221
AUTOMATIC
gdcm::Segment, 1071
AutoPixelMinMax
gdcm::IconImageGenerator, 625
AXIAL
gdcm::Orientation, 886

backslash
gdcm, 91
BadBigEndian
gdcm::SwapCode, 1226
BadLittleEndian
gdcm::SwapCode, 1226
BALCPProtect
gdcm::Anonymizer, 139
Base64
gdcm::Base64, 223
BaseQuery
gdcm::BaseQuery, 234
BaseRootQuery
gdcm::BaseRootQuery, 239
BasicApplicationLevelConfidentialityProfile
gdcm::Anonymizer, 139
BasicCodedEntry
gdcm::SegmentHelper::BasicCodedEntry, 244
BasicCodedEntryVector
gdcm::Segment, 1070
BasicOffsetTable
gdcm::BasicOffsetTable, 249
BasicTextSR
gdcm::MediaStorage, 799
BasicVoiceAudioWaveformStorage
gdcm::MediaStorage, 798
Begin

gdcm::CSAHeaderDict, 356
gdcm::DataSet, 392
gdcm::Dict, 422
gdcm::IODs, 713
gdcm::Scanner, 1053
gdcm::Scanner2, 1062
gdcm::SequenceOfFragments, 1093
gdcm::SequenceOfItems, 1101, 1102
gdcm::StrictScanner, 1171
gdcm::StrictScanner2, 1180
BigEndian
gdcm::SwapCode, 1226
Bitmap
gdcm::Bitmap, 253
gdcm::JPEG2000Codec, 745
gdcm::PixelFormat, 938
BitmapToBitmapFilter
gdcm::BitmapToBitmapFilter, 266
BitSample
gdcm::JPEGCodec, 759
gdcm::LookupTable, 783
black
gdcm::terminal, 110
blink
gdcm::terminal, 110
BLUE
gdcm::LookupTable, 778
blue
gdcm::terminal, 110
BOOL_FUNCTION_PFILE_PFILE_POINTER
gdcm, 86
BoundingBox
gdcm::BoxRegion, 269
BoxRegion
gdcm::BoxRegion, 269
BreakConnection
gdcm::network::ULConnectionManager, 1358
BreakConnectionNow
gdcm::network::ULConnectionManager, 1358
BreastProjectionXRayImageStorageForPresentation
gdcm::MediaStorage, 800
BreastProjectionXRayImageStorageForProcessing
gdcm::MediaStorage, 800
BreastTomosynthesisImageStorage
gdcm::MediaStorage, 800
bright
gdcm::terminal, 110
Bug List, 7
Build
vtkLookupTable16, 1516
ByteBuffer
gdcm::ByteBuffer, 272
bytes
gdcm::Tag, 1254
ByteSwap
gdcm::ByteSwapFilter, 276
ByteSwapFilter
gdcm::ByteSwapFilter, 275, 276
ByteValue
gdcm::ByteValue, 279

C_CANCEL_RQ
gdcm::network::DIMSE, 441
C_ECHO_RQ
gdcm::network::DIMSE, 441
C_ECHO_RSP
gdcm::network::DIMSE, 441
C_FIND_RQ
gdcm::network::DIMSE, 440
C_FIND_RSP
gdcm::network::DIMSE, 441
C_GET_RQ
gdcm::network::DIMSE, 440
C_GET_RSP
gdcm::network::DIMSE, 440
C_MOVE_RQ
gdcm::network::DIMSE, 441
C_MOVE_RSP
gdcm::network::DIMSE, 441
C_STORE_RQ
gdcm::network::DIMSE, 440
C_STORE_RSP
gdcm::network::DIMSE, 440
CALIBRATED
gdcm::Spacing, 1147
CanCode
gdcm::AudioCodec, 222
gdcm::Coder, 311
gdcm::ImageCodec, 659
gdcm::JPEG2000Codec, 741
gdcm::JPEGCodec, 754
gdcm::JPEGLSCodec, 763
gdcm::KAKADUCodec, 771
gdcm::PDFCodec, 915
gdcm::PGXCodec, 925
gdcm::PNMCodec, 961
gdcm::PVRGCodec, 1002
gdcm::RAWCodec, 1022
gdcm::RLECodec, 1043
CanDecode
gdcm::AudioCodec, 222
gdcm::Decoder, 406
gdcm::DeltaEncodingCodec, 415
gdcm::ImageCodec, 659
gdcm::JPEG2000Codec, 741
gdcm::JPEGCodec, 754
gdcm::JPEGLSCodec, 763
gdcm::KAKADUCodec, 771

gdcm::PDFCodec, 915
gdcm::PGXCodec, 925
gdcm::PNMCodec, 961
gdcm::PVRGCodec, 1002
gdcm::RAWCodec, 1022
gdcm::RLECodec, 1043
CanDisplay
gdcm::VR, 1407
CanEmptyTag
gdcm::Anonymizer, 139
CanRead
gdcm::Reader, 1027
CanReadFile
vtkGDCMImageReader, 1421
vtkGDCMImageReader2, 1436
CanReadImage
gdcm::StreamImageReader, 1157
CanStoreLossy
gdcm::TransferSyntax, 1272
CanWriteFile
gdcm::StreamImageWriter, 1162
CAPI
gdcm::CryptoFactory, 336
CAPICryptoFactory
gdcm::CAPICryptoFactory, 286
CAPICryptographicMessageSyntax
gdcm::CAPICryptographicMessageSyntax, 288
CardiacElectrophysiologyWaveformStorage
gdcm::MediaStorage, 798
CEcho
gdcm::CompositeNetworkFunctions, 327
CFind
gdcm::CompositeNetworkFunctions, 327
Change
gdcm::FileChangeTransferSyntax, 560
gdcm::FileDecompressLookupTable, 563
gdcm::FileExplicitFilter, 569
gdcm::ImageChangePhotometricInterpretation, 643
gdcm::ImageChangePlanarConfiguration, 648
gdcm::ImageChangeTransferSyntax, 652
ChangeFMI
gdcm::FileExplicitFilter, 569
ChangeMonochrome
gdcm::ImageChangePhotometricInterpretation, 643
ChangeRGB2YBR
gdcm::ImageChangePhotometricInterpretation, 643
ChangeYBR2RGB
gdcm::ImageChangePhotometricInterpretation, 643
CharacterDataHandler
gdcm::TableReader, 1241
gdcm::XMLDictReader, 1541
gdcm::XMLPrivateDictReader, 1547
CheckDataElement
gdcm::FileStreamer, 596
CheckEvent
gdcm::AnonymizeEvent, 134
gdcm::DataEvent, 387
gdcm::DataSetEvent, 403
gdcm::Event, 534
gdcm::FileNameEvent, 587
gdcm::ProgressEvent, 998
CheckFileMetaInformationOff
gdcm::Writer, 1536
CheckFileMetaInformationOn
gdcm::Writer, 1536
CheckTemplateFileName
gdcm::FileStreamer, 596
CipherTypes
gdcm::CryptographicMessageSyntax, 338
Clamp
gdcm, 91
Clean
gdcm::Cleaner, 301
clean
gdcm, 91
Cleaner
gdcm::Cleaner, 301
CleanupUnusedBits
gdcm::ImageCodec, 659
Clear
gdcm::Anonymizer, 139
gdcm::Bitmap, 253
gdcm::ByteValue, 280
gdcm::DataElement, 373
gdcm::DataSet, 392
gdcm::IOD, 707
gdcm::IODs, 713
gdcm::Item, 723
gdcm::LookupTable, 779
gdcm::Macro, 787
gdcm::Macros, 789
gdcm::Module, 827
gdcm::Modules, 834
gdcm::Preamble, 964
gdcm::SequenceOfFragments, 1093
gdcm::SequenceOfItems, 1102
gdcm::SerieHelper, 1109
gdcm::Value, 1389
vtkGDCMMedicalImageProperties, 1458
vtkRTStructSetProperties, 1521
ClearInternalUIDs
gdcm::Anonymizer, 139
ClearPrivateTags
gdcm::Scanner2, 1062
gdcm::StrictScanner2, 1180
ClearPublicTags
gdcm::Scanner2, 1063
gdcm::StrictScanner2, 1181

ClearSkipTags
gdcm::Scanner, 1053
gdcm::Scanner2, 1063
gdcm::StrictScanner, 1171
gdcm::StrictScanner2, 1181
ClearTags
gdcm::Scanner, 1053
gdcm::StrictScanner, 1171
Clone
gdcm::BoxRegion, 269
gdcm::ImageCodec, 659
gdcm::JPEG2000Codec, 741
gdcm::JPEGCodec, 755
gdcm::JPEGLSCodec, 763
gdcm::KAKADUCodec, 771
gdcm::PGXCodec, 925
gdcm::PNMCodec, 962
gdcm::PVRGCodec, 1002
gdcm::RAWCodec, 1022
gdcm::Region, 1033
gdcm::RLECodec, 1043
CM
gdcm::SegmentHelper::BasicCodedEntry, 245
cMaxEventID
gdcm::network, 108
cMaxStateID
gdcm::network, 108
CMove
gdcm::CompositeNetworkFunctions, 328
CMYK
gdcm::PhotometricInterpretation, 928
Code
gdcm::Coder, 311
gdcm::JPEG2000Codec, 741
gdcm::JPEGCodec, 755
gdcm::JPEGLSCodec, 764
gdcm::JSON, 767
gdcm::KAKADUCodec, 771
gdcm::PVRGCodec, 1003
gdcm::RAWCodec, 1023
gdcm::RLECodec, 1043
CodeMeaning
gdcm::RealWorldValueMappingContent, 1032
CodeString
gdcm::CodeString, 314, 315
CodeValue
gdcm::RealWorldValueMappingContent, 1032
Color
gdcm::terminal, 110
ColorArray
gdcm::SurfaceHelper, 1212
Command
gdcm::Command, 318
CommandDataSet
gdcm::CommandDataSet, 322
CommandTypes
gdcm::network::DIMSE, 440
Common Directory Reference, 57
Compatible
gdcm::VM, 1401
gdcm::VR, 1407
Component
gdcm::PersonName, 921
CompOperators
gdcm, 89
ComprehensiveSR
gdcm::MediaStorage, 799
CompressionTypes
vtkGDCMImageWriter, 1450
Compute
gdcm::EquipmentManufacturer, 532
gdcm::MD5, 792
gdcm::SHA1, 1123
ComputeBoundingBox
gdcm::BoxRegion, 270
gdcm::Region, 1034
ComputeBufferLength
gdcm::ImageRegionReader, 688
ComputeByteLength
gdcm::SequenceOfFragments, 1093
ComputeDataElement
gdcm::DataSet, 392
ComputeDataSetMediaStorageSOPClass
gdcm::FileMetaInformation, 575
ComputeDataSetTransferSyntax
gdcm::FileMetaInformation, 576
ComputeDistAlongNormal
gdcm::DirectionCosines, 443
ComputedRadiographyImageStorage
gdcm::MediaStorage, 798
ComputeFile
gdcm::MD5, 792
gdcm::SHA1, 1123
ComputeFileMD5
gdcm::Testing, 1259
ComputeGroupLength
gdcm::DataSet, 392
ComputeInterceptSlopePixelType
gdcm::Rescaler, 1036
ComputeLength
gdcm::ByteValue, 280
gdcm::Fragment, 613
gdcm::SequenceOfFragments, 1094
gdcm::SequenceOfItems, 1102
ComputeLossyFlag
gdcm::Bitmap, 253
ComputeMD5
gdcm::Testing, 1259

ComputeMediaStorageFromModality
gdcm::ImageHelper, 674
ComputeMOSAICDimensions
gdcm::SplitMosaicFilter, 1150
ComputeMOSAICImagePositionPatient
gdcm::SplitMosaicFilter, 1150
ComputeMOSAICSliceNormal
gdcm::SplitMosaicFilter, 1150
ComputeMOSAICSlicePosition
gdcm::SplitMosaicFilter, 1150
ComputeNumberOfSurfaces
gdcm::SurfaceWriter, 1223
ComputeOffsetTable
gdcm::JPEGCodec, 755
ComputePixelAspectRatioFromPixelSpacing
gdcm::Spacing, 1148
ComputePixelTypeFromMinMax
gdcm::Rescaler, 1036
ComputeSpacingFromImagePositionPatient
gdcm::ImageHelper, 674
ComputeTargetMediaStorage
gdcm::ImageWriter, 695
ComputeVR
gdcm::DataSetHelper, 405
ComputeZSpacing
gdcm::IPPSorter, 719
ConcatenatePDVBlobs
gdcm::network::PresentationDataValue, 980
ConcatenatePDVBlobsAsExplicit
gdcm::network::PresentationDataValue, 980
CONDENSED_STYLE
gdcm::Printer, 985
Conditional
gdcm::Usage, 1378
CONSOLE
gdcm::terminal, 110
const
gdcm::SOPClassUIDToIOD, 1140
const_iterator
gdcm::CodeString, 313
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1187
const_reference
gdcm::CodeString, 313
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1187
const_reverse_iterator
gdcm::CodeString, 313
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1188
ConstCharWrapper
gdcm::ConstCharWrapper, 331
ConstIterator
gdcm::CSAHeaderDict, 355
gdcm::DataSet, 391
gdcm::Dict, 421
gdcm::Scanner, 1051
gdcm::SequenceOfFragments, 1092
gdcm::SequenceOfItems, 1100
gdcm::StrictScanner, 1169
Construct
gdcm::BaseRootQuery, 240
ConstructAbortPDU
gdcm::network::PDUFactory, 917
ConstructCEchoRQ
gdcm::network::CompositeMessageFactory, 325
ConstructCFindRQ
gdcm::network::CompositeMessageFactory, 325
ConstructCMoveRQ
gdcm::network::CompositeMessageFactory, 325
ConstructCStoreRQ
gdcm::network::CompositeMessageFactory, 325
ConstructCStoreRSP
gdcm::network::CompositeMessageFactory, 325
ConstructFromString
gdcm::DPath, 452
gdcm::TagPath, 1256
ConstructFromTagList
gdcm::TagPath, 1256
ConstructNAction
gdcm::network::NormalizedMessageFactory, 864
ConstructNCreate
gdcm::network::NormalizedMessageFactory, 864
ConstructNDelete
gdcm::network::NormalizedMessageFactory, 865
ConstructNEventReport
gdcm::network::NormalizedMessageFactory, 865
ConstructNGet
gdcm::network::NormalizedMessageFactory, 865
ConstructNSet
gdcm::network::NormalizedMessageFactory, 865
ConstructorType
gdcm::Dicts, 436
ConstructPDU
gdcm::network::PDUFactory, 917
ConstructPDV
gdcm::network::BaseCompositeMessage, 227
gdcm::network::BaseNormalizedMessage, 229
gdcm::network::CEchoRQ, 292
gdcm::network::CFindRQ, 296
gdcm::network::CMoveRQ, 307
gdcm::network::CStoreRQ, 363
gdcm::network::CStoreRSP, 364
gdcm::network::NActionRQ, 846
gdcm::network::NCreateRQ, 849

gdcm::network::NDeleteRQ, 852
gdcm::network::NEventReportRQ, 858
gdcm::network::NGetRQ, 861
gdcm::network::NSetRQ, 869
ConstructPDVByDataSet
gdcm::network::CEchoRSP, 293
gdcm::network::CFindCancelRQ, 295
gdcm::network::CFindRSP, 298
gdcm::network::CMoveCancelRq, 306
gdcm::network::CMoveRSP, 308
gdcm::network::NActionRSP, 847
gdcm::network::NCreateRSP, 850
gdcm::network::NDeleteRSP, 853
gdcm::network::NEventReportRSP, 860
gdcm::network::NGetRSP, 863
gdcm::network::NSetRSP, 870
ConstructQuery
gdcm::CompositeNetworkFunctions, 328, 329
gdcm::NormalizedNetworkFunctions, 866
ConstructReleasePDU
gdcm::network::PDUFactory, 917
Convert
gdcm::DictConverter, 426
gdcm::ImageConverter, 668
ConvertRGBToPaletteColor
gdcm::IconImageGenerator, 625
ConvertToCXX
gdcm::DictConverter, 426
ConvertToUNC
gdcm::System, 1230
ConvertToXML
gdcm::DictConverter, 426
CORONAL
gdcm::Orientation, 886
Create
gdcm::Preamble, 964
CreateCEchoPDU
gdcm::network::PDUFactory, 917
CreateCFindPDU
gdcm::network::PDUFactory, 917
CreateCMovePDU
gdcm::network::PDUFactory, 917
CreateCMSProvider
gdcm::CAPICryptoFactory, 286
gdcm::CryptoFactory, 337
gdcm::OpenSSLCryptoFactory, 876
gdcm::OpenSSLP7CryptoFactory, 881
CreateCStoreRQPDU
gdcm::network::PDUFactory, 917
CreateCStoreRSPPDU
gdcm::network::PDUFactory, 918
CreateDefaultUniqueSeriesIdentifier
gdcm::SerieHelper, 1110
CreateNActionPDU
gdcm::network::PDUFactory, 918
CreateNCreatePDU
gdcm::network::PDUFactory, 918
CreateNDeletePDU
gdcm::network::PDUFactory, 918
CreateNEventReportPDU
gdcm::network::PDUFactory, 918
CreateNGetPDU
gdcm::network::PDUFactory, 918
CreateNSetPDU
gdcm::network::PDUFactory, 918
CreateUniqueSeriesIdentifier
gdcm::SerieHelper, 1110
Cross
gdcm::DirectionCosines, 443
CrossDot
gdcm::DirectionCosines, 443
CryptoFactory
gdcm::CryptoFactory, 336
CryptographicMessageSyntax
gdcm::CryptographicMessageSyntax, 338, 339
CryptoLib
gdcm::CryptoFactory, 336
CS
gdcm::VR, 1406
CSAElement
gdcm::CSAElement, 343
CSAHeader
gdcm::CSAHeader, 351
gdcm::DataSet, 400
CSAHeaderDict
gdcm::CSAHeaderDict, 355
CSAHeaderDictEntry
gdcm::CSAHeaderDictEntry, 359
CSAHeaderType
gdcm::CSAHeader, 351
CSANonImageStorage
gdcm::MediaStorage, 799
CSComp
gdcm, 86
CSD
gdcm::SegmentHelper::BasicCodedEntry, 245
CStore
gdcm::CompositeNetworkFunctions, 329
CSV
gdcm::SegmentHelper::BasicCodedEntry, 245
CT_private_ELE
gdcm::TransferSyntax, 1271
CTImageStorage
gdcm::MediaStorage, 798
Curve
gdcm::Curve, 366
vtkGDCMImageReader, 1430
vtkGDCMImageReader2, 1445

Curves
gdcm::Pixmap, 945
CV
gdcm::SegmentHelper::BasicCodedEntry, 245
CXX
gdcm::Printer, 985
cyan
gdcm::terminal, 110

DA
gdcm::VR, 1406
DAComp
gdcm, 86
DataDictionary Directory Reference, 59
DataElement
gdcm::DataElement, 373
gdcm::Value, 1390
DataElementSet
gdcm::DataSet, 391
DataElementType
gdcm::ModuleEntry, 832
DataEvent
gdcm::DataEvent, 386, 387
DataField
gdcm::CSAElement, 348
DataPtr
gdcm::CSAElement, 343
DATASET_FORMAT
gdcm::CSAHeader, 351
DataSetEvent
gdcm::DataSetEvent, 403
DataSetHandled
gdcm::network::ULConnectionCallback, 1352
DataSetHandles
gdcm::network::ULConnectionCallback, 1352
DataSetMS
gdcm::FileMetaInformation, 580
DataSetTS
gdcm::FileMetaInformation, 580
DataStructureAndEncodingDefinition Directory Reference, 60
DataWasPassed
vtkImageMapToColors16, 1502
dCor
gdcm::MrProtocol::Vector3, 1391
DebugOff
gdcm::Trace, 1265
DebugOn
gdcm::Trace, 1265
Decode
gdcm::AudioCodec, 222
gdcm::Base64, 223
gdcm::Curve, 367
gdcm::Decoder, 406
gdcm::DeltaEncodingCodec, 415
gdcm::ImageCodec, 659
gdcm::JPEG2000Codec, 741
gdcm::JPEGCodec, 755
gdcm::JPEGLSCodec, 764
gdcm::JSON, 767
gdcm::KAKADUCodec, 772
gdcm::LookupTable, 779
gdcm::PDFCodec, 916
gdcm::PVRGCodec, 1003
gdcm::RAWCodec, 1023
gdcm::RLECodec, 1044
Decode8
gdcm::LookupTable, 779
DecodeByStreams
gdcm::Decoder, 406
gdcm::ImageCodec, 660
gdcm::JPEG12Codec, 731
gdcm::JPEG16Codec, 736
gdcm::JPEG2000Codec, 742
gdcm::JPEG8Codec, 749
gdcm::JPEGCodec, 755
gdcm::RAWCodec, 1023
gdcm::RLECodec, 1044
DecodeBytes
gdcm::RAWCodec, 1023
DecodeExtent
gdcm::JPEG2000Codec, 742
gdcm::JPEGCodec, 756
gdcm::JPEGLSCodec, 764
gdcm::RLECodec, 1044
Decompress
gdcm::Overlay, 892
Decrypt
gdcm::CAPICryptographicMessageSyntax, 288
gdcm::CryptographicMessageSyntax, 339
gdcm::OpenSSLCryptographicMessageSyntax, 878
gdcm::OpenSSLP7CryptographicMessageSyntax, 883
DeepCopy
vtkRTStructSetProperties, 1521
DEFAULT
gdcm::CryptoFactory, 336
Default
gdcm::FileMetaInformation, 576
DefinedTerms
gdcm::DefinedTerms, 407
DefinePixelExtent
gdcm::StreamImageReader, 1157
gdcm::StreamImageWriter, 1162
DefineProperBufferLength
gdcm::StreamImageReader, 1157
gdcm::StreamImageWriter, 1162
DeflatedExplicitVRLittleEndian

gdcm::TransferSyntax, 1271
Defs
gdcm::Defs, 409
DeleteDirectory
gdcm::System, 1230
DeltaEncodingCodec
gdcm::DeltaEncodingCodec, 415
Deprecated List, 5
Derive
gdcm::FileDerivation, 566
DES3_CIPHER
gdcm::CryptographicMessageSyntax, 338
Description
gdcm::ModuleEntry, 830
DescriptionField
gdcm::ModuleEntry, 832
DetachedPatientManagementSOPClass
gdcm::MediaStorage, 799
DetachedStudyManagementSOPClass
gdcm::MediaStorage, 799
DetachedVisitManagementSOPClass
gdcm::MediaStorage, 799
DETECTOR
gdcm::Spacing, 1147
DetermineEventByPDU
gdcm::network::PDUFactory, 919
DICOMDIR
gdcm::DICOMDIR, 416
DICOMDIRGenerator
gdcm::DICOMDIRGenerator, 418
Dict
gdcm::Dict, 421
gdcm::DictEntry, 432
DICT_DEBUG
gdcm::DictConverter, 425
DICT_DEFAULT
gdcm::DictConverter, 425
DICT_XML
gdcm::DictConverter, 425
DictConverter
gdcm::DictConverter, 425
DictEntry
gdcm::DictEntry, 429
DictPrinter
gdcm::DictPrinter, 434
Dicts
gdcm::CSAHeaderDict, 357
gdcm::Dict, 424
gdcm::Dicts, 437
gdcm::PrivateDict, 989
difference_type
gdcm::CodeString, 313
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1188
DigitalIntraoralXrayImageStorageForPresentation
gdcm::MediaStorage, 798
DigitalIntraoralXRayImageStorageForProcessing
gdcm::MediaStorage, 798
DigitalMammographyImageStorageForPresentation
gdcm::MediaStorage, 798
DigitalMammographyImageStorageForProcessing
gdcm::MediaStorage, 798
DigitalXRayImageStorageForPresentation
gdcm::MediaStorage, 798
DigitalXRayImageStorageForProcessing
gdcm::MediaStorage, 798
dim
gdcm::terminal, 110
Dimensions
gdcm::Bitmap, 263
gdcm::ImageCodec, 666
DirCosTolerance
gdcm::IPPSorter, 719
DirectionCosines
gdcm::DirectionCosines, 442
vtkGDCMImageReader, 1430
vtkGDCMImageReader2, 1445
Directory
gdcm::Directory, 446
DoByteSwap
gdcm::ImageCodec, 660
DoIconImage
gdcm::PixmapWriter, 956
DoInvertMonochrome
gdcm::ImageCodec, 660
DoOverlayCleanup
gdcm::ImageCodec, 660
DoPaddedCompositePixelCode
gdcm::ImageCodec, 660
DoPlanarConfiguration
gdcm::ImageCodec, 660
doround
gdcm, 91
DoSimpleCopy
gdcm::ImageCodec, 661
Dot
gdcm::DirectionCosines, 443
DoYBR
gdcm::ImageCodec, 661
DoYBRFull422
gdcm::ImageCodec, 661
DPath
gdcm::DPath, 452
DropDuplicatePositions
gdcm::IPPSorter, 719
DS

gdcm::VR, 1406
dSag
gdcm::MrProtocol::Vector3, 1391
DT
gdcm::VR, 1406
DTComp
gdcm, 87
dTra
gdcm::MrProtocol::Vector3, 1391
Dumper
gdcm::Dumper, 456
DuplicateAttributeError
gdcm::Parser, 902

eAABORTPDUReceivedOpen
gdcm::network, 107
eAABORTRequest
gdcm::network, 107
eAASSOCIATE_RQPDUreceived
gdcm::network, 107
eAASSOCIATERequestLocalUser
gdcm::network, 107
eAASSOCIATEresponseAccept
gdcm::network, 107
eAASSOCIATEresponseReject
gdcm::network, 107
eArabic
gdcm, 89
eARELEASE_RPPDUReceived
gdcm::network, 107
eARELEASE_RQPDUReceivedOpen
gdcm::network, 107
eARELEASERequest
gdcm::network, 107
eARELEASEResponse
gdcm::network, 107
eARTIMTimerExpired
gdcm::network, 107
eASSOCIATE_ACPDUreceived
gdcm::network, 107
eASSOCIATE_RJPDUreceived
gdcm::network, 107
ECharSet
gdcm, 89
eCreateMMPS
gdcm, 90
eCyrillic
gdcm, 89
EDGE
gdcm::MeshPrimitive, 814
eEventDoesNotExist
gdcm::network, 107
EEventID
gdcm::network, 107
eFind
gdcm, 90
eGB18030
gdcm, 89
eGreek
gdcm, 89
eHebrew
gdcm, 89
eImage
gdcm, 90
eJapanese
gdcm, 89
eJapaneseKanjiMultibyte
gdcm, 89
eJapaneseSupplementaryKanjiMultibyte
gdcm, 89
eKoreanHangulHanjaMultibyte
gdcm, 89
eLatin1
gdcm, 89
eLatin2
gdcm, 89
eLatin3
gdcm, 89
eLatin4
gdcm, 89
eLatin5
gdcm, 89
Element
gdcm::Element< TVR, VM::VM1_n >, 470
eMove
gdcm, 90
Empty
gdcm::Anonymizer, 140
gdcm::BoxRegion, 270
gdcm::Cleaner, 301
gdcm::DataElement, 373
gdcm::FileAnonymizer, 555
gdcm::Region, 1034
EmptyMaskGenerator
gdcm::EmptyMaskGenerator, 522
EmptyWhenScrubFails
gdcm::Cleaner, 302
EncapsulatedCDAStorage
gdcm::MediaStorage, 799
EncapsulatedDocument
gdcm::EncapsulatedDocument, 524
EncapsulatedPDFStorage
gdcm::MediaStorage, 799
Encode
gdcm::Base64, 224
EncodeBuffer
gdcm::JPEG12Codec, 731
gdcm::JPEG16Codec, 736

gdcm::JPEG8Codec, 749
gdcm::JPEGCodec, 756
EncodeBytes
gdcm::System, 1230
Encrypt
gdcm::CAPICryptographicMessageSyntax, 288
gdcm::CryptographicMessageSyntax, 339
gdcm::OpenSSLCryptographicMessageSyntax, 878
gdcm::OpenSSLP7CryptographicMessageSyntax, 883
End
gdcm::CSAHeaderDict, 356
gdcm::DataSet, 392
gdcm::Dict, 422
gdcm::IODs, 713
gdcm::Scanner, 1053
gdcm::Scanner2, 1063
gdcm::SequenceOfFragments, 1094
gdcm::SequenceOfItems, 1102
gdcm::StrictScanner, 1171
gdcm::StrictScanner2, 1181
EndElement
gdcm::TableReader, 1241
gdcm::XMLDictReader, 1541
gdcm::XMLPrivateDictReader, 1547
EndElementHandler
gdcm::Parser, 901
EndFilter
gdcm::SimpleSubjectWatcher, 1130
EndWith
gdcm::Filename, 582
EnhancedCTImageStorage
gdcm::MediaStorage, 798
EnhancedMRColorImageStorage
gdcm::MediaStorage, 800
EnhancedMRImageStorage
gdcm::MediaStorage, 798
EnhancedPETImageStorage
gdcm::MediaStorage, 800
EnhancedSR
gdcm::MediaStorage, 799
EnhancedUSVolumeStorage
gdcm::MediaStorage, 800
EnhancedXAImageStorage
gdcm::MediaStorage, 799
ENQueryType
gdcm, 89
EnumeratedValues
gdcm::EnumeratedValues, 531
ePatient
gdcm, 90
ePatientRootType
gdcm, 90
ePDATArequest
gdcm::network, 107
ePDATATFPDU
gdcm::network, 107
EQueryLevel
gdcm, 90
EQueryType
gdcm, 90
ERootType
gdcm, 90
ErrorOff
gdcm::Trace, 1265
ErrorOn
gdcm::Trace, 1265
ErrorType
gdcm::Parser, 902
eSeries
gdcm, 90
eSetMMPS
gdcm, 90
eSta10ReleaseCollisionAc
gdcm::network, 108
eSta11ReleaseCollisionRq
gdcm::network, 108
eSta12ReleaseCollisionAcLocal
gdcm::network, 108
eSta13AwaitingClose
gdcm::network, 108
eSta1Idle
gdcm::network, 108
eSta2Open
gdcm::network, 108
eSta3WaitLocalAssoc
gdcm::network, 108
eSta4LocalAssocDone
gdcm::network, 108
eSta5WaitRemoteAssoc
gdcm::network, 108
eSta6TransferReady
gdcm::network, 108
eSta7WaitRelease
gdcm::network, 108
eSta8WaitLocalRelease
gdcm::network, 108
eSta9ReleaseCollisionRqLocal
gdcm::network, 108
EstablishConnection
gdcm::network::ULConnectionManager, 1358
EstablishConnectionMove
gdcm::network::ULConnectionManager, 1359
eStaDoesNotExist
gdcm::network, 107
EStateID
gdcm::network, 107
eStudy

gdcm, 90
eStudyRootType
gdcm, 90
eThai
gdcm, 89
eTransportConnConfirmLocal
gdcm::network, 107
eTransportConnectionClosed
gdcm::network, 107
eTransportConnIndicLocal
gdcm::network, 107
eUnrecognizedPDUReceived
gdcm::network, 107
eUTF8
gdcm, 89
Event
gdcm::Event, 534
eWLMFind
gdcm, 90
Exception
gdcm::Exception, 537
Execute
gdcm::Command, 319
gdcm::EmptyMaskGenerator, 522
gdcm::MemberCommand< T >, 809
gdcm::SimpleMemberCommand< T >, 1128
ExecuteData
vtkGDCMImageReader, 1421
vtkGDCMThreadedImageReader, 1475
ExecuteInformation
vtkGDCMImageReader, 1422
vtkGDCMThreadedImageReader, 1475
ExecuteQuery
gdcm::StringFilter, 1192
Explicit
gdcm::TransferSyntax, 1271
ExplicitVRBigEndian
gdcm::TransferSyntax, 1271
ExplicitVRLittleEndian
gdcm::TransferSyntax, 1271
Explore
gdcm::Directory, 447
Extract
gdcm::IconImageFilter, 623
ExtractIconImages
gdcm::IconImageFilter, 623
ExtractVeproIconImages
gdcm::IconImageFilter, 623

F
gdcm::Printer, 987
gdcm::Reader, 1031
gdcm::Validate, 1386
gdcm::XMLPrinter, 1545
FACET
gdcm::MeshPrimitive, 814
FD
gdcm::VR, 1406
Fiducials
gdcm::Fiducials, 548
File
gdcm::File, 550
FileAnonymizer
gdcm::FileAnonymizer, 555
FileChangeTransferSyntax
gdcm::FileChangeTransferSyntax, 559
gdcm::ImageCodec, 665
FileDecompressLookupTable
gdcm::FileDecompressLookupTable, 563
FileDerivation
gdcm::FileDerivation, 565
FileExists
gdcm::System, 1230
FileExplicitFilter
gdcm::FileExplicitFilter, 569
FileIsDirectory
gdcm::System, 1230
FileIsSymlink
gdcm::System, 1231
FileList
gdcm, 87
FileMetaInformation
gdcm::FileMetaInformation, 575
FileName
vtkGDCMPolyDataReader, 1463
Filename
gdcm::Filename, 582
filename
gdcm::FileWithName, 601
FileNameEvent
gdcm::FileNameEvent, 586
FilenameGenerator
gdcm::FilenameGenerator, 589
FileNameOrdering
gdcm::SerieHelper, 1110
FileNames
vtkGDCMImageReader, 1431
Filenames
gdcm::Sorter, 1145
FilenamesType
gdcm::DICOMDIRGenerator, 418
gdcm::Directory, 446
gdcm::FilenameGenerator, 589
FilenameType
gdcm::DICOMDIRGenerator, 418
gdcm::Directory, 446
gdcm::FilenameGenerator, 589
FileSet

gdcm::FileSet, 592
FileSize
gdcm::System, 1231
FileStreamer
gdcm::FileStreamer, 596
FilesType
gdcm::FileSet, 592
FileTime
gdcm::System, 1231
FileType
gdcm::FileSet, 592
FileWithName
gdcm::FileWithName, 601
Fill
gdcm::ByteValue, 280
FillFromDataSet
gdcm::FileMetaInformation, 576
FillMedicalImageInformation
vtkGDCMImageReader, 1422
vtkGDCMImageReader2, 1436
vtkGDCMPolyDataReader, 1461
FindContext
gdcm::network::ULConnection, 1347
FindCSAElementByName
gdcm::CSAHeader, 351
FindDataElement
gdcm::DataSet, 393
gdcm::Item, 723
gdcm::SequenceOfItems, 1102
FindDictEntry
gdcm::PrivateDict, 988
FindMacroEntry
gdcm::Macro, 787
FindModuleEntryInMacros
gdcm::Module, 827
FindMrProtocolByName
gdcm::MrProtocol, 844
FindNextDataElement
gdcm::DataSet, 393
FindPatientRootQuery
gdcm::FindPatientRootQuery, 604
FindPDBElementByName
gdcm::PDBHeader, 912
FindStudyRootQuery
gdcm::FindStudyRootQuery, 608
FirstRender
vtkImageColorViewer, 1496
FL
gdcm::VR, 1406
FLOAT16
gdcm::PixelFormat, 933
FLOAT32
gdcm::PixelFormat, 933
FLOAT64
gdcm::PixelFormat, 933
ForceRescale
vtkGDCMImageReader, 1431
vtkGDCMImageReader2, 1446
FormatDateTime
gdcm::System, 1231
Fragment
gdcm::Fragment, 613
FragmentVector
gdcm::SequenceOfFragments, 1092
FromString
gdcm::StringFilter, 1192
FUJI
gdcm::EquipmentManufacturer, 532
FujiPrivateCRImageStorage
gdcm::MediaStorage, 800
FujiPrivateMammoCRImageStorage
gdcm::MediaStorage, 800

gdcm, 71
add1, 91
AEComp, 86
ASComp, 86
backslash, 91
BOOL_FUNCTION_PFILE_PFILE_POINTER, 86
Clamp, 91
clean, 91
CompOperators, 89
CSComp, 86
DAComp, 86
doround, 91
DTComp, 87
eArabic, 89
ECharSet, 89
eCreateMMPS, 90
eCyrillic, 89
eFind, 90
eGB18030, 89
eGreek, 89
eHebrew, 89
eImage, 90
eJapanese, 89
eJapaneseKanjiMultibyte, 89
eJapaneseSupplementaryKanjiMultibyte, 89
eKoreanHangulHanjaMultibyte, 89
eLatin1, 89
eLatin2, 89
eLatin3, 89
eLatin4, 89
eLatin5, 89
eMove, 90
ENQueryType, 89
ePatient, 90
ePatientRootType, 90

EQueryLevel, 90
EQueryType, 90
ERootType, 90
eSeries, 90
eSetMMPS, 90
eStudy, 90
eStudyRootType, 90
eThai, 89
eUTF8, 89
eWLMFind, 90
FileList, 87
GDCM_DIFFERENT, 89
GDCM_EQUAL, 89
GDCM_GREATER, 89
GDCM_GREATEROREQUAL, 89
GDCM_LESS, 89
GDCM_LESSOREQUAL, 89
GetVRFromTag, 92
GlobalInstance, 102
IconImage, 87
LD_ALL, 91
LD_NOSEQ, 91
LD_NOSHADOW, 91
LD_NOSHADOWSEQ, 91
LOComp, 87
LodModeType, 90
LTComp, 87
MacroEntry, 87
NestedMacroEntries, 87
operator!=, 92
operator<<, 92–100
operator>>, 101
operator==, 101
PNComp, 87
Round, 101
roundat, 101
SHComp, 88
STComp, 88
TMComp, 88
UCComp, 88
UIComp, 88
URComp, 88
UTComp, 88
x16printf, 102
GDCM Documentation, 1
gdcm::AbortEvent, 128
gdcm::AnonymizeEvent, 131
~AnonymizeEvent, 133
AnonymizeEvent, 133
CheckEvent, 134
GetEventName, 134
GetTag, 134
MakeObject, 134
operator=, 134
Self, 133
SetTag, 134
Superclass, 133
gdcm::Anonymizer, 135
~Anonymizer, 138
Anonymizer, 138
BALCPProtect, 139
BasicApplicationLevelConfidentialityProfile, 139
CanEmptyTag, 139
Clear, 139
ClearInternalUIDs, 139
Empty, 140
GetBasicApplicationLevelConfidentialityProfileAttributes, 140
GetCryptographicMessageSyntax, 140
GetFile, 140
New, 141
RecurseDataSet, 141
Remove, 141
RemoveGroupLength, 141
RemovePrivateTags, 141
RemoveRetired, 142
Replace, 142
SetCryptographicMessageSyntax, 143
SetFile, 143
gdcm::AnyEvent, 144
gdcm::ApplicationEntity, 147
Internal, 149
IsValid, 148
MaxLength, 149
MaxNumberOfComponents, 149
Padding, 149
Print, 148
Separator, 149
SetBlob, 148
Squeeze, 148
gdcm::ASN1, 155
~ASN1, 156
ASN1, 156
operator=, 156
ParseDump, 156
ParseDumpFile, 156
TestPBKDF2, 157
gdcm::Attribute< Group, Element, TVR, TVM >, 158
ArrayType, 160
GDCM_STATIC_ASSERT, 161
GetAsDataElement, 161
GetDictVM, 162
GetDictVR, 162
GetNumberOfValues, 162
GetTag, 162
GetValue, 163
GetValues, 163
GetVM, 163

GetVR, 164
Internal, 167
operator!=, 164
operator<, 164
operator==, 164
operator[], 164, 165
Print, 165
Set, 165
SetByteValue, 165
SetByteValueNoSwap, 165
SetFromDataElement, 166
SetFromDataSet, 166
SetValue, 166
SetValues, 167
VMType, 161
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 168
ArrayType, 170
GDCM_STATIC_ASSERT, 171
GetAsDataElement, 171
GetDictVM, 171
GetDictVR, 172
GetNumberOfValues, 172
GetTag, 172
GetValue, 172
GetValues, 172
GetVM, 172
GetVR, 173
Internal, 175
operator!=, 173
operator<, 173
operator==, 173
operator[], 173
Print, 173
Set, 174
SetByteValue, 174
SetByteValueNoSwap, 174
SetFromDataElement, 174
SetFromDataSet, 174
SetValue, 175
SetValues, 175
VMType, 171
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 176
ArrayType, 178
GDCM_STATIC_ASSERT, 178
GetAsDataElement, 178
GetDictVM, 178
GetDictVR, 178
GetNumberOfValues, 179
GetTag, 179
GetValue, 179
GetValues, 179
GetVM, 179
GetVR, 179
Internal, 181
operator!=, 179
operator<, 179
operator==, 179
operator[], 180
Print, 180
Set, 180
SetByteValue, 180
SetByteValueNoSwap, 180
SetFromDataElement, 180
SetFromDataSet, 180
SetValue, 180
SetValues, 181
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 181
ArrayType, 183
GDCM_STATIC_ASSERT, 184
GetAsDataElement, 184
GetDictVM, 184
GetDictVR, 184
GetNumberOfValues, 184
GetTag, 184
GetValue, 184
GetValues, 184
GetVM, 185
GetVR, 185
Internal, 187
operator!=, 185
operator<, 185
operator==, 185
operator[], 185
Print, 185
Set, 185
SetByteValue, 186
SetByteValueNoSwap, 186
SetFromDataElement, 186
SetFromDataSet, 186
SetValue, 186
SetValues, 186
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 187
~Attribute, 190
ArrayType, 189
Attribute, 190
GDCM_STATIC_ASSERT, 190
GetAsDataElement, 190
GetDictVM, 190
GetDictVR, 191
GetNumberOfValues, 191
GetTag, 191
GetValue, 191
GetValues, 191
GetVM, 192
GetVR, 192

operator!=, 192
operator<, 192
operator==, 192
operator[], 192
Print, 193
Set, 193
SetByteValue, 193
SetByteValueNoSwap, 193
SetFromDataElement, 193
SetFromDataSet, 193
SetNumberOfValues, 194
SetValue, 194
SetValues, 194
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 195
ArrayType, 198
GDCM_STATIC_ASSERT, 198
GetAsDataElement, 198
GetDictVM, 199
GetDictVR, 199
GetNumberOfValues, 199
GetTag, 199
GetValue, 199
GetValues, 199
GetVM, 199
GetVR, 199
Internal, 201
operator!=, 199
operator<, 200
operator==, 200
operator[], 200
Print, 200
Set, 200
SetByteValue, 200
SetByteValueNoSwap, 200
SetFromDataElement, 200
SetFromDataSet, 201
SetValue, 201
SetValues, 201
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 202
ArrayType, 204
GDCM_STATIC_ASSERT, 204
GetAsDataElement, 204
GetDictVM, 204
GetDictVR, 204
GetNumberOfValues, 205
GetTag, 205
GetValue, 205
GetValues, 205
GetVM, 205
GetVR, 205
Internal, 207
operator!=, 205
operator<, 205
operator==, 205
operator[], 206
Print, 206
Set, 206
SetByteValue, 206
SetByteValueNoSwap, 206
SetFromDataElement, 206
SetFromDataSet, 206
SetValue, 206
SetValues, 207
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 207
ArrayType, 210
GDCM_STATIC_ASSERT, 211
GetAsDataElement, 211
GetDictVM, 211
GetDictVR, 211
GetNumberOfValues, 211
GetTag, 211
GetValue, 211
GetValues, 211
GetVM, 212
GetVR, 212
Internal, 214
operator!=, 212
operator<, 212
operator==, 212
operator[], 212
Print, 212
Set, 212
SetByteValue, 213
SetByteValueNoSwap, 213
SetFromDataElement, 213
SetFromDataSet, 213
SetValue, 213
SetValues, 213
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 214
ArrayType, 216
GDCM_STATIC_ASSERT, 217
GetAsDataElement, 217
GetDictVM, 217
GetDictVR, 217
GetNumberOfValues, 217
GetTag, 217
GetValue, 217
GetValues, 217
GetVM, 217
GetVR, 218
Internal, 219
operator!=, 218
operator<, 218
operator==, 218

operator[], 218
Print, 218
Set, 218
SetByteValue, 218
SetByteValueNoSwap, 219
SetFromDataElement, 219
SetFromDataSet, 219
SetValue, 219
SetValues, 219
gdcm::AudioCodec, 220
~AudioCodec, 221
AudioCodec, 221
CanCode, 222
CanDecode, 222
Decode, 222
gdcm::Base64, 222
Base64, 223
Decode, 223
Encode, 224
GetDecodeLength, 224
GetEncodeLength, 224
operator=, 225
gdcm::BaseQuery, 232
~BaseQuery, 234
AddQueryDataSet, 234
BaseQuery, 234
GetAbstractSyntaxUID, 234
GetQueryDataSet, 235
GetSOPInstanceUID, 235
mDataSet, 237
mSopInstanceUID, 237
Print, 235
QueryFactory, 236
SetSearchParameter, 235
SetSOPInstanceUID, 236
ValidateQuery, 236
ValidDataSet, 236
WriteHelpFile, 236
WriteQuery, 236
gdcm::BaseRootQuery, 237
~BaseRootQuery, 239
BaseRootQuery, 239
Construct, 240
GetQueryLevelFromQueryRoot, 240
GetQueryLevelFromString, 240
GetQueryLevelString, 240
GetTagListByLevel, 240
InitializeDataSet, 240
mHelpDescription, 241
mImage, 241
mPatient, 242
mRootType, 242
mSeries, 242
mStudy, 242
QueryFactory, 241
ValidateQuery, 241
gdcm::BasicOffsetTable, 246
BasicOffsetTable, 249
operator<<, 249
Read, 249
gdcm::Bitmap, 250
~Bitmap, 253
AreOverlaysInPixelData, 253
Bitmap, 253
Clear, 253
ComputeLossyFlag, 253
Dimensions, 263
GetBuffer, 254
GetBuffer2, 254
GetBufferLength, 254
GetColumns, 254
GetDataElement, 254
GetDimension, 255
GetDimensions, 255
GetLUT, 255
GetNeedByteSwap, 255
GetNumberOfDimensions, 256
GetPhotometricInterpretation, 256
GetPixelFormat, 256
GetPlanarConfiguration, 256
GetRows, 257
GetTransferSyntax, 257
ImageChangeTransferSyntax, 262
IsEmpty, 257
IsLossy, 257
IsTransferSyntaxCompatible, 257
LossyFlag, 263
LUT, 263
LUTPtr, 253
NeedByteSwap, 263
NumberOfDimensions, 263
PF, 263
PI, 263
PixelData, 264
PixmapReader, 262
PlanarConfiguration, 264
Print, 257
SetColumns, 258
SetDataElement, 258
SetDimension, 258
SetDimensions, 258
SetLossyFlag, 259
SetLUT, 259
SetNeedByteSwap, 259
SetNumberOfDimensions, 259
SetPhotometricInterpretation, 259
SetPixelFormat, 260
SetPlanarConfiguration, 260

SetRows, 260
SetTransferSyntax, 260
TryJPEG2000Codec, 261
TryJPEG2000Codec2, 261
TryJPEGCodec, 261
TryJPEGCodec2, 261
TryJPEGLSCodec, 261
TryKAKADUCodec, 261
TryPVRGCodec, 262
TryRAWCodec, 262
TryRLECodec, 262
TS, 264
UnusedBitsPresentInPixelData, 262
gdcm::BitmapToBitmapFilter, 264
~BitmapToBitmapFilter, 266
BitmapToBitmapFilter, 266
GetOutput, 266
GetOutputAsBitmap, 266
Input, 266
Output, 266
SetInput, 266
gdcm::BoxRegion, 267
~BoxRegion, 269
Area, 269
BoundingBox, 269
BoxRegion, 269
Clone, 269
ComputeBoundingBox, 270
Empty, 270
GetXMax, 270
GetXMin, 270
GetYMax, 270
GetYMin, 270
GetZMax, 270
GetZMin, 271
IsValid, 271
operator=, 271
Print, 271
SetDomain, 271
gdcm::ByteBuffer, 272
ByteBuffer, 272
Get, 272
GetStart, 272
ShiftEnd, 272
UpdatePosition, 273
gdcm::ByteSwap< T >, 273
Swap, 273
SwapFromSwapCodeIntoSystem, 273
SwapRange, 274
SwapRangeFromSwapCodeIntoSystem, 274
SystemIsBigEndian, 274
SystemIsLittleEndian, 274
gdcm::ByteSwapFilter, 275
~ByteSwapFilter, 275
ByteSwap, 276
ByteSwapFilter, 275, 276
operator=, 276
SetByteSwapTag, 276
gdcm::ByteValue, 277
~ByteValue, 279
Append, 280
ByteValue, 279
Clear, 280
ComputeLength, 280
Fill, 280
GetBuffer, 280
GetLength, 280
GetPointer, 281
GetVoidPointer, 281
IsEmpty, 281
IsPrintable, 282
operator const std::vector< char > &, 282
operator=, 282
operator==, 282
Print, 282
PrintASCII, 283
PrintASCIIXML, 283
PrintGroupLength, 283
PrintHex, 283
PrintHexXML, 283
PrintPNXML, 283
Read, 283, 284
SetLength, 284
SetLengthOnly, 284
Write, 284
WriteBuffer, 284
gdcm::CAPICryptoFactory, 285
CAPICryptoFactory, 286
CreateCMSProvider, 286
gdcm::CAPICryptographicMessageSyntax, 287
~CAPICryptographicMessageSyntax, 288
CAPICryptographicMessageSyntax, 288
Decrypt, 288
Encrypt, 288
GetCipherType, 289
GetInitialized, 289
ParseCertificateFile, 289
ParseKeyFile, 289
SetCipherType, 289
SetPassword, 289
gdcm::Cleaner, 298
~Cleaner, 301
Clean, 301
Cleaner, 301
Empty, 301
EmptyWhenScrubFails, 302
GetFile, 302
New, 302

Preserve, 302
Remove, 302, 303
RemoveAllGroupLength, 303
RemoveAllIllegal, 303
RemoveAllMissingPrivateCreator, 303
RemoveMissingPrivateCreator, 303
Scrub, 304
SetFile, 304
gdcm::Codec, 309
gdcm::Coder, 310
~Coder, 311
CanCode, 311
Code, 311
InternalCode, 311
gdcm::CodeString, 312
CodeString, 314, 315
const_iterator, 313
const_reference, 313
const_reverse_iterator, 313
difference_type, 313
GetAsString, 315
IsValid, 315
iterator, 314
operator!=, 316
operator<<, 316
operator==, 316
pointer, 314
reference, 314
reverse_iterator, 314
Size, 315
size_type, 314
TrimInternal, 315
value_type, 314
gdcm::Command, 316
~Command, 318
Command, 318
Execute, 319
operator=, 319
gdcm::CommandDataSet, 320
~CommandDataSet, 322
CommandDataSet, 322
Insert, 323
operator<<, 324
Read, 323
Replace, 323
Write, 323
gdcm::CompositeNetworkFunctions, 325
CEcho, 327
CFind, 327
CMove, 328
ConstructQuery, 328, 329
CStore, 329
KeyValuePairArrayType, 326
KeyValuePairType, 326
gdcm::ConstCharWrapper, 330
ConstCharWrapper, 331
operator const char *, 331
gdcm::CP246ExplicitDataElement, 331
GetLength, 334
Read, 334
ReadPreValue, 334
ReadValue, 334
ReadWithLength, 334
gdcm::CryptoFactory, 335
~CryptoFactory, 337
CAPI, 336
CreateCMSProvider, 337
CryptoFactory, 336
CryptoLib, 336
DEFAULT, 336
GetFactoryInstance, 337
OPENSSL, 336
OPENSSLP7, 336
gdcm::CryptographicMessageSyntax, 337
~CryptographicMessageSyntax, 338
AES128_CIPHER, 338
AES192_CIPHER, 338
AES256_CIPHER, 338
CipherTypes, 338
CryptographicMessageSyntax, 338, 339
Decrypt, 339
DES3_CIPHER, 338
Encrypt, 339
GetCipherType, 339
operator=, 339
ParseCertificateFile, 340
ParseKeyFile, 340
SetCipherType, 340
SetPassword, 340
gdcm::CSAElement, 341
CSAElement, 343
DataField, 348
DataPtr, 343
GetByteValue, 343
GetKey, 343
GetName, 344
GetNoOfItems, 344
GetSyngoDT, 344
GetValue, 344
GetVM, 345
GetVR, 345
IsEmpty, 345
KeyField, 348
NameField, 348
NoOfItemsField, 348
operator<, 345
operator<<, 347
operator=, 345

operator==, 346
SetByteValue, 346
SetKey, 346
SetName, 346
SetNoOfItems, 346
SetSyngoDT, 346
SetValue, 347
SetVM, 347
SetVR, 347
SyngoDTField, 348
ValueMultiplicityField, 348
VRField, 348
gdcm::CSAHeader, 349
~CSAHeader, 351
CSAHeader, 351
CSAHeaderType, 351
DATASET_FORMAT, 351
FindCSAElementByName, 351
GetCSADataInfo, 351
GetCSAEEnd, 352
GetCSAElementByName, 352
GetCSAImageHeaderInfoTag, 352
GetCSASeriesHeaderInfoTag, 352
GetDataSet, 352
GetFormat, 353
GetInterfile, 353
GetMrProtocol, 353
INTERFILE, 351
LoadFromDataElement, 353
NOMAGIC, 351
operator<<, 354
Print, 353
SV10, 351
UNKNOWN, 351
ZEROED_OUT, 351
gdcm::CSAHeaderDict, 354
AddCSAHeaderDictEntry, 356
Begin, 356
ConstIterator, 355
CSAHeaderDict, 355
Dicts, 357
End, 356
GetCSAHeaderDictEntry, 356
IsEmpty, 356
Iterator, 355
LoadDefault, 356
MapCSAHeaderDictEntry, 355
operator<<, 357
operator=, 356
gdcm::CSAHeaderDictEntry, 357
CSAHeaderDictEntry, 359
GetDescription, 359
GetName, 359
GetVM, 359
GetVR, 359
operator<, 359
operator<<, 360
SetDescription, 360
SetName, 360
SetVM, 360
SetVR, 360
gdcm::CSAHeaderDictException, 361
gdcm::Curve, 364
~Curve, 366
Curve, 366
Decode, 367
GetAsPoints, 367
GetCurveDataDescriptor, 367
GetDataValueRepresentation, 367
GetDimensions, 367
GetGroup, 367
GetNumberOfCurves, 367
GetNumberOfPoints, 367
GetTypeOfData, 367
GetTypeOfDataDescription, 368
IsEmpty, 368
Print, 368
SetCoordinateStartValue, 368
SetCoordinateStepValue, 368
SetCurve, 368
SetCurveDataDescriptor, 368
SetCurveDescription, 368
SetDataValueRepresentation, 369
SetDimensions, 369
SetGroup, 369
SetNumberOfPoints, 369
SetTypeOfData, 369
Update, 369
gdcm::DataElement, 370
Clear, 373
DataElement, 373
Empty, 373
GetByteValue, 373
GetLength, 374
GetSequenceOfFragments, 374
GetTag, 374, 375
GetValue, 375
GetValueAsSQ, 375
GetVL, 376
GetVR, 376
IsEmpty, 377
IsUndefinedLength, 377
operator<, 377
operator<<, 382
operator=, 378
operator==, 378
Read, 378
ReadOrSkip, 378

ReadPreValue, 378
ReadValue, 379
ReadValueWithLength, 379
ReadWithLength, 379
SetByteValue, 379
SetTag, 380
SetValue, 380
SetValueFieldLength, 381
SetVL, 381
SetVLToUndefined, 381
SetVR, 381
TagField, 383
ValueField, 383
ValueLengthField, 383
ValuePtr, 373
VRField, 383
Write, 382
gdcm::DataElementException, 384
gdcm::DataEvent, 384
~DataEvent, 386
CheckEvent, 387
DataEvent, 386, 387
GetData, 387
GetDataLength, 387
GetEventName, 387
MakeObject, 387
operator=, 387
Self, 386
SetData, 388
Superclass, 386
gdcm::DataSet, 388
Begin, 392
Clear, 392
ComputeDataElement, 392
ComputeGroupLength, 392
ConstIterator, 391
CSAHeader, 400
DataElementSet, 391
End, 392
FindDataElement, 393
FindNextDataElement, 393
GetDataElement, 393, 394
GetDEEnd, 394
GetDES, 394
GetLength, 395
GetMediaStorage, 395
GetPrivateCreator, 395
GetPrivateTag, 395
Insert, 395
InsertDataElement, 396
IsEmpty, 396
Iterator, 391
operator<<, 400
operator(), 396
operator=, 396
operator[], 397
Print, 397
Read, 397
ReadNested, 397
ReadSelectedPrivateTags, 397
ReadSelectedPrivateTagsWithLength, 397
ReadSelectedTags, 398
ReadSelectedTagsWithLength, 398
ReadUpToTag, 398
ReadUpToTagWithLength, 398
ReadWithLength, 398
Remove, 399
Replace, 399
ReplaceEmpty, 399
Size, 399
SizeType, 391
Write, 400
gdcm::DataSetEvent, 401
~DataSetEvent, 403
CheckEvent, 403
DataSetEvent, 403
GetDataSet, 403
GetEventName, 403
m_DataSet, 404
MakeObject, 403
operator=, 404
Self, 402
Superclass, 402
gdcm::DataSetHelper, 404
ComputeVR, 405
gdcm::Decoder, 405
~Decoder, 406
CanDecode, 406
Decode, 406
DecodeByStreams, 406
gdcm::DefinedTerms, 407
DefinedTerms, 407
gdcm::Defs, 408
~Defs, 409
Defs, 409
GetIODFromFile, 409
GetIODNameFromMediaStorage, 409
GetIODs, 409
GetMacros, 410
GetModules, 410
GetTypeFromTag, 410
Global, 411
IsEmpty, 410
LoadDefaults, 411
LoadFromFile, 411
operator=, 411
Verify, 411
gdcm::DeltaEncodingCodec, 412

~DeltaEncodingCodec, 415
CanDecode, 415
Decode, 415
DeltaEncodingCodec, 415
gdcm::DICOMDIR, 415
DICOMDIR, 416
gdcm::DICOMDIRGenerator, 416
~DICOMDIRGenerator, 418
AddImageDirectoryRecord, 418
AddPatientDirectoryRecord, 418
AddSeriesDirectoryRecord, 418
AddStudyDirectoryRecord, 418
DICOMDIRGenerator, 418
FilenamesType, 418
FilenameType, 418
Generate, 418
GetFile, 419
GetScanner, 419
SetDescriptor, 419
SetFile, 419
SetFilenames, 419
SetRootDirectory, 420
gdcm::Dict, 420
AddDictEntry, 422
Begin, 422
ConstIterator, 421
Dict, 421
Dicts, 424
End, 422
GetDictEntry, 422
GetDictEntryByKeyword, 422
GetDictEntryByName, 423
GetKeywordFromTag, 423
IsEmpty, 423
Iterator, 421
LoadDefault, 423
MapDictEntry, 421
operator<<, 424
operator=, 423
gdcm::DictConverter, 424
~DictConverter, 425
AddGroupLength, 426
Convert, 426
ConvertToCXX, 426
ConvertToXML, 426
DICT_DEBUG, 425
DICT_DEFAULT, 425
DICT_XML, 425
DictConverter, 425
GetDictName, 426
GetInputFilename, 426
GetOutputFilename, 426
GetOutputType, 426
OutputTypes, 425
Readuint16, 427
ReadVM, 427
ReadVR, 427
SetDictName, 427
SetInputFileName, 427
SetOutputFileName, 427
SetOutputType, 427
WriteFooter, 427
WriteHeader, 428
gdcm::DictEntry, 428
Dict, 432
DictEntry, 429
GetKeyword, 429
GetName, 429
GetRetired, 430
GetVM, 430
GetVR, 430
IsUnique, 430
operator<<, 432
SetElementXX, 430
SetGroupXX, 431
SetKeyword, 431
SetName, 431
SetRetired, 431
SetVM, 431
SetVR, 431
gdcm::DictPrinter, 432
~DictPrinter, 434
DictPrinter, 434
Print, 435
PrintDataElement2, 435
PrintDataSet2, 435
gdcm::Dicts, 435
~Dicts, 437
ConstructorType, 436
Dicts, 437
GEMS, 436
GetConstructorString, 437
GetCSAHeaderDict, 437
GetDictEntry, 437
GetPrivateDict, 438
GetPublicDict, 438
Global, 439
IsEmpty, 438
LoadDefaults, 438
operator<<, 439
operator=, 439
PHILIPS, 436
SIEMENS, 437
gdcm::DirectionCosines, 441
~DirectionCosines, 442
ComputeDistAlongNormal, 443
Cross, 443
CrossDot, 443

DirectionCosines, 442
Dot, 443
IsValid, 443
Norm, 444
Normalize, 444
operator const double ∗, 444
Print, 444
SetFromString, 444
gdcm::Directory, 445
∼Directory, 446
Directory, 446
Explore, 447
FilenamesType, 446
FilenameType, 446
GetDirectories, 447
GetFilenames, 447
GetToplevel, 447
Load, 447
operator<<, 448
Print, 448
gdcm::DirectoryHelper, 449
GetCTImageSeriesUIDs, 449
GetFilenamesFromSeriesUIDs, 449
GetFrameOfReference, 449
GetMRImageSeriesUIDs, 450
GetRTStructSeriesUIDs, 450
GetSeriesUIDsBySOPClassUID, 450
GetSOPClassUID, 450
GetStringValueFromTag, 450
LoadImageFromFiles, 450
RetrieveSOPInstanceUIDFromIndex, 450
RetrieveSOPInstanceUIDFromZPosition, 451
gdcm::DPath, 451
∼DPath, 452
ConstructFromString, 452
DPath, 452
IsValid, 452
Match, 452
operator<, 452
operator<<, 453
Print, 453
gdcm::DummyValueGenerator, 453
Generate, 454
gdcm::Dumper, 454
∼Dumper, 456
Dumper, 456
gdcm::Element< TVR, TVM >, 457
GetAsDataElement, 459
GetLength, 459
GetValue, 460
GetValues, 460
GetVM, 460
GetVR, 460
Internal, 463
operator[], 461
Print, 461
Read, 461
Set, 461
SetFromDataElement, 461
SetNoSwap, 462
SetValue, 462
Type, 459
Write, 462
gdcm::Element< TVR, VM::VM1_2 >, 463
GetAsDataElement, 466
GetLength, 466
GetValue, 466
GetValues, 466
GetVM, 466
GetVR, 467
Internal, 468
operator[], 467
Parent, 466
Print, 467
Read, 467
Set, 467
SetFromDataElement, 467
SetLength, 467
SetNoSwap, 467
SetValue, 468
Type, 466
Write, 468
gdcm::Element< TVR, VM::VM1_n >, 468
∼Element, 470
Element, 470
GetAsDataElement, 471
GetLength, 471
GetValue, 471
GetValues, 471
GetVM, 471
GetVR, 471
operator=, 472
operator[], 472
Print, 472
Read, 472
Set, 472
SetArray, 473
SetFromDataElement, 473
SetLength, 473
SetNoSwap, 473
SetValue, 473
Type, 470
Write, 474
WriteASCII, 474
gdcm::Element< TVR, VM::VM2_2n >, 474
GetAsDataElement, 479
GetLength, 479
GetValue, 479

GetValues, 479
GetVM, 479
GetVR, 479
Internal, 481
operator[], 479
Parent, 478
Print, 479
Read, 479
Set, 480
SetFromDataElement, 480
SetLength, 480
SetNoSwap, 480
SetValue, 480
Type, 478
Write, 480
gdcm::Element< TVR, VM::VM2_n >, 481
GetAsDataElement, 484
GetLength, 484
GetValue, 484
GetValues, 484
GetVM, 484
GetVR, 485
Internal, 486
operator[], 485
Parent, 484
Print, 485
Read, 485
Set, 485
SetFromDataElement, 485
SetLength, 485
SetNoSwap, 485
SetValue, 486
Type, 484
Write, 486
gdcm::Element< TVR, VM::VM3_3n >, 486
GetAsDataElement, 491
GetLength, 491
GetValue, 491
GetValues, 491
GetVM, 491
GetVR, 491
Internal, 493
operator[], 491
Parent, 490
Print, 491
Read, 491
Set, 492
SetFromDataElement, 492
SetLength, 492
SetNoSwap, 492
SetValue, 492
Type, 490
Write, 492
gdcm::Element< TVR, VM::VM3_4 >, 493
GetAsDataElement, 496
GetLength, 496
GetValue, 496
GetValues, 496
GetVM, 496
GetVR, 497
Internal, 498
operator[], 497
Parent, 496
Print, 497
Read, 497
Set, 497
SetFromDataElement, 497
SetLength, 497
SetNoSwap, 497
SetValue, 498
Type, 496
Write, 498
gdcm::Element< TVR, VM::VM3_n >, 498
GetAsDataElement, 502
GetLength, 502
GetValue, 502
GetValues, 502
GetVM, 502
GetVR, 503
Internal, 504
operator[], 503
Parent, 502
Print, 503
Read, 503
Set, 503
SetFromDataElement, 503
SetLength, 503
SetNoSwap, 503
SetValue, 504
Type, 502
Write, 504
gdcm::Element< VR::AS, VM::VM5 >, 504
GetAsDataElement, 506
GetLength, 506
GetValue, 506
GetValues, 506
GetVM, 506
GetVR, 507
Internal, 508
operator[], 507
Print, 507
Read, 507
Set, 507
SetFromDataElement, 507
SetNoSwap, 507
SetValue, 507
Type, 506
Write, 508

gdcm::Element< VR::OB, VM::VM1 >, 508
GetAsDataElement, 511
GetLength, 511
GetValue, 511
GetValues, 511
GetVM, 511
GetVR, 511
Internal, 513
operator[], 511
Print, 512
Read, 512
Set, 512
SetFromDataElement, 512
SetNoSwap, 512
SetValue, 512
Type, 511
Write, 512
gdcm::Element< VR::OW, VM::VM1 >, 513
GetAsDataElement, 516
GetLength, 516
GetValue, 516
GetValues, 516
GetVM, 516
GetVR, 516
Internal, 518
operator[], 516
Print, 517
Read, 517
Set, 517
SetFromDataElement, 517
SetNoSwap, 517
SetValue, 517
Type, 516
Write, 517
gdcm::ElementDisableCombinations< TVR, TVM >, 518
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >, 519
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >, 520
gdcm::EmptyMaskGenerator, 521
~EmptyMaskGenerator, 522
EmptyMaskGenerator, 522
Execute, 522
SetInputDirectory, 522
SetOutputDirectory, 522
SetSOPClassUIDMode, 523
SOPClassUIDMode, 522
UseGrayscaleSecondaryImageStorage, 522
UseOriginalSOPClassUID, 522
gdcm::EncapsulatedDocument, 523
EncapsulatedDocument, 524
gdcm::EncodingImplementation< T >, 524
gdcm::EncodingImplementation< VR::VRASCII >, 525
Read, 526
ReadComputeLength, 526
ReadNoSwap, 526
Write, 526, 527
gdcm::EncodingImplementation< VR::VRBINARY >, 527
Read, 528
ReadComputeLength, 528
ReadNoSwap, 528
Write, 529
gdcm::EndEvent, 529
gdcm::EnumeratedValues, 530
EnumeratedValues, 531
gdcm::EquipmentManufacturer, 531
AGFA, 532
Compute, 532
FUJI, 532
GEMS, 532
HITACHI, 532
KODAK, 532
MARCONI, 532
PMS, 532
SAMSUNG, 532
SIEMENS, 532
TOSHIBA, 532
Type, 532
TypeToString, 532
UIH, 532
UNKNOWN, 532
gdcm::Event, 533
~Event, 534
CheckEvent, 534
Event, 534
GetEventName, 534
MakeObject, 535
operator=, 535
Print, 535
gdcm::Exception, 536
~Exception, 537
Exception, 537
GetDescription, 537
what, 537
gdcm::ExitEvent, 538
gdcm::ExplicitDataElement, 539
GetLength, 542
Read, 542
ReadPreValue, 542
ReadValue, 543
ReadWithLength, 543
Write, 543
gdcm::ExplicitImplicitDataElement, 543
GetLength, 546
Read, 546
ReadPreValue, 546
ReadValue, 547
ReadWithLength, 547

gdcm::Fiducials, 547
Fiducials, 548
gdcm::File, 548
~File, 550
File, 550
GetDataSet, 550
GetHeader, 551
operator<<, 552
Read, 551
SetDataSet, 551
SetHeader, 552
Write, 552
gdcm::FileAnonymizer, 553
~FileAnonymizer, 555
Empty, 555
FileAnonymizer, 555
Remove, 555
Replace, 556
SetInputFileName, 556
SetOutputFileName, 556
Write, 557
gdcm::FileChangeTransferSyntax, 557
~FileChangeTransferSyntax, 559
Change, 560
FileChangeTransferSyntax, 559
GetCodec, 560
New, 560
SetInputFileName, 560
SetOutputFileName, 560
SetTransferSyntax, 561
gdcm::FileDecompressLookupTable, 561
~FileDecompressLookupTable, 563
Change, 563
FileDecompressLookupTable, 563
GetFile, 563
GetPixmap, 564
SetFile, 564
SetPixmap, 564
gdcm::FileDerivation, 564
~FileDerivation, 565
AddDerivationDescription, 566
AddPurposeOfReferenceCodeSequence, 566
AddReference, 566
AddSourceImageSequence, 566
Derive, 566
FileDerivation, 565
GetFile, 566, 567
SetAppendDerivationHistory, 567
SetDerivationCodeSequenceCodeValue, 567
SetDerivationDescription, 567
SetFile, 567
SetPurposeOfReferenceCodeSequenceCodeValue, 568
gdcm::FileExplicitFilter, 568
~FileExplicitFilter, 569
Change, 569
ChangeFMI, 569
FileExplicitFilter, 569
GetFile, 570
ProcessDataSet, 570
SetChangePrivateTags, 570
SetFile, 570
SetRecomputeItemLength, 570
SetRecomputeSequenceLength, 570
SetUseVRUN, 570
gdcm::FileMetaInformation, 571
~FileMetaInformation, 575
AppendImplementationClassUID, 575
ComputeDataSetMediaStorageSOPClass, 575
ComputeDataSetTransferSyntax, 576
DataSetMS, 580
DataSetTS, 580
Default, 576
FileMetaInformation, 575
FillFromDataSet, 576
GetDataSetTransferSyntax, 576
GetFileMetaInformationVersion, 576
GetFullLength, 576
GetGDCMImplementationClassUID, 576
GetGDCMImplementationVersionName, 577
GetGDCMSourceApplicationEntityTitle, 577
GetImplementationClassUID, 577
GetImplementationVersionName, 577
GetMediaStorage, 577
GetMediaStorageAsString, 577
GetMetaInformationTS, 577
GetPreamble, 577
GetSourceApplicationEntityTitle, 578
Insert, 578
IsValid, 578
MetaInformationTS, 580
operator<<, 580
operator=, 578
Read, 578
ReadCompat, 578
ReadCompatInternal, 578
Replace, 579
SetDataSetTransferSyntax, 579
SetImplementationClassUID, 579
SetImplementationVersionName, 579
SetPreamble, 579
SetSourceApplicationEntityTitle, 579
Write, 580
gdcm::Filename, 581
EndWith, 582
Filename, 582
GetExtension, 582
GetFileName, 582

GetName, 582
GetPath, 582
IsEmpty, 583
IsIdentical, 583
Join, 583
operator const char ∗, 583
ToUnixSlashes, 583
ToWindowsSlashes, 583
gdcm::FileNameEvent, 584
∼FileNameEvent, 586
CheckEvent, 587
FileNameEvent, 586
GetEventName, 587
GetFileName, 587
MakeObject, 587
operator=, 587
Self, 586
SetFileName, 587
Superclass, 586
gdcm::FilenameGenerator, 588
∼FilenameGenerator, 589
FilenameGenerator, 589
FilenamesType, 589
FilenameType, 589
Generate, 589
GetFilename, 589
GetFilenames, 590
GetNumberOfFilenames, 590
GetPattern, 590
GetPrefix, 590
SetNumberOfFilenames, 590
SetPattern, 590
SetPrefix, 591
SizeType, 589
gdcm::FileSet, 591
AddFile, 592
FileSet, 592
FilesType, 592
FileType, 592
GetFiles, 592
operator<<, 593
SetFiles, 593
gdcm::FileStreamer, 593
∼FileStreamer, 596
AppendToDataElement, 596
AppendToGroupDataElement, 596
CheckDataElement, 596
CheckTemplateFileName, 596
FileStreamer, 596
New, 597
ReserveDataElement, 597
ReserveGroupDataElement, 597
SetOutputFileName, 597
SetTemplateFileName, 597
StartDataElement, 598
StartGroupDataElement, 598
StopDataElement, 598
StopGroupDataElement, 598
gdcm::FileWithName, 599
filename, 601
FileWithName, 601
gdcm::FindPatientRootQuery, 602
FindPatientRootQuery, 604
GetAbstractSyntaxUID, 604
GetTagListByLevel, 604
InitializeDataSet, 605
QueryFactory, 605
ValidateQuery, 605
gdcm::FindStudyRootQuery, 606
FindStudyRootQuery, 608
GetAbstractSyntaxUID, 608
GetTagListByLevel, 608
InitializeDataSet, 609
QueryFactory, 609
ValidateQuery, 609
gdcm::Fragment, 610
ComputeLength, 613
Fragment, 613
GetLength, 613
operator<<, 614
Read, 613
ReadBacktrack, 613
ReadPreValue, 613
ReadValue, 614
Write, 614
gdcm::Global, 615
∼Global, 616
Append, 616
GetDefs, 616
GetDicts, 616, 617
GetInstance, 617
Global, 616
LoadResourcesFiles, 617
Locate, 617
operator<<, 618
operator=, 618
Prepend, 618
gdcm::GroupDict, 618
∼GroupDict, 619
Add, 620
GetAbbreviation, 620
GetName, 620
GroupDict, 619
GroupStringVector, 619
Insert, 620
operator<<, 621
Size, 620
gdcm::IconImageFilter, 621

~IconImageFilter, 622
Extract, 623
ExtractIconImages, 623
ExtractVeproIconImages, 623
GetFile, 623
GetIconImage, 623
GetNumberOfIconImages, 623
IconImageFilter, 622
SetFile, 624
gdcm::IconImageGenerator, 624
~IconImageGenerator, 625
AutoPixelMinMax, 625
ConvertRGBToPaletteColor, 625
Generate, 626
GetIconImage, 626
GetPixmap, 626
IconImageGenerator, 625
SetOutputDimensions, 626
SetOutsideValuePixel, 626
SetPixelMinMax, 627
SetPixmap, 627
gdcm::ignore_char, 627
ignore_char, 628
m_char, 628
gdcm::Image, 628
~Image, 633
GetDirectionCosines, 634
GetIntercept, 634
GetOrigin, 634
GetSlope, 634
GetSpacing, 634
Image, 633
Print, 635
SetDirectionCosines, 635
SetIntercept, 635
SetOrigin, 635, 636
SetSlope, 636
SetSpacing, 636
gdcm::ImageApplyLookupTable, 637
~ImageApplyLookupTable, 639
Apply, 640
ImageApplyLookupTable, 639
SetRGB8, 640
gdcm::ImageChangePhotometricInterpretation, 640
~ImageChangePhotometricInterpretation, 643
Change, 643
ChangeMonochrome, 643
ChangeRGB2YBR, 643
ChangeYBR2RGB, 643
GetPhotometricInterpretation, 643
ImageChangePhotometricInterpretation, 643
RGB2YBR, 644
SetPhotometricInterpretation, 644
YBR2RGB, 644
gdcm::ImageChangePlanarConfiguration, 645
~ImageChangePlanarConfiguration, 648
Change, 648
GetPlanarConfiguration, 648
ImageChangePlanarConfiguration, 648
RGBPixelsToRGBPlanes, 648
RGBPlanesToRGBPixels, 648
SetPlanarConfiguration, 649
gdcm::ImageChangeTransferSyntax, 649
~ImageChangeTransferSyntax, 652
Change, 652
GetTransferSyntax, 652
ImageChangeTransferSyntax, 652
SetCompressIconImage, 653
SetForce, 653
SetTransferSyntax, 653
SetUserCodec, 653
TryJPEG2000Codec, 654
TryJPEGCodec, 654
TryJPEGLSCodec, 654
TryRAWCodec, 654
TryRLECodec, 654
gdcm::ImageCodec, 655
~ImageCodec, 658
AppendFrameEncode, 658
AppendRowEncode, 658
CanCode, 659
CanDecode, 659
CleanupUnusedBits, 659
Clone, 659
Decode, 659
DecodeByStreams, 660
Dimensions, 666
DoByteSwap, 660
DoInvertMonochrome, 660
DoOverlayCleanup, 660
DoPaddedCompositePixelCode, 660
DoPlanarConfiguration, 660
DoSimpleCopy, 661
DoYBR, 661
DoYBRFull422, 661
FileChangeTransferSyntax, 665
GetDimensions, 661
GetHeaderInfo, 661
GetLossyFlag, 661
GetLUT, 661
GetNeedByteSwap, 662
GetNumberOfDimensions, 662
GetPhotometricInterpretation, 662
GetPixelFormat, 662
GetPlanarConfiguration, 662
ImageChangePhotometricInterpretation, 665
ImageCodec, 658
IsFrameEncoder, 662

IsLossy, 663
IsRowEncoder, 663
IsValid, 663
LossyFlag, 666
LUT, 666
LUTPtr, 658
NeedByteSwap, 666
NeedOverlayCleanup, 666
NumberOfDimensions, 666
PF, 666
PI, 667
PlanarConfiguration, 667
RequestPaddedCompositePixelCode, 667
RequestPlanarConfiguration, 667
SetDimensions, 663
SetLossyFlag, 663
SetLUT, 663
SetNeedByteSwap, 664
SetNeedOverlayCleanup, 664
SetNumberOfDimensions, 664
SetPhotometricInterpretation, 664
SetPixelFormat, 664
SetPlanarConfiguration, 665
StartEncode, 665
StopEncode, 665
gdcm::ImageConverter, 667
~ImageConverter, 668
Convert, 668
GetOuput, 668
ImageConverter, 668
SetInput, 668
gdcm::ImageFragmentSplitter, 669
~ImageFragmentSplitter, 671
GetFragmentSizeMax, 672
ImageFragmentSplitter, 671
SetForce, 672
SetFragmentSizeMax, 672
Split, 672
gdcm::ImageHelper, 672
ComputeMediaStorageFromModality, 674
ComputeSpacingFromImagePositionPatient, 674
GetDimensionsValue, 674
GetDirectionCosinesFromDataSet, 674
GetDirectionCosinesValue, 675
GetForcePixelSpacing, 675
GetForceRescaleInterceptSlope, 675
GetLUT, 675
GetOriginValue, 675
GetPhotometricInterpretationValue, 675
GetPixelFormatValue, 675
GetPlanarConfigurationValue, 676
GetPMSRescaleInterceptSlope, 676
GetPointerFromElement, 676
GetRealWorldValueMappingContent, 676
GetRescaleInterceptSlopeValue, 676
GetSecondaryCaptureImagePlaneModule, 676
GetSpacingTagFromMediaStorage, 677
GetSpacingValue, 677
GetZSpacingTagFromMediaStorage, 677
SetDimensionsValue, 677
SetDirectionCosinesValue, 677
SetForcePixelSpacing, 677
SetForceRescaleInterceptSlope, 677
SetOriginValue, 678
SetPMSRescaleInterceptSlope, 678
SetRescaleInterceptSlopeValue, 678
SetSecondaryCaptureImagePlaneModule, 678
SetSpacingValue, 678
gdcm::ImageReader, 679
~ImageReader, 682
GetImage, 682
ImageReader, 682
Read, 683
ReadACRNEMAImage, 683
ReadImage, 683
gdcm::ImageRegionReader, 684
~ImageRegionReader, 687
ComputeBufferLength, 688
GetRegion, 688
ImageRegionReader, 687
Read, 688
ReadInformation, 688
ReadIntoBuffer, 688
SetRegion, 689
gdcm::ImageToImageFilter, 689
~ImageToImageFilter, 691
GetInput, 691
GetOutput, 691
ImageToImageFilter, 691
gdcm::ImageWriter, 692
~ImageWriter, 695
ComputeTargetMediaStorage, 695
GetImage, 695, 696
ImageWriter, 695
Write, 696
gdcm::ImplicitDataElement, 700
GetLength, 703
Read, 703
ReadPreValue, 703
ReadValue, 703
ReadValueWithLength, 703
ReadWithLength, 703
Write, 703
gdcm::InitializeEvent, 704
gdcm::IOD, 705
AddIODEntry, 707
Clear, 707
GetIODEntry, 707

GetNumberOfIODs, 707
GetTypeFromTag, 707
IOD, 707
MapIODEntry, 706
operator<<, 708
SizeType, 706
gdcm::IODEntry, 708
GetIE, 709
GetName, 709
GetRef, 710
GetUsage, 710
GetUsageType, 710
IODEntry, 709
operator<<, 711
SetIE, 710
SetName, 710
SetRef, 710
SetUsage, 710
gdcm::IODs, 711
AddIOD, 713
Begin, 713
Clear, 713
End, 713
GetIOD, 713
IODMapType, 712
IODMapTypeConstIterator, 712
IODName, 712
IODs, 712
operator<<, 713
gdcm::IPPSorter, 714
ComputeZSpacing, 719
DirCosTolerance, 719
DropDuplicatePositions, 719
GetDirectionCosinesTolerance, 716
GetZSpacing, 716
GetZSpacingTolerance, 717
IPPSorter, 716
SetComputeZSpacing, 717
SetDirectionCosinesTolerance, 717
SetDropDuplicatePositions, 718
SetZSpacingTolerance, 718
Sort, 718
ZSpacing, 719
ZTolerance, 719
gdcm::Item, 720
Clear, 723
FindDataElement, 723
GetDataElement, 724
GetLength, 724
GetNestedDataSet, 724
InsertDataElement, 724
Item, 723
operator<<, 725
Read, 724
SetNestedDataSet, 725
Write, 725
gdcm::IterationEvent, 726
gdcm::JPEG12Codec, 727
~JPEG12Codec, 731
DecodeByStreams, 731
EncodeBuffer, 731
GetHeaderInfo, 731
InternalCode, 731
IsStateSuspension, 731
JPEG12Codec, 731
gdcm::JPEG16Codec, 732
~JPEG16Codec, 736
DecodeByStreams, 736
EncodeBuffer, 736
GetHeaderInfo, 736
InternalCode, 736
IsStateSuspension, 736
JPEG16Codec, 736
gdcm::JPEG2000Codec, 737
~JPEG2000Codec, 740
AppendFrameEncode, 740
AppendRowEncode, 740
Bitmap, 745
CanCode, 741
CanDecode, 741
Clone, 741
Code, 741
Decode, 741
DecodeByStreams, 742
DecodeExtent, 742
GetHeaderInfo, 742
GetQuality, 742
GetRate, 742
ImageRegionReader, 745
IsFrameEncoder, 743
IsRowEncoder, 743
JPEG2000Codec, 740
SetMCT, 743
SetNumberOfResolutions, 743
SetNumberOfThreadsForDecompression, 743
SetQuality, 743
SetRate, 744
SetReversible, 744
SetTileSize, 744
StartEncode, 744
StopEncode, 744
gdcm::JPEG8Codec, 745
~JPEG8Codec, 749
DecodeByStreams, 749
EncodeBuffer, 749
GetHeaderInfo, 749
InternalCode, 749
IsStateSuspension, 749

JPEG8Codec, 749
gdcm::JPEGCodec, 750
~JPEGCodec, 754
AppendFrameEncode, 754
AppendRowEncode, 754
BitSample, 759
CanCode, 754
CanDecode, 754
Clone, 755
Code, 755
ComputeOffsetTable, 755
Decode, 755
DecodeByStreams, 755
DecodeExtent, 756
EncodeBuffer, 756
GetHeaderInfo, 756
GetLossless, 756
GetQuality, 756
ImageRegionReader, 758
IsFrameEncoder, 757
IsRowEncoder, 757
IsStateSuspension, 757
IsValid, 757
JPEGCodec, 754
Quality, 759
SetBitSample, 757
SetLossless, 757
SetPixelFormat, 757
SetQuality, 758
StartEncode, 758
StopEncode, 758
gdcm::JPEGLSCodec, 759
~JPEGLSCodec, 762
AppendFrameEncode, 763
AppendRowEncode, 763
CanCode, 763
CanDecode, 763
Clone, 763
Code, 764
Decode, 764
DecodeExtent, 764
GetBufferLength, 764
GetHeaderInfo, 765
GetLossless, 765
ImageRegionReader, 766
IsFrameEncoder, 765
IsRowEncoder, 765
JPEGLSCodec, 762
SetBufferLength, 765
SetLossless, 765
SetLossyError, 765
StartEncode, 765
StopEncode, 766
gdcm::JSON, 766
~JSON, 767
Code, 767
Decode, 767
GetPrettyPrint, 767
JSON, 767
PrettyPrintOff, 767
PrettyPrintOn, 768
SetPrettyPrint, 768
gdcm::KAKADUCodec, 768
~KAKADUCodec, 771
CanCode, 771
CanDecode, 771
Clone, 771
Code, 771
Decode, 772
KAKADUCodec, 771
gdcm::LO, 772
const_iterator, 774
const_reference, 774
const_reverse_iterator, 774
difference_type, 774
IsValid, 775
iterator, 774
LO, 775
pointer, 774
reference, 774
reverse_iterator, 774
size_type, 774
Superclass, 774
value_type, 775
gdcm::LookupTable, 776
~LookupTable, 778
Allocate, 779
BitSample, 783
BLUE, 778
Clear, 779
Decode, 779
Decode8, 779
GetBitSample, 780
GetBufferAsRGBA, 780
GetLUT, 780
GetLUTDescriptor, 780
GetLUTLength, 780
GetPointer, 780
GRAY, 778
GREEN, 778
IncompleteLUT, 783
InitializeBlueLUT, 781
Initialized, 781
InitializeGreenLUT, 781
InitializeLUT, 781
InitializeRedLUT, 781
Internal, 783
IsRGB8, 781

LookupTable, 778, 779
LookupTableType, 778
Print, 782
RED, 778
SetBlueLUT, 782
SetGreenLUT, 782
SetLUT, 782
SetRedLUT, 782
UNKNOWN, 778
WriteBufferAsRGBA, 782
gdcm::Macro, 785
AddMacroEntry, 787
ArrayIncludeMacrosType, 786
Clear, 787
FindMacroEntry, 787
GetMacroEntry, 787
GetName, 787
Macro, 787
MapModuleEntry, 786
operator<<, 788
SetName, 787
Verify, 788
gdcm::Macros, 788
AddMacro, 789
Clear, 789
GetMacro, 790
IsEmpty, 790
Macros, 789
ModuleMapType, 789
operator<<, 790
gdcm::MD5, 792
Compute, 792
ComputeFile, 792
gdcm::MEC_MR3, 793
GetCanonMECMR3Tag, 793
GetPMTFInformationDataTag, 793
GetToshibaMECMR3Tag, 793
Print, 794
gdcm::MediaStorage, 794
AmbulatoryECGWaveformStorage, 798
Audio, 801
BasicTextSR, 799
BasicVoiceAudioWaveformStorage, 798
BreastProjectionXRayImageStorageForPresentation, 800
BreastProjectionXRayImageStorageForProcessing, 800
BreastTomosynthesisImageStorage, 800
CardiacElectrophysiologyWaveformStorage, 798
ComprehensiveSR, 799
ComputedRadiographyImageStorage, 798
CSANonImageStorage, 799
CTImageStorage, 798
DetachedPatientManagementSOPClass, 799
DetachedStudyManagementSOPClass, 799
DetachedVisitManagementSOPClass, 799
DigitalIntraoralXrayImageStorageForPresentation, 798
DigitalIntraoralXRayImageStorageForProcessing, 798
DigitalMammographyImageStorageForPresentation, 798
DigitalMammographyImageStorageForProcessing, 798
DigitalXRayImageStorageForPresentation, 798
DigitalXRayImageStorageForProcessing, 798
EncapsulatedCDAStorage, 799
EncapsulatedPDFStorage, 799
EnhancedCTImageStorage, 798
EnhancedMRColorImageStorage, 800
EnhancedMRImageStorage, 798
EnhancedPETImageStorage, 800
EnhancedSR, 799
EnhancedUSVolumeStorage, 800
EnhancedXAImageStorage, 799
FujiPrivateCRImageStorage, 800
FujiPrivateMammoCRImageStorage, 800
GeneralECGWaveformStorage, 798
GeneralElectricMagneticResonanceImageStorage, 799
GEPrivate3DModelStorage, 799
GetModality, 801
GetModalityDimension, 801
GetMSString, 801
GetMSType, 801
GetNumberOfModality, 802
GetNumberOfMSString, 802
GetNumberOfMSType, 802
GetString, 802
GrayscaleSoftcopyPresentationStateStorageSOP-Class, 799
GuessFromModality, 802
HangingProtocolStorage, 799
HardcopyColorImageStorage, 800
HardcopyGrayscaleImageStorage, 799
HemodynamicWaveformStorage, 798
IsImage, 802
IsUndefined, 803
IVOCTForPresentation, 800
IVOCTForProcessing, 800
KeyObjectSelectionDocument, 799
LeadECGWaveformStorage, 798
LegacyConvertedEnhancedCTImageStorage, 800
LegacyConvertedEnhancedMRImageStorage, 800
LegacyConvertedEnhancedPETImageStorage, 800
MammographyCADSR, 799
MediaStorage, 801
MediaStorageDirectoryStorage, 798

ModalityPerformedProcedureStepSOPClass, 799
MRImageStorage, 798
MRSpectroscopyStorage, 798
MS_END, 800
MSType, 798
MultiframeGrayscaleByteSecondaryCaptureImageStorage, 798
MultiframeGrayscaleWordSecondaryCaptureImageStorage, 798
MultiframeSingleBitSecondaryCaptureImageStorage, 798
MultiframeTrueColorSecondaryCaptureImageStorage, 798
NoObject, 800
NuclearMedicineImageStorage, 799
NuclearMedicineImageStorageRetired, 798
ObjectEnd, 801
ObjectType, 800
operator MSType, 803
operator<<, 804
OphthalmicPhotography16BitImageStorage, 800
OphthalmicPhotography8BitImageStorage, 800
OphthalmicTomographyImageStorage, 800
PDF, 801
PETImageStorage, 799
Philips3D, 799
PhilipsPrivateMRSyntheticImageStorage, 799
RawDataStorage, 799
RTDoseStorage, 799
RTImageStorage, 799
RTIonBeamsTreatmentRecordStorage, 800
RTIonPlanStorage, 799
RTPlanStorage, 799
RTStructureSetStorage, 799
RTTreatmentSummaryRecordStorage, 800
SecondaryCaptureImageStorage, 798
Segmentation, 801
SegmentationStorage, 799
SetFromDataSet, 803
SetFromFile, 803
SetFromHeader, 803
SetFromModality, 804
SetFromSourceImageSequence, 804
SpacialFiducialsStorage, 799
SpacialRegistrationStorage, 799
StandaloneCurveStorage, 798
StandaloneModalityLUTStorage, 798
StandaloneOverlayStorage, 798
StandaloneVOILUTStorage, 798
StudyComponentManagementSOPClass, 799
SurfaceSegmentationStorage, 800
ToshibaPrivateDataStorage, 799
UltrasoundImageStorage, 798
UltrasoundImageStorageRetired, 798
UltrasoundMultiFrameImageStorage, 798
UltrasoundMultiFrameImageStorageRetired, 798
URI, 801
Video, 801
VideoEndoscopicImageStorage, 799
VideoMicroscopicImageStorage, 800
VideoPhotographicImageStorage, 800
VLEndoscopicImageStorage, 800
VLMicroscopicImageStorage, 800
VLPhotographicImageStorage, 799
VLWholeSlideMicroscopyImageStorage, 800
Waveform, 801
XRay3DAngiographicImageStorage, 799
XRay3DCraniofacialImageStorage, 800
XRayAngiographicBiPlaneImageStorageRetired, 799
XRayAngiographicImageStorage, 799
XRayRadiationDoseSR, 800
XRayRadiofluoroscopingImageStorage, 799
gdcm::MemberCommand< T >, 804
~MemberCommand, 808
Execute, 809
m_ConstMemberFunction, 810
m_MemberFunction, 810
m_This, 810
MemberCommand, 808
New, 809
operator=, 809
Self, 808
SetCallbackFunction, 809, 810
TConstMemberFunctionPointer, 808
TMemberFunctionPointer, 808
gdcm::MeshPrimitive, 811
~MeshPrimitive, 814
AddPrimitiveData, 814
EDGE, 814
FACET, 814
GetMPType, 814
GetMPTypeString, 814
GetNumberOfPrimitivesData, 814
GetPrimitiveData, 815
GetPrimitivesData, 815
GetPrimitiveType, 815
LINE, 814
MeshPrimitive, 814
MPType, 813
MPType_END, 814
PrimitiveData, 816
PrimitivesData, 813
PrimitiveType, 816
SetPrimitiveData, 815
SetPrimitivesData, 816
SetPrimitiveType, 816
TRIANGLE, 814
TRIANGLE_FAN, 814

TRIANGLE_STRIP, 814
VERTEX, 814
gdcm::ModalityPerformedProcedureStepCreateQuery, 816
GetAbstractSyntaxUID, 819
GetRequiredDataSet, 819
ModalityPerformedProcedureStepCreateQuery, 819
QueryFactory, 820
ValidateQuery, 819
gdcm::ModalityPerformedProcedureStepSetQuery, 820
GetAbstractSyntaxUID, 823
GetRequiredDataSet, 823
ModalityPerformedProcedureStepSetQuery, 822
QueryFactory, 823
ValidateQuery, 823
gdcm::ModifiedEvent, 824
gdcm::Module, 825
AddMacro, 826
AddModuleEntry, 826
ArrayIncludeMacrosType, 826
Clear, 827
FindModuleEntryInMacros, 827
GetModuleEntryInMacros, 827
GetName, 827
MapModuleEntry, 826
Module, 826
operator<<, 828
SetName, 827
Verify, 827
gdcm::ModuleEntry, 828
~ModuleEntry, 830
DataElementType, 832
Description, 830
DescriptionField, 832
GetDescription, 831
GetName, 831
GetType, 831
ModuleEntry, 830
Name, 832
operator<<, 832
SetDescription, 831
SetName, 831
SetType, 831
gdcm::Modules, 832
AddModule, 834
Clear, 834
GetModule, 834
IsEmpty, 834
ModuleMapType, 833
Modules, 833
operator<<, 834
gdcm::MovePatientRootQuery, 835
GetAbstractSyntaxUID, 837
GetTagListByLevel, 837
InitializeDataSet, 838
MovePatientRootQuery, 837
QueryFactory, 838
ValidateQuery, 838
gdcm::MoveStudyRootQuery, 839
GetAbstractSyntaxUID, 841
GetTagListByLevel, 841
InitializeDataSet, 842
MoveStudyRootQuery, 841
QueryFactory, 842
ValidateQuery, 842
gdcm::MrProtocol, 843
~MrProtocol, 843
FindMrProtocolByName, 844
GetMrProtocolByName, 844
GetSliceArray, 844
GetVersion, 844
Load, 844
MrProtocol, 843
operator<<, 845
Print, 844
gdcm::MrProtocol::Slice, 1132
Normal, 1133
Position, 1133
gdcm::MrProtocol::SliceArray, 1133
Slices, 1134
gdcm::MrProtocol::Vector3, 1391
dCor, 1391
dSag, 1391
dTra, 1391
gdcm::NestedModuleEntries, 854
AddModuleEntry, 856
GetModuleEntry, 856
GetNumberOfModuleEntries, 857
NestedModuleEntries, 856
operator<<, 857
SizeType, 856
gdcm::network, 102
cMaxEventID, 108
cMaxStateID, 108
eAABORTPDUReceivedOpen, 107
eAABORTRequest, 107
eAASSOCIATE_RQPDUreceived, 107
eAASSOCIATERequestLocalUser, 107
eAASSOCIATEresponseAccept, 107
eAASSOCIATEresponseReject, 107
eARELEASE_RPPDUReceived, 107
eARELEASE_RQPDUReceivedOpen, 107
eARELEASERequest, 107
eARELEASEResponse, 107
eARTIMTimerExpired, 107
eASSOCIATE_ACPDUreceived, 107
eASSOCIATE_RJPDUreceived, 107
eEventDoesNotExist, 107

EEventID, 107
ePDATArequest, 107
ePDATATFPDU, 107
eSta10ReleaseCollisionAc, 108
eSta11ReleaseCollisionRq, 108
eSta12ReleaseCollisionAcLocal, 108
eSta13AwaitingClose, 108
eSta1Idle, 108
eSta2Open, 108
eSta3WaitLocalAssoc, 108
eSta4LocalAssocDone, 108
eSta5WaitRemoteAssoc, 108
eSta6TransferReady, 108
eSta7WaitRelease, 108
eSta8WaitLocalRelease, 108
eSta9ReleaseCollisionRqLocal, 108
eStaDoesNotExist, 107
EStateID, 107
eTransportConnConfirmLocal, 107
eTransportConnectionClosed, 107
eTransportConnIndicLocal, 107
eUnrecognizedPDUReceived, 107
GetStateIndex, 108
gdcm::network::AAbortPDU, 113
AAbortPDU, 114
IsLastFragment, 114
Print, 114
Read, 114
SetReason, 115
SetSource, 115
Size, 115
Write, 115
gdcm::network::AAssociateACPDU, 116
AAssociateACPDU, 117
AAssociateRQPDU, 119
AddPresentationContextAC, 118
GetNumberOfPresentationContextAC, 118
GetPresentationContextAC, 118
GetUserInformation, 118
InitFromRQ, 118
IsLastFragment, 118
Print, 118
Read, 118
SetCalledAETitle, 119
SetCallingAETitle, 119
Size, 119
SizeType, 117
Write, 119
gdcm::network::AAssociateRJPDU, 120
AAssociateRJPDU, 121
IsLastFragment, 121
Print, 121
Read, 121
Size, 121
Write, 121
gdcm::network::AAssociateRQPDU, 122
AAssociateACPDU, 127
AAssociateRQPDU, 124
AddPresentationContext, 124
GetCalledAETitle, 124
GetCallingAETitle, 124
GetNumberOfPresentationContext, 125
GetPresentationContext, 125
GetPresentationContextByAbstractSyntax, 125
GetPresentationContextByID, 125
GetPresentationContexts, 125
GetReserved43_74, 125
GetUserInformation, 125
IsAETitleValid, 125
IsLastFragment, 126
PresentationContextArrayType, 124
Print, 126
Read, 126
SetCalledAETitle, 126
SetCallingAETitle, 126
SetUserInformation, 126
Size, 127
SizeType, 124
Write, 127
gdcm::network::AbstractSyntax, 129
AbstractSyntax, 129
GetAsDataElement, 130
GetName, 130
operator==, 130
Print, 130
Read, 130
SetName, 130
SetNameFromUID, 130
Size, 130
Write, 130
gdcm::network::ApplicationContext, 145
ApplicationContext, 146
GetName, 146
Print, 146
Read, 146
SetName, 146
Size, 146
Write, 146
gdcm::network::AReleaseRPPDU, 149
AReleaseRPPDU, 150
IsLastFragment, 150
Print, 150
Read, 151
Size, 151
Write, 151
gdcm::network::AReleaseRQPDU, 151
AReleaseRQPDU, 153
IsLastFragment, 153

Print, 153
Read, 153
Size, 153
Write, 153
gdcm::network::ARTIMTimer, 154
ARTIMTimer, 154
GetElapsedTime, 154
GetHasExpired, 154
GetTimeout, 155
SetTimeout, 155
Start, 155
Stop, 155
gdcm::network::AsynchronousOperationsWindowSub, 157
AsynchronousOperationsWindowSub, 157
Print, 158
Read, 158
Size, 158
Write, 158
gdcm::network::BaseCompositeMessage, 225
~BaseCompositeMessage, 227
ConstructPDV, 227
gdcm::network::BaseNormalizedMessage, 227
~BaseNormalizedMessage, 229
ConstructPDV, 229
gdcm::network::BasePDU, 230
~BasePDU, 231
IsLastFragment, 231
Print, 231
Read, 231
Size, 231
Write, 232
gdcm::network::CEchoRQ, 290
AffectedSOPClassUID, 292
ConstructPDV, 292
MessageID, 292
gdcm::network::CEchoRSP, 292
ConstructPDVByDataSet, 293
gdcm::network::CFind, 293
gdcm::network::CFindCancelRQ, 294
ConstructPDVByDataSet, 295
gdcm::network::CFindRQ, 295
ConstructPDV, 296
gdcm::network::CFindRSP, 297
ConstructPDVByDataSet, 298
gdcm::network::CMoveCancelRq, 305
ConstructPDVByDataSet, 306
gdcm::network::CMoveRQ, 306
ConstructPDV, 307
gdcm::network::CMoveRSP, 307
ConstructPDVByDataSet, 308
gdcm::network::CompositeMessageFactory, 324
ConstructCEchoRQ, 325
ConstructCFindRQ, 325
ConstructCMoveRQ, 325
ConstructCStoreRQ, 325
ConstructCStoreRSP, 325
gdcm::network::CStoreRQ, 361
ConstructPDV, 363
gdcm::network::CStoreRSP, 363
ConstructPDV, 364
gdcm::network::DIMSE, 439
C_CANCEL_RQ, 441
C_ECHO_RQ, 441
C_ECHO_RSP, 441
C_FIND_RQ, 440
C_FIND_RSP, 441
C_GET_RQ, 440
C_GET_RSP, 440
C_MOVE_RQ, 441
C_MOVE_RSP, 441
C_STORE_RQ, 440
C_STORE_RSP, 440
CommandTypes, 440
N_ACTION_RQ, 441
N_ACTION_RSP, 441
N_CREATE_RQ, 441
N_CREATE_RSP, 441
N_DELETE_RQ, 441
N_DELETE_RSP, 441
N_EVENT_REPORT_RQ, 441
N_EVENT_REPORT_RSP, 441
N_GET_RQ, 441
N_GET_RSP, 441
N_SET_RQ, 441
N_SET_RSP, 441
gdcm::network::ImplementationClassUIDSub, 696
ImplementationClassUIDSub, 697
Print, 697
Read, 697
Size, 697
Write, 697
gdcm::network::ImplementationUIDSub, 698
ImplementationUIDSub, 698
Write, 698
gdcm::network::ImplementationVersionNameSub, 698
ImplementationVersionNameSub, 699
Print, 699
Read, 699
Size, 699
Write, 699
gdcm::network::MaximumLengthSub, 790
GetMaximumLength, 791
MaximumLengthSub, 791
Print, 791
Read, 791
SetMaximumLength, 791
Size, 791

Write, 791
gdcm::network::NActionRQ, 845
ConstructPDV, 846
gdcm::network::NActionRSP, 846
ConstructPDVByDataSet, 847
gdcm::network::NCreateRQ, 848
ConstructPDV, 849
gdcm::network::NCreateRSP, 849
ConstructPDVByDataSet, 850
gdcm::network::NDeleteRQ, 851
ConstructPDV, 852
gdcm::network::NDeleteRSP, 852
ConstructPDVByDataSet, 853
gdcm::network::NEventReportRQ, 857
ConstructPDV, 858
gdcm::network::NEventReportRSP, 859
ConstructPDVByDataSet, 860
gdcm::network::NGetRQ, 860
ConstructPDV, 861
gdcm::network::NGetRSP, 862
ConstructPDVByDataSet, 863
gdcm::network::NormalizedMessageFactory, 864
ConstructNAction, 864
ConstructNCreate, 864
ConstructNDelete, 865
ConstructNEventReport, 865
ConstructNGet, 865
ConstructNSet, 865
gdcm::network::NSetRQ, 868
ConstructPDV, 869
gdcm::network::NSetRSP, 869
ConstructPDVByDataSet, 870
gdcm::network::PDataTFPDU, 904
AddPresentationDataValue, 906
GetNumberOfPresentationDataValues, 906
GetPresentationDataValue, 906
IsLastFragment, 906
PDataTFPDU, 906
Print, 906
Read, 906
ReadInto, 907
Size, 907
SizeType, 906
Write, 907
gdcm::network::PDUFactory, 916
ConstructAbortPDU, 917
ConstructPDU, 917
ConstructReleasePDU, 917
CreateCEchoPDU, 917
CreateCFindPDU, 917
CreateCMovePDU, 917
CreateCStoreRQPDU, 917
CreateCStoreRSPPDU, 918
CreateNActionPDU, 918
CreateNCreatePDU, 918
CreateNDeletePDU, 918
CreateNEventReportPDU, 918
CreateNGetPDU, 918
CreateNSetPDU, 918
DetermineEventByPDU, 919
GetPDVs, 919
gdcm::network::PresentationContextAC, 970
GetPresentationContextID, 971
GetReason, 971
GetTransferSyntax, 971
PresentationContextAC, 971
Print, 971
Read, 972
SetPresentationContextID, 972
SetReason, 972
SetTransferSyntax, 972
Size, 972
Write, 972
gdcm::network::PresentationContextRQ, 976
AddTransferSyntax, 977
GetAbstractSyntax, 977
GetNumberOfTransferSyntaxes, 977
GetPresentationContextID, 977
GetTransferSyntax, 978
GetTransferSyntaxes, 978
operator==, 978
PresentationContextRQ, 977
Print, 978
Read, 978
SetAbstractSyntax, 978
SetPresentationContextID, 978
Size, 979
SizeType, 976
Write, 979
gdcm::network::PresentationDataValue, 979
ConcatenatePDVBlobs, 980
ConcatenatePDVBlobsAsExplicit, 980
GetBlob, 980
GetIsCommand, 980
GetIsLastFragment, 980
GetMessageHeader, 981
GetPresentationContextID, 981
PresentationDataValue, 980
Print, 981
Read, 981
ReadInto, 981
SetBlob, 981
SetCommand, 981
SetDataSet, 981
SetLastFragment, 982
SetMessageHeader, 982
SetPresentationContextID, 982
Size, 982

Write, 982
gdcm::network::RoleSelectionSub, 1046
Print, 1047
Read, 1047
RoleSelectionSub, 1046
SetTuple, 1047
Size, 1047
Write, 1047
gdcm::network::ServiceClassApplicationInformation, 1112
Print, 1113
Read, 1113
ServiceClassApplicationInformation, 1112
SetTuple, 1113
Size, 1113
Write, 1113
gdcm::network::SOPClassExtendedNegociationSub, 1138
Print, 1138
Read, 1138
SetTuple, 1138
Size, 1139
SOPClassExtendedNegociationSub, 1138
Write, 1139
gdcm::network::TableRow, 1243
~TableRow, 1244
TableRow, 1244
transitions, 1244
gdcm::network::TransferSyntaxSub, 1274
GetName, 1275
operator==, 1275
Print, 1275
Read, 1275
SetName, 1275
SetNameFromUID, 1276
Size, 1276
TransferSyntaxSub, 1275
Write, 1276
gdcm::network::Transition, 1276
~Transition, 1277
mAction, 1278
MakeNew, 1278
mEnd, 1278
Transition, 1277
gdcm::network::ULAction, 1302
~ULAction, 1304
operator=, 1305
PerformAction, 1305
ULAction, 1304
gdcm::network::ULActionAA1, 1306
PerformAction, 1307
gdcm::network::ULActionAA2, 1307
PerformAction, 1308
gdcm::network::ULActionAA3, 1308
PerformAction, 1309
gdcm::network::ULActionAA4, 1310
PerformAction, 1311
gdcm::network::ULActionAA5, 1311
PerformAction, 1312
gdcm::network::ULActionAA6, 1312
PerformAction, 1313
gdcm::network::ULActionAA7, 1314
PerformAction, 1315
gdcm::network::ULActionAA8, 1315
PerformAction, 1316
gdcm::network::ULActionAE1, 1316
PerformAction, 1317
gdcm::network::ULActionAE2, 1318
PerformAction, 1319
gdcm::network::ULActionAE3, 1319
PerformAction, 1320
gdcm::network::ULActionAE4, 1320
PerformAction, 1321
gdcm::network::ULActionAE5, 1322
PerformAction, 1323
gdcm::network::ULActionAE6, 1323
PerformAction, 1324
gdcm::network::ULActionAE7, 1324
PerformAction, 1325
gdcm::network::ULActionAE8, 1326
PerformAction, 1327
gdcm::network::ULActionAR1, 1327
PerformAction, 1328
gdcm::network::ULActionAR10, 1328
PerformAction, 1329
gdcm::network::ULActionAR2, 1330
PerformAction, 1331
gdcm::network::ULActionAR3, 1331
PerformAction, 1332
gdcm::network::ULActionAR4, 1332
PerformAction, 1333
gdcm::network::ULActionAR5, 1334
PerformAction, 1335
gdcm::network::ULActionAR6, 1335
PerformAction, 1336
gdcm::network::ULActionAR7, 1336
PerformAction, 1337
gdcm::network::ULActionAR8, 1338
PerformAction, 1339
gdcm::network::ULActionAR9, 1339
PerformAction, 1340
gdcm::network::ULActionDT1, 1340
PerformAction, 1341
gdcm::network::ULActionDT2, 1342
PerformAction, 1343
gdcm::network::ULBasicCallback, 1343
~ULBasicCallback, 1345
GetDataSets, 1345
GetResponses, 1345

HandleDataSet, 1345
HandleResponse, 1345
ULBasicCallback, 1345
gdcm::network::ULConnection, 1346
~ULConnection, 1347
AddAcceptedPresentationContext, 1347
FindContext, 1347
GetAcceptedPresentationContexts, 1348
GetConnectionInfo, 1348
GetMaxPDUSize, 1348
GetPresentationContextACByID, 1348
GetPresentationContextIDFromPresentationContext, 1348
GetPresentationContextRQByID, 1348
GetPresentationContexts, 1348
GetProtocol, 1349
GetState, 1349
GetTimer, 1349
InitializeConnection, 1349
InitializeIncomingConnection, 1349
operator=, 1349
SetMaxPDUSize, 1349
SetPresentationContexts, 1349, 1350
SetState, 1350
StopProtocol, 1350
ULActionAE6, 1350
ULConnection, 1347
ULConnectionManager, 1350
gdcm::network::ULConnectionCallback, 1351
~ULConnectionCallback, 1352
DataSetHandled, 1352
DataSetHandles, 1352
HandleDataSet, 1352
HandleResponse, 1352
mImplicit, 1353
ResetHandledDataSet, 1352
SetImplicitFlag, 1352
ULConnectionCallback, 1352
gdcm::network::ULConnectionInfo, 1353
GetCalledAETitle, 1354
GetCalledComputerName, 1354
GetCalledIPAddress, 1354
GetCalledIPPort, 1354
GetCallingAETitle, 1354
GetMaxPDULength, 1354
Initialize, 1354
SetMaxPDULength, 1354
ULConnectionInfo, 1354
gdcm::network::ULConnectionManager, 1355
~ULConnectionManager, 1358
BreakConnection, 1358
BreakConnectionNow, 1358
EstablishConnection, 1358
EstablishConnectionMove, 1359
mConnection, 1362
mSecondaryConnection, 1362
mTransitions, 1362
RunEventLoop, 1359
RunMoveEventLoop, 1359
SendEcho, 1359
SendFind, 1359, 1360
SendMove, 1360
SendNAction, 1360
SendNCreate, 1360
SendNDelete, 1361
SendNEventReport, 1361
SendNGet, 1361
SendNSet, 1361
SendStore, 1362
ULConnectionManager, 1358
gdcm::network::ULEvent, 1363
~ULEvent, 1363
GetDataSetPos, 1364
GetEvent, 1364
GetIStream, 1364
GetPDUs, 1364
SetEvent, 1364
SetPDU, 1364
ULEvent, 1363
gdcm::network::ULTransitionTable, 1364
HandleEvent, 1365
PrintTable, 1365
ULTransitionTable, 1365
gdcm::network::ULWritingCallback, 1366
~ULWritingCallback, 1367
HandleDataSet, 1367
HandleResponse, 1367
SetDirectory, 1367
ULWritingCallback, 1367
gdcm::network::UserInformation, 1381
~UserInformation, 1382
AddRoleSelectionSub, 1382
AddSOPClassExtendedNegociationSub, 1382
GetMaximumLengthSub, 1382
operator=, 1382
Print, 1383
Read, 1383
Size, 1383
UserInformation, 1382
Write, 1383
gdcm::NoEvent, 863
gdcm::NormalizedNetworkFunctions, 865
ConstructQuery, 866
NAction, 866
NCreate, 867
NDelete, 867
NEventReport, 867
NGet, 867

NSet, 867
gdcm::Object, 871
~Object, 872
Object, 872
operator<<, 873
operator=, 873
Print, 873
Register, 873
SmartPointer, 873
UnRegister, 873
gdcm::OpenSSLCryptoFactory, 874
CreateCMSProvider, 876
InitOpenSSL, 876
OpenSSLCryptoFactory, 875
gdcm::OpenSSLCryptographicMessageSyntax, 876
~OpenSSLCryptographicMessageSyntax, 878
Decrypt, 878
Encrypt, 878
GetCipherType, 878
OpenSSLCryptographicMessageSyntax, 878
ParseCertificateFile, 878
ParseKeyFile, 879
SetCipherType, 879
SetPassword, 879
gdcm::OpenSSLP7CryptoFactory, 880
CreateCMSProvider, 881
OpenSSLP7CryptoFactory, 881
gdcm::OpenSSLP7CryptographicMessageSyntax, 882
~OpenSSLP7CryptographicMessageSyntax, 883
Decrypt, 883
Encrypt, 883
GetCipherType, 884
OpenSSLP7CryptographicMessageSyntax, 883
ParseCertificateFile, 884
ParseKeyFile, 884
SetCipherType, 884
SetPassword, 884
gdcm::Orientation, 885
~Orientation, 886
AXIAL, 886
CORONAL, 886
GetLabel, 887
GetMajorAxisFromPatientRelativeDirectionCosine, 887
GetObliquityThresholdCosineValue, 887
GetType, 887
OBLIQUE, 886
operator<<, 888
Orientation, 886
OrientationType, 886
Print, 887
SAGITTAL, 886
SetObliquityThresholdCosineValue, 887
UNKNOWN, 886
gdcm::Overlay, 888
~Overlay, 891
Decompress, 892
GetBitPosition, 892
GetBitsAllocated, 892
GetColumns, 892
GetDescription, 892
GetGroup, 892
GetOrigin, 893
GetOverlayData, 893
GetOverlayTypeAsString, 893
GetOverlayTypeFromString, 893
GetRows, 893
GetType, 893
GetTypeAsEnum, 893
GetUnpackBuffer, 894
GetUnpackBufferLength, 894
GrabOverlayFromPixelData, 894
Graphics, 891
Invalid, 891
IsEmpty, 894
IsInPixelData, 894
IsZero, 895
operator=, 895
Overlay, 891, 892
OverlayType, 891
Print, 895
ROI, 891
SetBitPosition, 895
SetBitsAllocated, 895
SetColumns, 895
SetDescription, 896
SetFrameOrigin, 896
SetGroup, 896
SetNumberOfFrames, 896
SetOrigin, 896
SetOverlay, 896
SetRows, 897
SetType, 897
Update, 897
gdcm::ParseException, 898
~ParseException, 899
GetLastElement, 900
operator=, 900
ParseException, 899
SetLastElement, 900
gdcm::Parser, 900
~Parser, 902
DuplicateAttributeError, 902
EndElementHandler, 901
ErrorType, 902
GetBuffer, 902
GetCurrentByteIndex, 902
GetErrorCode, 902

GetErrorString, 903
GetUserData, 903
JunkAfterDocElementError, 902
NoElementsError, 902
NoError, 902
NoMemoryError, 902
Parse, 903
ParseBuffer, 903
Parser, 902
Process, 903
SetElementHandler, 903
SetUserData, 903
StartElementHandler, 901
SyntaxError, 902
TagMismatchError, 902
UndefinedEntityError, 902
UnexpectedStateError, 902
gdcm::Patient, 904
Patient, 904
gdcm::PDBElement, 908
GetName, 909
GetValue, 909
NameField, 910
operator<<, 910
operator==, 909
PDBElement, 909
SetName, 909
SetValue, 910
ValueField, 910
gdcm::PDBHeader, 911
~PDBHeader, 912
FindPDBElementByName, 912
GetPDBEEnd, 912
GetPDBElementByName, 912
GetPDBInfoTag, 912
LoadFromDataElement, 912
operator<<, 913
PDBHeader, 912
Print, 913
gdcm::PDFCodec, 913
~PDFCodec, 915
CanCode, 915
CanDecode, 915
Decode, 916
PDFCodec, 915
gdcm::PersonName, 919
Component, 921
GetMaxLength, 920
GetNumberOfComponents, 920
MaxLength, 921
MaxNumberOfComponents, 921
Padding, 921
Print, 920
Separator, 921
SetBlob, 920
SetComponents, 920, 921
gdcm::PGXCodec, 922
~PGXCodec, 925
CanCode, 925
CanDecode, 925
Clone, 925
GetHeaderInfo, 925
PGXCodec, 925
Read, 926
Write, 926
gdcm::PhotometricInterpretation, 926
ARGB, 928
CMYK, 928
GetPIString, 928
GetPIType, 928
GetSamplesPerPixel, 928
GetString, 929
GetType, 929
HSV, 928
IsLossless, 929
IsLossy, 929
IsRetired, 929
IsSameColorSpace, 929
MONOCHROME1, 927
MONOCHROME2, 927
operator PIType, 929
operator<<, 930
PALETTE_COLOR, 928
PhotometricInterpretation, 928
PI_END, 928
PIType, 927
RGB, 928
UNKNOWN, 927
YBR_FULL, 928
YBR_FULL_422, 928
YBR_ICT, 928
YBR_PARTIAL_420, 928
YBR_PARTIAL_422, 928
YBR_RCT, 928
gdcm::PixelFormat, 930
Bitmap, 938
FLOAT16, 933
FLOAT32, 933
FLOAT64, 933
GetBitsAllocated, 933
GetBitsStored, 933
GetHighBit, 934
GetMax, 934
GetMin, 934
GetPixelRepresentation, 934
GetPixelSize, 934
GetSamplesPerPixel, 934
GetScalarType, 935

GetScalarTypeAsString, 935
INT12, 932
INT16, 932
INT32, 932
INT64, 933
INT8, 932
IsCompatible, 935
IsValid, 935
operator ScalarType, 935
operator!=, 936
operator<<, 938
operator==, 936
PixelFormat, 933
Print, 936
ScalarType, 932
SetBitsAllocated, 936
SetBitsStored, 937
SetHighBit, 937
SetPixelRepresentation, 937
SetSamplesPerPixel, 937
SetScalarType, 937
SINGLEBIT, 933
UINT12, 932
UINT16, 932
UINT32, 932
UINT64, 932
UINT8, 932
UNKNOWN, 933
Validate, 938
gdcm::Pixmap, 939
~Pixmap, 943
AreOverlaysInPixelData, 943
Curves, 945
GetCurve, 943
GetIconImage, 943
GetNumberOfCurves, 944
GetNumberOfOverlays, 944
GetOverlay, 944
Icon, 945
Overlays, 946
Pixmap, 943
Print, 944
RemoveOverlay, 944
SetIconImage, 945
SetNumberOfCurves, 945
SetNumberOfOverlays, 945
UnusedBitsPresentInPixelData, 945
gdcm::PixmapReader, 946
~PixmapReader, 949
GetPixmap, 949
PixelData, 950
PixmapReader, 949
Read, 949
ReadACRNEMAImage, 949
ReadImage, 950
ReadImageInternal, 950
gdcm::PixmapToPixmapFilter, 950
~PixmapToPixmapFilter, 952
GetInput, 952
GetOutput, 952
GetOutputAsPixmap, 952
PixmapToPixmapFilter, 952
gdcm::PixmapWriter, 953
~PixmapWriter, 956
DoIconImage, 956
GetImage, 956
GetPixmap, 956
PixelData, 958
PixmapWriter, 956
PrepareWrite, 957
SetImage, 957
SetPixmap, 957
Write, 957
gdcm::PNMCodec, 958
~PNMCodec, 961
CanCode, 961
CanDecode, 961
Clone, 962
GetBufferLength, 962
GetHeaderInfo, 962
PNMCodec, 961
Read, 962
SetBufferLength, 962
Write, 962
gdcm::Preamble, 963
~Preamble, 964
Clear, 964
Create, 964
GetInternal, 965
GetLength, 965
IsEmpty, 965
IsValid, 965
operator<<, 966
operator=, 965
Preamble, 964
Print, 965
Read, 965
Remove, 966
Valid, 966
Write, 966
gdcm::PresentationContext, 967
AbstractSyntax, 970
AddTransferSyntax, 969
GetAbstractSyntax, 969
GetNumberOfTransferSyntaxes, 969
GetPresentationContextID, 969
GetTransferSyntax, 969
ID, 970

operator==, 969
PresentationContext, 968
Print, 969
SetAbstractSyntax, 969
SetPresentationContextID, 970
SizeType, 968
TransferSyntaxArrayType, 968
TransferSyntaxes, 970
gdcm::PresentationContextGenerator, 973
AddFromFile, 974
AddPresentationContext, 974
GenerateFromFilenames, 974
GenerateFromUID, 974
GetDefaultTransferSyntax, 975
GetPresentationContexts, 975
PresentationContextArrayType, 974
PresentationContextGenerator, 974
SetDefaultTransferSyntax, 975
SetMergeModeToAbstractSyntax, 975
SetMergeModeToTransferSyntax, 975
SizeType, 974
gdcm::Printer, 983
~Printer, 985
CONDENSED_STYLE, 985
CXX, 985
F, 987
GetPrintStyle, 985
MaxPrintLength, 987
Print, 985
PrintDataElement, 985
PrintDataSet, 986
Printer, 985
PrintSQ, 986
PrintStyle, 987
PrintStyles, 985
SetColor, 986
SetFile, 986
SetStyle, 986
VERBOSE_STYLE, 985
XML, 985
gdcm::PrivateDict, 987
~PrivateDict, 988
AddDictEntry, 988
Dicts, 989
FindDictEntry, 988
GetDictEntry, 988
IsEmpty, 989
LoadDefault, 989
operator<<, 989
PrintXML, 989
PrivateDict, 988
RemoveDictEntry, 989
gdcm::PrivateTag, 990
GetAsDataElement, 993
GetOwner, 993
operator!=, 993
operator<, 994
operator<<, 995
operator=, 994
operator==, 994
PrivateTag, 993
ReadFromCommaSeparatedString, 994
SetOwner, 994
gdcm::ProgressEvent, 995
~ProgressEvent, 997
CheckEvent, 998
GetEventName, 998
GetProgress, 998
MakeObject, 998
operator=, 998
ProgressEvent, 997
Self, 997
SetProgress, 998
Superclass, 997
gdcm::PVRGCodec, 999
~PVRGCodec, 1002
CanCode, 1002
CanDecode, 1002
Clone, 1002
Code, 1003
Decode, 1003
PVRGCodec, 1002
SetLossyFlag, 1003
gdcm::PythonFilter, 1003
~PythonFilter, 1004
GetFile, 1004
PythonFilter, 1004
SetDicts, 1004
SetFile, 1004
ToPyObject, 1005
UseDictAlways, 1005
gdcm::QueryBase, 1005
~QueryBase, 1006
GetAllRequiredTags, 1006
GetAllTags, 1006
GetHierachicalSearchTags, 1006
GetName, 1007
GetOptionalTags, 1007
GetQueryLevel, 1007
GetRequiredTags, 1007
GetUniqueTags, 1007
gdcm::QueryFactory, 1008
GetCharacterFromCurrentLocale, 1008
ListCharSets, 1008
ProduceCharacterSetDataElement, 1008
ProduceQuery, 1009
gdcm::QueryImage, 1009
GetHierachicalSearchTags, 1010

GetName, 1010
GetOptionalTags, 1011
GetQueryLevel, 1011
GetRequiredTags, 1011
GetUniqueTags, 1011
gdcm::QueryPatient, 1012
GetHierachicalSearchTags, 1013
GetName, 1013
GetOptionalTags, 1013
GetQueryLevel, 1013
GetRequiredTags, 1013
GetUniqueTags, 1014
gdcm::QuerySeries, 1014
GetHierachicalSearchTags, 1015
GetName, 1015
GetOptionalTags, 1016
GetQueryLevel, 1016
GetRequiredTags, 1016
GetUniqueTags, 1016
gdcm::QueryStudy, 1017
GetHierachicalSearchTags, 1018
GetName, 1018
GetOptionalTags, 1018
GetQueryLevel, 1018
GetRequiredTags, 1018
GetUniqueTags, 1019
gdcm::RAWCodec, 1019
~RAWCodec, 1022
CanCode, 1022
CanDecode, 1022
Clone, 1022
Code, 1023
Decode, 1023
DecodeByStreams, 1023
DecodeBytes, 1023
GetHeaderInfo, 1023
RAWCodec, 1022
gdcm::Reader, 1024
~Reader, 1027
CanRead, 1027
F, 1031
GetFile, 1027
GetStreamCurrentPosition, 1027
GetStreamPtr, 1028
Read, 1028
ReadDataSet, 1028
Reader, 1027
ReadMetaInformation, 1028
ReadPreamble, 1028
ReadSelectedPrivateTags, 1029
ReadSelectedTags, 1029
ReadUpToTag, 1029
SetFile, 1029
SetFileName, 1029
SetStream, 1030
StreamImageReader, 1030
gdcm::RealWorldValueMappingContent, 1031
CodeMeaning, 1032
CodeValue, 1032
RealWorldValueIntercept, 1032
RealWorldValueSlope, 1032
gdcm::Region, 1032
~Region, 1033
Area, 1033
Clone, 1033
ComputeBoundingBox, 1034
Empty, 1034
IsValid, 1034
Print, 1034
Region, 1033
gdcm::Rescaler, 1035
~Rescaler, 1036
ComputeInterceptSlopePixelType, 1036
ComputePixelTypeFromMinMax, 1036
GetIntercept, 1037
GetSlope, 1037
InverseRescale, 1037
InverseRescaleFunctionIntoBestFit, 1037
Rescale, 1037
RescaleFunctionIntoBestFit, 1037
Rescaler, 1036
SetIntercept, 1038
SetMinMaxForPixelType, 1038
SetPixelFormat, 1038
SetSlope, 1038
SetTargetPixelType, 1038
SetUseTargetPixelType, 1039
gdcm::RLECodec, 1039
~RLECodec, 1042
AppendFrameEncode, 1043
AppendRowEncode, 1043
CanCode, 1043
CanDecode, 1043
Clone, 1043
Code, 1043
Decode, 1044
DecodeByStreams, 1044
DecodeExtent, 1044
GetBufferLength, 1044
GetHeaderInfo, 1044
ImageRegionReader, 1046
IsFrameEncoder, 1045
IsRowEncoder, 1045
RLECodec, 1042
SetBufferLength, 1045
SetLength, 1045
StartEncode, 1045
StopEncode, 1045

gdcm::Scanner, 1048
~Scanner, 1052
AddPrivateTag, 1052
AddSkipTag, 1052
AddTag, 1052
Begin, 1053
ClearSkipTags, 1053
ClearTags, 1053
ConstIterator, 1051
End, 1053
GetAllFilenamesFromTagToValue, 1053
GetFilenameFromTagToValue, 1053
GetFilenames, 1053
GetKeys, 1054
GetMapping, 1054
GetMappingFromTagToValue, 1054
GetMappings, 1054
GetOrderedValues, 1054
GetValue, 1054
GetValues, 1055
IsKey, 1055
MappingType, 1051
New, 1055
operator<<, 1057
Print, 1056
PrintTable, 1056
ProcessPublicTag, 1056
Scan, 1056
Scanner, 1052
TagToValue, 1051
TagToValueValueType, 1051
ValuesType, 1052
gdcm::Scanner2, 1057
~Scanner2, 1062
AddPrivateTag, 1062
AddPublicTag, 1062
AddSkipTag, 1062
Begin, 1062
ClearPrivateTags, 1062
ClearPublicTags, 1063
ClearSkipTags, 1063
End, 1063
GetAllFilenamesFromPrivateTagToValue, 1063
GetAllFilenamesFromPublicTagToValue, 1063
GetFilenameFromPrivateTagToValue, 1063
GetFilenameFromPublicTagToValue, 1063
GetFilenames, 1063
GetKeys, 1064
GetMappingFromPrivateTagToValue, 1064
GetMappingFromPublicTagToValue, 1064
GetPrivateMapping, 1064
GetPrivateMappings, 1064
GetPrivateOrderedValues, 1064
GetPrivateValue, 1064
GetPrivateValues, 1065
GetPublicMapping, 1065
GetPublicMappings, 1065
GetPublicOrderedValues, 1065
GetPublicValue, 1065
GetPublicValues, 1065
GetValues, 1066
IsKey, 1066
New, 1066
operator<<, 1067
Print, 1066
PrintTable, 1066
PrivateBegin, 1066
PrivateConstIterator, 1061
PrivateEnd, 1067
PrivateMappingType, 1061
PrivateTagToValue, 1061
PrivateTagToValueValueType, 1061
ProcessPrivateTag, 1067
ProcessPublicTag, 1067
PublicConstIterator, 1061
PublicMappingType, 1061
PublicTagToValue, 1061
PublicTagToValueValueType, 1061
Scan, 1067
Scanner2, 1062
ValuesType, 1061
gdcm::Scanner2::ltstr, 783
operator(), 784
gdcm::Scanner::ltstr, 784
operator(), 784
gdcm::Segment, 1068
~Segment, 1071
AddSurface, 1071
ALGOType, 1071
ALGOType_END, 1071
AnatomicRegion, 1075
AnatomicRegionModifiers, 1075
AUTOMATIC, 1071
BasicCodedEntryVector, 1070
GetALGOType, 1071
GetALGOTypeString, 1071
GetAnatomicRegion, 1071, 1072
GetAnatomicRegionModifiers, 1072
GetPropertyCategory, 1072
GetPropertyType, 1072
GetPropertyTypeModifiers, 1072
GetSegmentAlgorithmName, 1072
GetSegmentAlgorithmType, 1073
GetSegmentDescription, 1073
GetSegmentLabel, 1073
GetSegmentNumber, 1073
GetSurface, 1073
GetSurfaceCount, 1073

GetSurfaces, 1073
MANUAL, 1071
PropertyCategory, 1075
PropertyType, 1075
PropertyTypeModifiers, 1075
Segment, 1071
SegmentAlgorithmName, 1076
SegmentAlgorithmType, 1076
SegmentDescription, 1076
SegmentLabel, 1076
SegmentNumber, 1076
SEMIAUTOMATIC, 1071
SetAnatomicRegion, 1073
SetAnatomicRegionModifiers, 1074
SetPropertyCategory, 1074
SetPropertyType, 1074
SetPropertyTypeModifiers, 1074
SetSegmentAlgorithmName, 1074
SetSegmentAlgorithmType, 1074
SetSegmentDescription, 1074
SetSegmentLabel, 1075
SetSegmentNumber, 1075
SetSurfaceCount, 1075
SurfaceCount, 1076
Surfaces, 1076
SurfaceVector, 1070
gdcm::SegmentedPaletteColorLookupTable, 1077
~SegmentedPaletteColorLookupTable, 1079
Print, 1080
SegmentedPaletteColorLookupTable, 1079
SetLUT, 1080
gdcm::SegmentHelper, 109
gdcm::SegmentHelper::BasicCodedEntry, 242
BasicCodedEntry, 244
CM, 245
CSD, 245
CSV, 245
CV, 245
IsEmpty, 245
gdcm::SegmentReader, 1080
~SegmentReader, 1083
GetSegments, 1083
Read, 1083
ReadSegment, 1084
ReadSegments, 1084
SegmentMap, 1083
SegmentReader, 1083
Segments, 1084
SegmentVector, 1083
gdcm::SegmentWriter, 1084
~SegmentWriter, 1088
AddSegment, 1088
GetNumberOfSegments, 1088
GetSegment, 1088
GetSegments, 1088
PrepareWrite, 1088
Segments, 1089
SegmentVector, 1088
SegmentWriter, 1088
SetNumberOfSegments, 1089
SetSegments, 1089
Write, 1089
gdcm::SequenceOfFragments, 1090
AddFragment, 1093
Begin, 1093
Clear, 1093
ComputeByteLength, 1093
ComputeLength, 1094
ConstIterator, 1092
End, 1094
FragmentVector, 1092
GetBuffer, 1094
GetFragBuffer, 1094
GetFragment, 1094
GetLength, 1094
GetNumberOfFragments, 1095
GetTable, 1095
Iterator, 1092
New, 1095
operator==, 1095
Print, 1095
Read, 1096
ReadPreValue, 1096
ReadValue, 1096
SequenceOfFragments, 1093
SetLength, 1096
SizeType, 1092
Write, 1096
WriteBuffer, 1097
gdcm::SequenceOfItems, 1097
AddItem, 1101
AddNewUndefinedLengthItem, 1101
Begin, 1101, 1102
Clear, 1102
ComputeLength, 1102
ConstIterator, 1100
End, 1102
FindDataElement, 1102
GetItem, 1102, 1103
GetLength, 1103
GetNumberOfItems, 1103
IsEmpty, 1103
IsUndefinedLength, 1103
Items, 1106
ItemVector, 1100
Iterator, 1100
New, 1104
operator=, 1104

operator==, 1104
Print, 1104
Read, 1104
RemoveItemByIndex, 1105
SequenceLengthField, 1106
SequenceOfItems, 1101
SetLength, 1105
SetLengthToUndefined, 1105
SetNumberOfItems, 1105
SizeType, 1101
Write, 1105
gdcm::SerieHelper, 1106
~SerieHelper, 1109
AddFile, 1109
AddFileName, 1109
AddRestriction, 1109
Clear, 1109
CreateDefaultUniqueSeriesIdentifier, 1110
CreateUniqueSeriesIdentifier, 1110
FileNameOrdering, 1110
GetFirstSingleSerieUIDFileSet, 1110
GetNextSingleSerieUIDFileSet, 1110
ImageNumberOrdering, 1110
ImagePositionPatientOrdering, 1110
ItFileSetHt, 1111
OrderFileList, 1110
Rule, 1108
SerieHelper, 1109
SerieRestrictions, 1108
SetDirectory, 1110
SetLoadMode, 1111
SetUseSeriesDetails, 1111
SingleSerieUIDFileSetHT, 1111
SingleSerieUIDFileSetmap, 1108
UserOrdering, 1111
gdcm::Series, 1111
Series, 1112
gdcm::ServiceClassUser, 1114
~ServiceClassUser, 1117
GetAETitle, 1117
GetCalledAETitle, 1117
GetTimeout, 1117
InitializeConnection, 1117
IsPresentationContextAccepted, 1118
New, 1118
operator=, 1118
SendEcho, 1118
SendFind, 1118
SendMove, 1118, 1119
SendStore, 1119
ServiceClassUser, 1117
SetAETitle, 1120
SetCalledAETitle, 1120
SetHostname, 1120
SetPort, 1120
SetPortSCP, 1120
SetPresentationContexts, 1121
SetTimeout, 1121
StartAssociation, 1121
StopAssociation, 1121
gdcm::SHA1, 1122
~SHA1, 1123
Compute, 1123
ComputeFile, 1123
operator=, 1123
SHA1, 1123
gdcm::SimpleMemberCommand< T >, 1124
~SimpleMemberCommand, 1127
Execute, 1128
m_MemberFunction, 1129
m_This, 1129
New, 1128
operator=, 1128
Self, 1127
SetCallbackFunction, 1128
SimpleMemberCommand, 1127
TMemberFunctionPointer, 1127
gdcm::SimpleSubjectWatcher, 1129
~SimpleSubjectWatcher, 1130
EndFilter, 1130
operator=, 1130
ShowAbort, 1130
ShowAnonymization, 1131
ShowData, 1131
ShowDataSet, 1131
ShowFileName, 1131
ShowIteration, 1131
ShowProgress, 1131
SimpleSubjectWatcher, 1130
StartFilter, 1131
TestAbortOff, 1132
TestAbortOn, 1132
gdcm::SmartPointer< ObjectType >, 1134
~SmartPointer, 1136
GetPointer, 1136
operator ObjectType *, 1136
operator->, 1137
operator=, 1137
operator*, 1137
SmartPointer, 1136
gdcm::SOPClassUIDToIOD, 1139
const, 1140
GetIOD, 1140
GetIODFromSOPClassUID, 1140
GetNumberOfSOPClassToIOD, 1140
GetSOPClassUIDFromIOD, 1140
GetSOPClassUIDToIOD, 1140
GetSOPClassUIDToIODs, 1141

gdcm::Sorter, 1141
~Sorter, 1143
AddSelect, 1143
Filenames, 1145
GetFilenames, 1143
operator<<, 1145
Print, 1143
Selection, 1145
SelectionMap, 1143
SetSortFunction, 1144
SetTagsToRead, 1144
Sort, 1144
Sorter, 1143
SortFunc, 1145
SortFunction, 1143
StableSort, 1144
TagsToRead, 1145
gdcm::Spacing, 1146
~Spacing, 1147
CALIBRATED, 1147
ComputePixelAspectRatioFromPixelSpacing, 1148
DETECTOR, 1147
MAGNIFIED, 1147
Spacing, 1147
SpacingType, 1147
UNKNOWN, 1147
gdcm::Spectroscopy, 1148
Spectroscopy, 1148
gdcm::SplitMosaicFilter, 1149
~SplitMosaicFilter, 1150
ComputeMOSAICDimensions, 1150
ComputeMOSAICImagePositionPatient, 1150
ComputeMOSAICSliceNormal, 1150
ComputeMOSAICSlicePosition, 1150
GetAcquisitionSize, 1150
GetFile, 1151
GetImage, 1151
GetNumberOfImagesInMosaic, 1151
SetFile, 1151
SetImage, 1151
Split, 1151
SplitMosaicFilter, 1150
gdcm::StartEvent, 1152
gdcm::static_assert_test< x >, 1153
gdcm::STATIC_ASSERTION_FAILURE< true >, 1154
value, 1155
gdcm::STATIC_ASSERTION_FAILURE< x >, 1154
gdcm::StreamImageReader, 1155
~StreamImageReader, 1156
CanReadImage, 1157
DefinePixelExtent, 1157
DefineProperBufferLength, 1157
GetDimensionsValueForResolution, 1157
GetFile, 1158
Read, 1158
ReadImageInformation, 1158
SetFileName, 1158
SetStream, 1159
StreamImageReader, 1156
gdcm::StreamImageWriter, 1159
~StreamImageWriter, 1161
CanWriteFile, 1162
DefinePixelExtent, 1162
DefineProperBufferLength, 1162
mElementOffsets, 1164
mElementOffsets1, 1164
mspFile, 1164
mWriter, 1165
mXMax, 1165
mXMin, 1165
mYMax, 1165
mYMin, 1165
mZMax, 1165
mZMin, 1165
SetFile, 1162
SetFileName, 1163
SetStream, 1163
StreamImageWriter, 1161
Write, 1163
WriteImageInformation, 1163
WriteImageSubregionRAW, 1164
WriteRawHeader, 1164
gdcm::StrictScanner, 1166
~StrictScanner, 1170
AddPrivateTag, 1170
AddSkipTag, 1170
AddTag, 1171
Begin, 1171
ClearSkipTags, 1171
ClearTags, 1171
ConstIterator, 1169
End, 1171
GetAllFilenamesFromTagToValue, 1171
GetFilenameFromTagToValue, 1171
GetFilenames, 1172
GetKeys, 1172
GetMapping, 1172
GetMappingFromTagToValue, 1172
GetMappings, 1172
GetOrderedValues, 1172
GetValue, 1173
GetValues, 1173
IsKey, 1173
MappingType, 1169
New, 1173
operator<<, 1175
Print, 1174
PrintTable, 1174

ProcessPublicTag, 1174
Scan, 1174
StrictScanner, 1170
TagToValue, 1169
TagToValueValueType, 1170
ValuesType, 1170
gdcm::StrictScanner2, 1175
~StrictScanner2, 1180
AddPrivateTag, 1180
AddPublicTag, 1180
AddSkipTag, 1180
Begin, 1180
ClearPrivateTags, 1180
ClearPublicTags, 1181
ClearSkipTags, 1181
End, 1181
GetAllFilenamesFromPrivateTagToValue, 1181
GetAllFilenamesFromPublicTagToValue, 1181
GetFilenameFromPrivateTagToValue, 1181
GetFilenameFromPublicTagToValue, 1181
GetFilenames, 1181
GetKeys, 1182
GetMappingFromPrivateTagToValue, 1182
GetMappingFromPublicTagToValue, 1182
GetPrivateMapping, 1182
GetPrivateMappings, 1182
GetPrivateOrderedValues, 1182
GetPrivateValue, 1182
GetPrivateValues, 1183
GetPublicMapping, 1183
GetPublicMappings, 1183
GetPublicOrderedValues, 1183
GetPublicValue, 1183
GetPublicValues, 1183
GetValues, 1184
IsKey, 1184
New, 1184
operator<<, 1185
Print, 1184
PrintTable, 1184
PrivateBegin, 1184
PrivateConstIterator, 1179
PrivateEnd, 1185
PrivateMappingType, 1179
PrivateTagToValue, 1179
PrivateTagToValueValueType, 1179
ProcessPrivateTag, 1185
ProcessPublicTag, 1185
PublicConstIterator, 1179
PublicMappingType, 1179
PublicTagToValue, 1179
PublicTagToValueValueType, 1179
Scan, 1185
StrictScanner2, 1180
ValuesType, 1179
gdcm::StrictScanner2::ltstr, 784
operator(), 785
gdcm::StrictScanner::ltstr, 785
operator(), 785
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1186
const_iterator, 1187
const_reference, 1187
const_reverse_iterator, 1188
difference_type, 1188
IsValid, 1189
iterator, 1188
operator const char *, 1189
pointer, 1188
reference, 1188
reverse_iterator, 1188
size_type, 1188
String, 1189
Trim, 1190
Truncate, 1190
value_type, 1188
gdcm::StringFilter, 1190
~StringFilter, 1191
ExecuteQuery, 1192
FromString, 1192
GetFile, 1192
SetDicts, 1192
SetFile, 1192
StringFilter, 1191
ToString, 1193
ToStringPair, 1193, 1194
UseDictAlways, 1194
gdcm::Study, 1194
Study, 1194
gdcm::Subject, 1195
~Subject, 1196
AddObserver, 1197
GetCommand, 1197
HasObserver, 1197
InvokeEvent, 1197
RemoveAllObservers, 1198
RemoveObserver, 1198
Subject, 1196
gdcm::Surface, 1198
~Surface, 1202
GetAlgorithmFamily, 1202
GetAlgorithmName, 1203
GetAlgorithmVersion, 1203
GetAxisOfRotation, 1203
GetCenterOfRotation, 1203
GetFiniteVolume, 1203
GetManifold, 1203
GetMaximumPointDistance, 1203

GetMeanPointDistance, 1203
GetMeshPrimitive, 1204
GetNumberOfSurfacePoints, 1204
GetNumberOfVectors, 1204
GetPointCoordinatesData, 1204
GetPointPositionAccuracy, 1204
GetPointsBoundingBoxCoordinates, 1204
GetProcessingAlgorithm, 1205
GetRecommendedDisplayCIELabValue, 1205
GetRecommendedDisplayGrayscaleValue, 1205
GetRecommendedPresentationOpacity, 1205
GetRecommendedPresentationType, 1205
GetSTATES, 1205
GetSTATESString, 1206
GetSurfaceComments, 1206
GetSurfaceNumber, 1206
GetSurfaceProcessing, 1206
GetSurfaceProcessingDescription, 1206
GetSurfaceProcessingRatio, 1206
GetVectorAccuracy, 1206
GetVectorCoordinateData, 1206
GetVectorDimensionality, 1207
GetVIEWType, 1207
GetVIEWTypeString, 1207
NO, 1201
POINTS, 1202
SetAlgorithmFamily, 1207
SetAlgorithmName, 1207
SetAlgorithmVersion, 1207
SetAxisOfRotation, 1207
SetCenterOfRotation, 1207
SetFiniteVolume, 1208
SetManifold, 1208
SetMaximumPointDistance, 1208
SetMeanPointDistance, 1208
SetMeshPrimitive, 1208
SetNumberOfSurfacePoints, 1208
SetNumberOfVectors, 1208
SetPointCoordinatesData, 1208
SetPointPositionAccuracy, 1209
SetPointsBoundingBoxCoordinates, 1209
SetProcessingAlgorithm, 1209
SetRecommendedDisplayCIELabValue, 1209
SetRecommendedDisplayGrayscaleValue, 1209
SetRecommendedPresentationOpacity, 1209
SetRecommendedPresentationType, 1210
SetSurfaceComments, 1210
SetSurfaceNumber, 1210
SetSurfaceProcessing, 1210
SetSurfaceProcessingDescription, 1210
SetSurfaceProcessingRatio, 1210
SetVectorAccuracy, 1210
SetVectorCoordinateData, 1210
SetVectorDimensionality, 1211
STATES, 1201
STATES_END, 1202
SURFACE, 1202
Surface, 1202
UNKNOWN, 1202
VIEWType, 1202
VIEWType_END, 1202
WIREFRAME, 1202
YES, 1202
gdcm::SurfaceHelper, 1211
ColorArray, 1212
RecommendedDisplayCIELabToRGB, 1212
RGBToRecommendedDisplayCIELab, 1213
RGBToRecommendedDisplayGrayscale, 1214
gdcm::SurfaceReader, 1214
~SurfaceReader, 1218
GetNumberOfSurfaces, 1218
Read, 1218
ReadPointMacro, 1219
ReadSurface, 1219
ReadSurfaces, 1219
SurfaceReader, 1218
gdcm::SurfaceWriter, 1219
~SurfaceWriter, 1223
ComputeNumberOfSurfaces, 1223
GetNumberOfSurfaces, 1223
NumberOfSurfaces, 1224
PrepareWrite, 1224
PrepareWritePointMacro, 1224
SetNumberOfSurfaces, 1224
SurfaceWriter, 1223
Write, 1224
gdcm::SwapCode, 1224
BadBigEndian, 1226
BadLittleEndian, 1226
BigEndian, 1226
GetIndex, 1226
GetSwapCodeString, 1226
LittleEndian, 1226
operator SwapCode::SwapCodeType, 1226
operator<<, 1227
SwapCode, 1226
SwapCodeType, 1225
Unknown, 1226
gdcm::SwapperDoOp, 1227
Swap, 1227
SwapArray, 1227
gdcm::SwapperNoOp, 1228
Swap, 1228
SwapArray, 1228
gdcm::System, 1228
ConvertToUNC, 1230
DeleteDirectory, 1230
EncodeBytes, 1230

FileExists, 1230
FileIsDirectory, 1230
FileIsSymlink, 1231
FileSize, 1231
FileTime, 1231
FormatDateTime, 1231
GetCurrentDateTime, 1232
GetCurrentModuleFileName, 1232
GetCurrentProcessFileName, 1232
GetCurrentResourcesDirectory, 1232
GetCWD, 1232
GetHostName, 1232
GetLastSystemError, 1233
GetLocaleCharset, 1233
GetPermissions, 1233
GetTimezoneOffsetFromUTC, 1233
MakeDirectory, 1233
ParseDateTime, 1233, 1234
RemoveFile, 1234
SetPermissions, 1234
StrCaseCmp, 1234
StrNCaseCmp, 1234
StrSep, 1235
StrTokR, 1235
gdcm::Table, 1235
~Table, 1237
GetTableEntry, 1237
InsertEntry, 1237
MapTableEntry, 1237
operator<<, 1238
operator=, 1238
Table, 1237
TableInternal, 1238
gdcm::TableEntry, 1238
~TableEntry, 1239
TableEntry, 1239
gdcm::TableReader, 1239
~TableReader, 1240
CharacterDataHandler, 1241
EndElement, 1241
GetDefs, 1241
GetFilename, 1241
HandleIOD, 1241
HandleIODEntry, 1241
HandleMacro, 1241
HandleMacroEntry, 1241
HandleMacroEntryDescription, 1242
HandleModule, 1242
HandleModuleEntry, 1242
HandleModuleEntryDescription, 1242
HandleModuleInclude, 1242
Read, 1242
SetFilename, 1242
StartElement, 1242
TableReader, 1240
gdcm::Tag, 1244
bytes, 1254
GetElement, 1247
GetElementTag, 1247
GetGroup, 1248
GetLength, 1248
GetPrivateCreator, 1248
IsGroupLength, 1248
IsGroupXX, 1248
IsIllegal, 1249
IsPrivate, 1249
IsPrivateCreator, 1249
IsPublic, 1249
operator!=, 1250
operator<, 1250
operator<<, 1254
operator<=, 1250
operator>>, 1254
operator=, 1250
operator==, 1250
operator[], 1250, 1251
PrintAsContinuousString, 1251
PrintAsContinuousUpperCaseString, 1251
PrintAsPipeSeparatedString, 1251
Read, 1251
ReadFromCommaSeparatedString, 1251
ReadFromContinuousString, 1252
ReadFromPipeSeparatedString, 1252
SetElement, 1252
SetElementTag, 1252, 1253
SetGroup, 1253
SetPrivateCreator, 1253
Tag, 1247
tag, 1254
tags, 1254
Write, 1253
gdcm::TagPath, 1255
~TagPath, 1255
ConstructFromString, 1256
ConstructFromTagList, 1256
IsValid, 1256
Print, 1256
Push, 1256
TagPath, 1255
gdcm::terminal, 109
Attribute, 110
black, 110
blink, 110
blue, 110
bright, 110
Color, 110
CONSOLE, 110
cyan, 110

dim, 110
green, 110
hidden, 110
magenta, 110
Mode, 110
red, 110
reset, 110
reverse, 110
setattribute, 111
setbgcolor, 111
setfgcolor, 111
setmode, 111
underline, 110
VT100, 110
white, 110
yellow, 110
gdcm::Testing, 1257
~Testing, 1258
ComputeFileMD5, 1259
ComputeMD5, 1259
GetDataExtraRoot, 1259
GetDataRoot, 1259
GetFileName, 1259
GetFileNames, 1260
GetLossyFlagFromFile, 1260
GetMD5DataImage, 1260
GetMD5DataImages, 1260
GetMD5FromBrokenFile, 1260
GetMD5FromFile, 1260
GetMediaStorageDataFile, 1261
GetMediaStorageDataFiles, 1261
GetMediaStorageFromFile, 1261
GetNumberOfFileNames, 1261
GetNumberOfMD5DataImages, 1261
GetNumberOfMediaStorageDataFiles, 1261
GetPixelSpacingDataRoot, 1261
GetSelectedPrivateGroupOffsetFromFile, 1262
GetSelectedTagsOffsetFromFile, 1262
GetSourceDirectory, 1262
GetStreamOffsetFromFile, 1262
GetTempDirectory, 1262
GetTempDirectoryW, 1262
GetTempFilename, 1263
GetTempFilenameW, 1263
MD5DataImagesType, 1258
MediaStorageDataFilesType, 1258
Print, 1263
Testing, 1258
gdcm::Trace, 1263
~Trace, 1265
DebugOff, 1265
DebugOn, 1265
ErrorOff, 1265
ErrorOn, 1265
GetDebugFlag, 1265
GetDebugStream, 1266
GetErrorFlag, 1266
GetErrorStream, 1266
GetStream, 1266
GetWarningFlag, 1266
GetWarningStream, 1266
SetDebug, 1266
SetDebugStream, 1266
SetError, 1267
SetErrorStream, 1267
SetStream, 1267
SetStreamToFile, 1267
SetWarning, 1267
SetWarningStream, 1268
Trace, 1265
WarningOff, 1268
WarningOn, 1268
gdcm::TransferSyntax, 1268
CanStoreLossy, 1272
CT_private_ELE, 1271
DeflatedExplicitVRLittleEndian, 1271
Explicit, 1271
ExplicitVRBigEndian, 1271
ExplicitVRLittleEndian, 1271
GetNegociatedType, 1272
GetString, 1272
GetSwapCode, 1272
GetTSString, 1272
GetTSType, 1273
Implicit, 1271
ImplicitVRBigEndianACRNEMA, 1271
ImplicitVRBigEndianPrivateGE, 1271
ImplicitVRLittleEndian, 1271
IsEncapsulated, 1273
IsEncoded, 1273
IsExplicit, 1273
IsImplicit, 1273
IsLossless, 1273
IsLossy, 1273
IsValid, 1274
JPEG2000, 1271
JPEG2000Lossless, 1271
JPEG2000Part2, 1271
JPEG2000Part2Lossless, 1271
JPEGBaselineProcess1, 1271
JPEGExtendedProcess2_4, 1271
JPEGExtendedProcess3_5, 1271
JPEGFullProgressionProcess10_12, 1271
JPEGLosslessProcess14, 1271
JPEGLosslessProcess14_1, 1271
JPEGLSLossless, 1271
JPEGLSNearLossless, 1271
JPEGSpectralSelectionProcess6_8, 1271

JPIPReferenced, 1271
MPEG2MainProfile, 1271
MPEG2MainProfileHighLevel, 1271
MPEG4AVCH264BDcompatibleHighProfileLevel4_1, 1272
MPEG4AVCH264HighProfileLevel4_1, 1271
NegociatedType, 1270
operator TSType, 1274
operator<<, 1274
RLELossless, 1271
TransferSyntax, 1272
TS_END, 1272
TSType, 1271
Unknown, 1270
WeirdPapryus, 1271
gdcm::Type, 1278
GetTypeString, 1280
GetTypeType, 1280
operator TypeType, 1280
operator<<, 1280
T1, 1279
T1C, 1279
T2, 1279
T2C, 1280
T3, 1280
Type, 1280
TypeType, 1279
UNKNOWN, 1280
gdcm::UI, 1281
Internal, 1281
operator<<, 1281
gdcm::UIDGenerator, 1281
Generate, 1283
GenerateUUID, 1283
GetGDCMUID, 1283
GetRoot, 1283
IsValid, 1283
SetRoot, 1283
UIDGenerator, 1282
gdcm::UIDs, 1284
GetName, 1300
GetNumberOfTransferSyntaxStrings, 1300
GetString, 1301
GetTransferSyntaxString, 1301
GetTransferSyntaxStrings, 1301
GetUIDName, 1301
GetUIDString, 1301
operator TSType, 1301
SetFromUID, 1301
TransferSyntaxStringsType, 1300
TSName, 1300
TSType, 1300
gdcm::UNExplicitDataElement, 1368
GetLength, 1371
Read, 1371
ReadPreValue, 1371
ReadValue, 1371
ReadWithLength, 1371
gdcm::UNExplicitImplicitDataElement, 1372
GetLength, 1375
Read, 1375
ReadPreValue, 1375
ReadValue, 1375
gdcm::Unpacker12Bits, 1376
Pack, 1376
Unpack, 1376
gdcm::Usage, 1377
Conditional, 1378
GetUsageString, 1379
GetUsageType, 1379
Invalid, 1378
Mandatory, 1378
operator UsageType, 1379
operator<<, 1379
Usage, 1378
UsageType, 1378
UserOption, 1378
gdcm::UserEvent, 1380
gdcm::UUIDGenerator, 1383
Generate, 1384
IsValid, 1384
gdcm::Validate, 1384
~Validate, 1385
F, 1386
GetValidatedFile, 1386
SetFile, 1386
V, 1386
Validate, 1385
Validation, 1386
gdcm::Value, 1387
~Value, 1388
Clear, 1389
DataElement, 1390
GetLength, 1389
operator==, 1389
SetLength, 1389
SetLengthOnly, 1389
Value, 1388
gdcm::ValueIO< TDE, TSwap, TType >, 1390
Read, 1390
Write, 1390
gdcm::Version, 1392
~Version, 1392
GetBuildVersion, 1393
GetMajorVersion, 1393
GetMinorVersion, 1393
GetVersion, 1393
operator<<, 1393

Print, 1393
Version, 1392
gdcm::VL, 1394
GetLength, 1395
GetVL16Max, 1395
GetVL32Max, 1396
IsOdd, 1396
IsUndefined, 1396
operator uint32_t, 1396
operator<<, 1398
operator++, 1396
operator+=, 1396
Read, 1397
Read16, 1397
SetToUndefined, 1397
Type, 1395
VL, 1395
Write, 1397
Write16, 1397
gdcm::VM, 1398
Compatible, 1401
GetIndex, 1401
GetLength, 1401
GetNumberOfElementsFromArray, 1402
GetVMString, 1402
GetVMType, 1402
GetVMTypeFromLength, 1402
IsValid, 1402
operator VMType, 1402
operator<<, 1403
VM, 1401
VM0, 1400
VM1, 1400
VM10, 1400
VM12, 1400
VM16, 1400
VM18, 1400
VM1_2, 1400
VM1_3, 1400
VM1_32, 1400
VM1_4, 1400
VM1_5, 1400
VM1_8, 1400
VM1_99, 1400
VM1_n, 1400
VM2, 1400
VM24, 1400
VM256, 1400
VM28, 1400
VM2_2n, 1400
VM2_n, 1401
VM3, 1400
VM30_30n, 1401
VM32, 1400
VM35, 1400
VM3_3n, 1401
VM3_4, 1401
VM3_n, 1401
VM4, 1400
VM47_47n, 1401
VM4_4n, 1401
VM5, 1400
VM6, 1400
VM6_6n, 1401
VM6_n, 1401
VM7_7n, 1401
VM8, 1400
VM9, 1400
VM99, 1400
VM_END, 1401
VMType, 1400
gdcm::VMToLength< T >, 1403
gdcm::VR, 1403
AE, 1405
AS, 1405
AT, 1405
CanDisplay, 1407
Compatible, 1407
CS, 1406
DA, 1406
DS, 1406
DT, 1406
FD, 1406
FL, 1406
GetLength, 1407
GetSize, 1408
GetSizeof, 1408
GetVRString, 1408
GetVRStringFromFile, 1408
GetVRType, 1408
GetVRTypeFromFile, 1408
INVALID, 1405
IS, 1406
IsASCII, 1408
IsASCII2, 1409
IsBinary, 1409
IsBinary2, 1409
IsDual, 1409
IsSwap, 1409
IsValid, 1409
IsVRFile, 1409
LO, 1406
LT, 1406
OB, 1406
OB_OW, 1406
OD, 1406
OF, 1406
OL, 1406

operator VRType, 1410
operator<<, 1410
OV, 1406
OW, 1406
PN, 1406
Read, 1410
SH, 1406
SL, 1406
SQ, 1406
SS, 1406
ST, 1406
SV, 1406
TM, 1406
UC, 1406
UI, 1406
UL, 1406
UN, 1406
UR, 1406
US, 1406
US_OW, 1406
US_SS, 1406
US_SS_OW, 1406
UT, 1406
UV, 1406
VL16, 1406
VL32, 1406
VR, 1407
VR_END, 1407
VR_VM1, 1407
VRALL, 1407
VRASCII, 1406
VRBINARY, 1406
VRType, 1405
Write, 1410
gdcm::VR16ExplicitDataElement, 1411
GetLength, 1413
Read, 1413
ReadPreValue, 1414
ReadValue, 1414
ReadWithLength, 1414
gdcm::VRToEncoding< T >, 1414
gdcm::VRToType< T >, 1415
gdcm::VRVLSize< 0 >, 1416
Read, 1416
Write, 1416
gdcm::VRVLSize< 1 >, 1417
Read, 1418
Write, 1418
gdcm::VRVLSize< T >, 1415
gdcm::Waveform, 1528
Waveform, 1529
gdcm::WLMFindQuery, 1529
GetAbstractSyntaxUID, 1532
GetTagListByLevel, 1532
GetValidDataSet, 1532
InitializeDataSet, 1532
QueryFactory, 1533
ValidateQuery, 1532
WLMFindQuery, 1532
gdcm::Writer, 1533
~Writer, 1536
CheckFileMetaInformationOff, 1536
CheckFileMetaInformationOn, 1536
GetCheckFileMetaInformation, 1536
GetFile, 1536
GetStreamPtr, 1536
Ofstream, 1538
SetCheckFileMetaInformation, 1537
SetFile, 1537
SetFileName, 1537
SetStream, 1537
SetWriteDataSetOnly, 1538
Stream, 1538
StreamImageWriter, 1538
Write, 1538
Writer, 1536
gdcm::XMLDictReader, 1539
~XMLDictReader, 1540
CharacterDataHandler, 1541
EndElement, 1541
GetDict, 1541
HandleDescription, 1541
HandleEntry, 1541
StartElement, 1541
XMLDictReader, 1540
gdcm::XMLPrinter, 1542
~XMLPrinter, 1543
F, 1545
GetPrintStyle, 1543
HandleBulkData, 1543
LOADBULKDATA, 1543
OnlyUUID, 1543
Print, 1543
PrintDataElement, 1544
PrintDataSet, 1544
PrintSQ, 1544
PrintStyle, 1545
PrintStyles, 1543
SetFile, 1544
SetStyle, 1544
XMLPrinter, 1543
gdcm::XMLPrivateDictReader, 1545
~XMLPrivateDictReader, 1547
CharacterDataHandler, 1547
EndElement, 1547
GetPrivateDict, 1547
HandleDescription, 1547
HandleEntry, 1548

StartElement, 1548
XMLPrivateDictReader, 1547
GDCM_DIFFERENT
gdcm, 89
GDCM_DO_JOIN
gdcmStaticAssert.h, 1603
GDCM_DO_JOIN2
gdcmStaticAssert.h, 1603
GDCM_EQUAL
gdcm, 89
GDCM_EXPORT
gdcmWin32.h, 1631
GDCM_FUNCTION
gdcmTrace.h, 1622
GDCM_GREATER
gdcm, 89
GDCM_GREATEROREQUAL
gdcm, 89
GDCM_JOIN
gdcmStaticAssert.h, 1603
GDCM_LEGACY
gdcmLegacyMacro.h, 1582
GDCM_LEGACY_BODY
gdcmLegacyMacro.h, 1582
GDCM_LEGACY_REPLACED_BODY
gdcmLegacyMacro.h, 1582
GDCM_LESS
gdcm, 89
GDCM_LESSOREQUAL
gdcm, 89
GDCM_NOOP_STATEMENT
gdcmLegacyMacro.h, 1583
GDCM_STATIC_ASSERT
gdcm::Attribute< Group, Element, TVR, TVM >, 161
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 171
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 178
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 190
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 198
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 204
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 211
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
gdcmStaticAssert.h, 1603
gdcmAAbortPDU.h, 2056, 2057
gdcmAAssociateACPDU.h, 2057, 2058
gdcmAAssociateRJPDU.h, 2060
gdcmAAssociateRQPDU.h, 2061, 2062
gdcmAbstractSyntax.h, 2064, 2065
gdcmAnonymizeEvent.h, 1860, 1862
gdcmAnonymizer.h, 1862, 1863
gdcmApplicationContext.h, 2066, 2067
gdcmApplicationEntity.h, 1864, 1865
gdcmAReleaseRPPDU.h, 2067, 2068
gdcmAReleaseRQPDU.h, 2069, 2070
gdcmARTIMTimer.h, 2070, 2071
gdcmASN1.h, 1549, 1550
gdcmAssertAlwaysMacro
gdcmTrace.h, 1622
gdcmAssertMacro
gdcmTrace.h, 1622
gdcmAsynchronousOperationsWindowSub.h, 2072
gdcmAttribute.h, 1669, 1670
gdcmAudioCodec.h, 1866, 1867
gdcmBase64.h, 1551
gdcmBaseCompositeMessage.h, 2073, 2074
gdcmBaseNormalizedMessage.h, 2075, 2076
gdcmBasePDU.h, 2076, 2077
gdcmBaseQuery.h, 2078, 2079
gdcmBaseRootQuery.h, 2080, 2081
gdcmBasicOffsetTable.h, 1683, 1684
gdcmBitmap.h, 1867, 1868
gdcmBitmapToBitmapFilter.h, 1871, 1872
gdcmBoxRegion.h, 1552, 1553
gdcmByteBuffer.h, 1686, 1687
gdcmByteSwap.h, 1553, 1554
gdcmByteSwapFilter.h, 1689
gdcmByteValue.h, 1690, 1691
gdcmCAPICryptoFactory.h, 1555, 1556
gdcmCAPICryptographicMessageSyntax.h, 1556, 1557
gdcmCEchoMessages.h, 2083
gdcmCFindMessages.h, 2084, 2085
gdcmCleaner.h, 1872, 1873
gdcmCMoveMessages.h, 2085, 2086
gdcmCodec.h, 1874, 1875
gdcmCoder.h, 1876, 1877
gdcmCodeString.h, 1694, 1695
gdcmCommand.h, 1559, 1560
gdcmCommandDataSet.h, 2087, 2088
gdcmCompositeMessageFactory.h, 2089
gdcmCompositeNetworkFunctions.h, 2090, 2091
gdcmConstCharWrapper.h, 1877, 1878
gdcmCP246ExplicitDataElement.h, 1696
gdcmCryptoFactory.h, 1562, 1563
gdcmCryptographicMessageSyntax.h, 1564, 1565
gdcmCSAElement.h, 1697, 1699
gdcmCSAHeader.h, 1701
gdcmCSAHeaderDict.h, 1632, 1633
gdcmCSAHeaderDictEntry.h, 1635, 1636
gdcmCStoreMessages.h, 2092
gdcmCurve.h, 1878, 1880

gdcmDataElement.h, 1703, 1704
gdcmDataEvent.h, 1566, 1567
gdcmDataSet.h, 1707, 1708
gdcmDataSetEvent.h, 1711, 1712
gdcmDataSetHelper.h, 1881
gdcmDebugMacro
    gdcmTrace.h, 1623
gdcmDecoder.h, 1882, 1883
gdcmDefinedTerms.h, 1809, 1810
gdcmDeflateStream.h, 1568
gdcmDefs.h, 1810, 1812
gdcmDeltaEncodingCodec.h, 1884
gdcmDICOMDIR.h, 1885, 1886
gdcmDICOMDIRGenerator.h, 1886, 1887
gdcmDict.h, 1638, 1639
gdcmDictConverter.h, 1643, 1644
gdcmDictEntry.h, 1645, 1646
gdcmDictPrinter.h, 1888, 1889
gdcmDicts.h, 1648, 1649
gdcmDIMSE.h, 2093, 2094
gdcmDirectionCosines.h, 1889, 1890
gdcmDirectory.h, 1568, 1569
gdcmDirectoryHelper.h, 1891
gdcmDPath.h, 1892, 1893
gdcmDummyValueGenerator.h, 1571
gdcmDumper.h, 1894, 1895
gdcmElement.h, 1713, 1714
gdcmEmptyMaskGenerator.h, 1896
gdcmEncapsulatedDocument.h, 1897, 1898
gdcmEnumeratedValues.h, 1813
gdcmEquipmentManufacturer.h, 1898, 1899
gdcmErrorMacro
    gdcmTrace.h, 1623
gdcmEvent.h, 1572, 1574
    gdcmEventMacro, 1573
gdcmEventMacro
    gdcmEvent.h, 1573
gdcmException.h, 1575, 1576
gdcmExplicitDataElement.h, 1725, 1726
gdcmExplicitImplicitDataElement.h, 1727, 1728
gdcmFiducials.h, 1900
gdcmFile.h, 1729, 1730
gdcmFileAnonymizer.h, 1901, 1902
gdcmFileChangeTransferSyntax.h, 1902, 1903
gdcmFileDecompressLookupTable.h, 1904, 1905
gdcmFileDerivation.h, 1906
gdcmFileExplicitFilter.h, 1908
gdcmFileMetaInformation.h, 1730, 1732
gdcmFilename.h, 1577, 1578
gdcmFileNameEvent.h, 1578, 1579
gdcmFilenameGenerator.h, 1580, 1581
gdcmFileSet.h, 1733, 1735
gdcmFileStreamer.h, 1909, 1910
gdcmFindPatientRootQuery.h, 2095, 2096
gdcmFindStudyRootQuery.h, 2097
gdcmFragment.h, 1735, 1737
gdcmGlobal.h, 1650, 1651
gdcmGroupDict.h, 1652, 1653
gdcmIconImage.h, 1911, 1912
gdcmIconImageFilter.h, 1913, 1914
gdcmIconImageGenerator.h, 1915, 1916
gdcmImage.h, 1916, 1918
gdcmImageApplyLookupTable.h, 1919
gdcmImageChangePhotometricInterpretation.h, 1920, 1921
gdcmImageChangePlanarConfiguration.h, 1923
gdcmImageChangeTransferSyntax.h, 1924, 1925
gdcmImageCodec.h, 1926, 1927
gdcmImageConverter.h, 1929, 1930
gdcmImageFragmentSplitter.h, 1931
gdcmImageHelper.h, 1932, 1933
gdcmImageReader.h, 1934, 1936
gdcmImageRegionReader.h, 1936, 1937
gdcmImageToImageFilter.h, 1938, 1939
gdcmImageWriter.h, 1939, 1940
gdcmImplementationClassUIDSub.h, 2098, 2099
gdcmImplementationUIDSub.h, 2100, 2101
gdcmImplementationVersionNameSub.h, 2101, 2102
gdcmImplicitDataElement.h, 1740, 1741
gdcmIOD.h, 1814, 1815
gdcmIODEntry.h, 1817, 1819
gdcmIODs.h, 1819, 1821
gdcmIPPSorter.h, 1941, 1942
gdcmItem.h, 1741, 1743
gdcmJPEG12Codec.h, 1943
gdcmJPEG16Codec.h, 1944, 1945
gdcmJPEG2000Codec.h, 1946
gdcmJPEG8Codec.h, 1948
gdcmJPEGCodec.h, 1949, 1950
gdcmJPEGLSCodec.h, 1952
gdcmJSON.h, 1953, 1954
gdcmKAKADUCodec.h, 1955, 1956
gdcmLegacyMacro.h, 1581, 1583
    GDCM_LEGACY, 1582
    GDCM_LEGACY_BODY, 1582
    GDCM_LEGACY_REPLACED_BODY, 1582
    GDCM_NOOP_STATEMENT, 1583
gdcmLO.h, 1747
gdcmLookupTable.h, 1956, 1957
gdcmMacro.h, 1822, 1823
gdcmMacroEntry.h, 1825, 1827
    GDCMMACROENTRY_H, 1826
GDCMMACROENTRY_H
    gdcmMacroEntry.h, 1826
gdcmMacros.h, 1828, 1829
gdcmMaximumLengthSub.h, 2103, 2104
gdcmMD5.h, 1584, 1585
gdcmMEC_MR3.h, 1959, 1960

gdcmMediaStorage.h, 1748, 1749
gdcmMeshPrimitive.h, 1960, 1961
gdcmModalityPerformedProcedureStepCreateQuery.h, 2105
gdcmModalityPerformedProcedureStepSetQuery.h, 2106, 2107
gdcmModule.h, 1830, 1832
gdcmModuleEntry.h, 1833, 1835
gdcmModules.h, 1836, 1837
gdcmMovePatientRootQuery.h, 2107, 2108
gdcmMoveStudyRootQuery.h, 2109
gdcmMrProtocol.h, 1752, 1753
gdcmNActionMessages.h, 2110, 2111
gdcmNCreateMessages.h, 2111, 2112
gdcmNDeleteMessages.h, 2113
gdcmNestedModuleEntries.h, 1838, 1840
gdcmNetworkEvents.h, 2114, 2115
gdcmNetworkStateID.h, 2116, 2117
gdcmNEventReportMessages.h, 2118, 2119
gdcmNGetMessages.h, 2119, 2120
gdcmNormalizedMessageFactory.h, 2120, 2121
gdcmNormalizedNetworkFunctions.h, 2122, 2123
gdcmNSetMessages.h, 2124
gdcmObject.h, 1585, 1586
gdcmOpenSSLCryptoFactory.h, 1588, 1589
gdcmOpenSSLCryptographicMessageSyntax.h, 1589, 1591
gdcmOpenSSLP7CryptoFactory.h, 1591, 1592
gdcmOpenSSLP7CryptographicMessageSyntax.h, 1593, 1594
gdcmOrientation.h, 1963
gdcmOverlay.h, 1964, 1965
gdcmParseException.h, 1754, 1755
gdcmParser.h, 1756, 1757
gdcmPatient.h, 1840, 1841
gdcmPDataTFPDU.h, 2125, 2126
gdcmPDBElement.h, 1759, 1760
gdcmPDBHeader.h, 1761, 1762
gdcmPDFCodec.h, 1967, 1968
gdcmPDUFactory.h, 2127
gdcmPersonName.h, 1968, 1969
gdcmPGXCodec.h, 1970, 1971
gdcmPhotometricInterpretation.h, 1971, 1972
gdcmPixelFormat.h, 1973, 1975
gdcmPixmap.h, 1977, 1978
gdcmPixmapReader.h, 1980, 1981
gdcmPixmapToPixmapFilter.h, 1982
gdcmPixmapWriter.h, 1983, 1984
gdcmPNMCodec.h, 1985, 1986
gdcmPreamble.h, 1763, 1764
gdcmPresentationContext.h, 2128, 2130
gdcmPresentationContextAC.h, 2130, 2132
gdcmPresentationContextGenerator.h, 2132, 2133
gdcmPresentationContextRQ.h, 2134, 2135
gdcmPresentationDataValue.h, 2136, 2137
gdcmPrinter.h, 1986, 1988
gdcmPrivateTag.h, 1765, 1766
gdcmProgressEvent.h, 1595, 1596
gdcmPVRGCodec.h, 1989, 1990
gdcmPythonFilter.h, 2231, 2232
gdcmQueryBase.h, 2138, 2140
gdcmQueryFactory.h, 2141, 2142
gdcmQueryImage.h, 2142, 2143
gdcmQueryPatient.h, 2144, 2145
gdcmQuerySeries.h, 2146
gdcmQueryStudy.h, 2147, 2148
gdcmRAWCodec.h, 1991
gdcmReader.h, 1767, 1768
gdcmRegion.h, 1596, 1598
gdcmRescaler.h, 1992, 1993
gdcmRLECodec.h, 1994
gdcmRoleSelectionSub.h, 2149
gdcmScanner.h, 1995, 1996
gdcmScanner2.h, 1998, 1999
gdcmSegment.h, 2001, 2003
gdcmSegmentedPaletteColorLookupTable.h, 2005
gdcmSegmentHelper.h, 2006, 2007
gdcmSegmentReader.h, 2008, 2010
gdcmSegmentWriter.h, 2010, 2012
gdcmSequenceOfFragments.h, 1769, 1770
gdcmSequenceOfItems.h, 1774, 1775
gdcmSerieHelper.h, 2012, 2014
gdcmSeries.h, 1842, 1843
gdcmServiceClassApplicationInformation.h, 2150, 2151
gdcmServiceClassUser.h, 2152, 2153
gdcmSHA1.h, 1599
gdcmSimpleSubjectWatcher.h, 2015, 2016
gdcmSmartPointer.h, 1600, 1601
gdcmSOPClassExtendedNegociationSub.h, 2154, 2155
gdcmSOPClassUIDToIOD.h, 1654
gdcmSorter.h, 2017, 2019
gdcmSpacing.h, 2020
gdcmSpectroscopy.h, 2021, 2022
gdcmSplitMosaicFilter.h, 2022, 2023
gdcmStaticAssert.h, 1602, 1604
    GDCM_DO_JOIN, 1603
    GDCM_DO_JOIN2, 1603
    GDCM_JOIN, 1603
    GDCM_STATIC_ASSERT, 1603
gdcmStreamImageReader.h, 2025
gdcmStreamImageWriter.h, 2026, 2027
gdcmStrictScanner.h, 2028, 2029
gdcmStrictScanner2.h, 2031, 2032
gdcmString.h, 1604, 1606
gdcmStringFilter.h, 2034, 2035
gdcmStudy.h, 1843, 1845
gdcmSubject.h, 1608
gdcmSurface.h, 2036, 2037

gdcmSurfaceHelper.h, 2040, 2041
gdcmSurfaceReader.h, 2043, 2044
gdcmSurfaceWriter.h, 2045, 2046
gdcmSwapCode.h, 1609, 1610
gdcmSwapper.h, 1611, 1612
gdcmSystem.h, 1614
gdcmTable.h, 1845, 1846
gdcmTableEntry.h, 1847, 1849
gdcmTableReader.h, 1849, 1851
gdcmTag.h, 1778, 1780
gdcmTagPath.h, 2046, 2047
gdcmTagToVR.h, 1783
gdcmTerminal.h, 1616, 1617
gdcmTestDriver.h, 1618
gdcmTesting.h, 1619
gdcmTrace.h, 1621, 1625
    GDCM_FUNCTION, 1622
    gdcmAssertAlwaysMacro, 1622
    gdcmAssertMacro, 1622
    gdcmDebugMacro, 1623
    gdcmErrorMacro, 1623
    gdcmWarningMacro, 1624
gdcmTransferSyntax.h, 1784, 1785
gdcmTransferSyntaxSub.h, 2155, 2157
gdcmType.h, 1852, 1853
gdcmTypes.h, 1627
gdcmUIDGenerator.h, 2048, 2049
gdcmUIDs.h, 1655, 1656
gdcmULAction.h, 2157, 2158
gdcmULActionAA.h, 2159, 2160
gdcmULActionAE.h, 2161, 2162
gdcmULActionAR.h, 2163, 2164
gdcmULActionDT.h, 2166
gdcmULBasicCallback.h, 2167, 2168
gdcmULConnection.h, 2168, 2169
gdcmULConnectionCallback.h, 2171, 2172
gdcmULConnectionInfo.h, 2172, 2174
gdcmULConnectionManager.h, 2174, 2175
gdcmULEvent.h, 2177, 2178
gdcmULTransitionTable.h, 2179, 2180
gdcmULWritingCallback.h, 2182
gdcmUNExplicitDataElement.h, 1786, 1787
gdcmUNExplicitImplicitDataElement.h, 1788, 1789
gdcmUnpacker12Bits.h, 1628, 1629
gdcmUsage.h, 1854, 1857
gdcmUserInformation.h, 2183, 2184
gdcmUUIDGenerator.h, 2050
gdcmValidate.h, 2051, 2052
gdcmValue.h, 1789, 1790
gdcmValueIO.h, 1791, 1792
gdcmVersion.h, 1629, 1630
gdcmVL.h, 1792, 1793
gdcmVM.h, 1795, 1797
    TYPETOLENGTH, 1796
gdcmVR.h, 1798, 1801
    TYPETOENCODING, 1800
    VRTypeTemplateCase, 1800
gdcmVR16ExplicitDataElement.h, 1805, 1806
gdcmWarningMacro
    gdcmTrace.h, 1624
gdcmWaveform.h, 2052, 2053
gdcmWin32.h, 1630, 1631
    GDCM_EXPORT, 1631
gdcmWLMFindQuery.h, 2185, 2186
gdcmWriter.h, 1807, 1808
gdcmXMLDictReader.h, 1857, 1858
gdcmXMLPrinter.h, 2053, 2054
gdcmXMLPrivateDictReader.h, 1859, 1860
GEMS
    gdcm::Dicts, 436
    gdcm::EquipmentManufacturer, 532
GeneralECGWaveformStorage
    gdcm::MediaStorage, 798
GeneralElectricMagneticResonanceImageStorage
    gdcm::MediaStorage, 799
Generate
    gdcm::DICOMDIRGenerator, 418
    gdcm::DummyValueGenerator, 454
    gdcm::FilenameGenerator, 589
    gdcm::IconImageGenerator, 626
    gdcm::UIDGenerator, 1283
    gdcm::UUIDGenerator, 1384
GenerateFromFilenames
    gdcm::PresentationContextGenerator, 974
GenerateFromUID
    gdcm::PresentationContextGenerator, 974
GenerateUUID
    gdcm::UIDGenerator, 1283
GEPrivate3DModelStorage
    gdcm::MediaStorage, 799
Get
    gdcm::ByteBuffer, 272
GetAbbreviation
    gdcm::GroupDict, 620
GetAbstractSyntax
    gdcm::network::PresentationContextRQ, 977
    gdcm::PresentationContext, 969
GetAbstractSyntaxUID
    gdcm::BaseQuery, 234
    gdcm::FindPatientRootQuery, 604
    gdcm::FindStudyRootQuery, 608
    gdcm::ModalityPerformedProcedureStepCreateQuery, 819
    gdcm::ModalityPerformedProcedureStepSetQuery, 823
    gdcm::MovePatientRootQuery, 837
    gdcm::MoveStudyRootQuery, 841
    gdcm::WLMFindQuery, 1532

GetAcceptedPresentationContexts
gdcm::network::ULConnection, 1348
GetAcquisitionSize
gdcm::SplitMosaicFilter, 1150
GetAETitle
gdcm::ServiceClassUser, 1117
GetAlgorithmFamily
gdcm::Surface, 1202
GetAlgorithmName
gdcm::Surface, 1203
GetAlgorithmVersion
gdcm::Surface, 1203
GetALGOType
gdcm::Segment, 1071
GetALGOTypeString
gdcm::Segment, 1071
GetAllFilenamesFromPrivateTagToValue
gdcm::Scanner2, 1063
gdcm::StrictScanner2, 1181
GetAllFilenamesFromPublicTagToValue
gdcm::Scanner2, 1063
gdcm::StrictScanner2, 1181
GetAllFilenamesFromTagToValue
gdcm::Scanner, 1053
gdcm::StrictScanner, 1171
GetAllRequiredTags
gdcm::QueryBase, 1006
GetAllTags
gdcm::QueryBase, 1006
GetAnatomicRegion
gdcm::Segment, 1071, 1072
GetAnatomicRegionModifiers
gdcm::Segment, 1072
GetAsDataElement
gdcm::Attribute< Group, Element, TVR, TVM >, 161
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 171
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 178
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 190
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 198
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 204
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 211
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
gdcm::Element< TVR, TVM >, 459
gdcm::Element< TVR, VM::VM1_2 >, 466
gdcm::Element< TVR, VM::VM1_n >, 471
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 484
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 496
gdcm::Element< TVR, VM::VM3_n >, 502
gdcm::Element< VR::AS, VM::VM5 >, 506
gdcm::Element< VR::OB, VM::VM1 >, 511
gdcm::Element< VR::OW, VM::VM1 >, 516
gdcm::network::AbstractSyntax, 130
gdcm::PrivateTag, 993
GetAsPoints
gdcm::Curve, 367
GetAsString
gdcm::CodeString, 315
GetAxisOfRotation
gdcm::Surface, 1203
GetBasicApplicationLevelConfidentialityProfileAttributes
gdcm::Anonymizer, 140
GetBitPosition
gdcm::Overlay, 892
GetBitsAllocated
gdcm::Overlay, 892
gdcm::PixelFormat, 933
GetBitSample
gdcm::LookupTable, 780
GetBitsStored
gdcm::PixelFormat, 933
GetBlob
gdcm::network::PresentationDataValue, 980
GetBuffer
gdcm::Bitmap, 254
gdcm::ByteValue, 280
gdcm::Parser, 902
gdcm::SequenceOfFragments, 1094
GetBuffer2
gdcm::Bitmap, 254
GetBufferAsRGBA
gdcm::LookupTable, 780
GetBufferLength
gdcm::Bitmap, 254
gdcm::JPEGLSCodec, 764
gdcm::PNMCodec, 962
gdcm::RLECodec, 1044
GetBuildVersion
gdcm::Version, 1393
GetByteValue
gdcm::CSAElement, 343
gdcm::DataElement, 373
GetCalledAETitle
gdcm::network::AAssociateRQPDU, 124
gdcm::network::ULConnectionInfo, 1354
gdcm::ServiceClassUser, 1117
GetCalledComputerName
gdcm::network::ULConnectionInfo, 1354

GetCalledIPAddress
gdcm::network::ULConnectionInfo, 1354
GetCalledIPPort
gdcm::network::ULConnectionInfo, 1354
GetCallingAETitle
gdcm::network::AAssociateRQPDU, 124
gdcm::network::ULConnectionInfo, 1354
GetCanonMECMR3Tag
gdcm::MEC_MR3, 793
GetCenterOfRotation
gdcm::Surface, 1203
GetCharacterFromCurrentLocale
gdcm::QueryFactory, 1008
GetCheckFileMetaInformation
gdcm::Writer, 1536
GetCipherType
gdcm::CAPICryptographicMessageSyntax, 289
gdcm::CryptographicMessageSyntax, 339
gdcm::OpenSSLCryptographicMessageSyntax, 878
gdcm::OpenSSLP7CryptographicMessageSyntax, 884
GetCodec
gdcm::FileChangeTransferSyntax, 560
GetColorLevel
vtkImageColorViewer, 1488
GetColorWindow
vtkImageColorViewer, 1488
GetColumns
gdcm::Bitmap, 254
gdcm::Overlay, 892
GetCommand
gdcm::Subject, 1197
GetConnectionInfo
gdcm::network::ULConnection, 1348
GetConstructorString
gdcm::Dicts, 437
GetContourReferencedFrameOfReferenceClassUID
vtkRTStructSetProperties, 1521
GetContourReferencedFrameOfReferenceInstanceUID
vtkRTStructSetProperties, 1521
GetCryptographicMessageSyntax
gdcm::Anonymizer, 140
GetCSADataInfo
gdcm::CSAHeader, 351
GetCSAEEnd
gdcm::CSAHeader, 352
GetCSAElementByName
gdcm::CSAHeader, 352
GetCSAHeaderDict
gdcm::Dicts, 437
GetCSAHeaderDictEntry
gdcm::CSAHeaderDict, 356
GetCSAImageHeaderInfoTag
gdcm::CSAHeader, 352
GetCSASeriesHeaderInfoTag
gdcm::CSAHeader, 352
GetCTImageSeriesUIDs
gdcm::DirectoryHelper, 449
GetCurrentByteIndex
gdcm::Parser, 902
GetCurrentDateTime
gdcm::System, 1232
GetCurrentModuleFileName
gdcm::System, 1232
GetCurrentProcessFileName
gdcm::System, 1232
GetCurrentResourcesDirectory
gdcm::System, 1232
GetCurve
gdcm::Pixmap, 943
GetCurveDataDescriptor
gdcm::Curve, 367
GetCWD
gdcm::System, 1232
GetData
gdcm::DataEvent, 387
GetDataElement
gdcm::Bitmap, 254
gdcm::DataSet, 393, 394
gdcm::Item, 724
GetDataExtraRoot
gdcm::Testing, 1259
GetDataLength
gdcm::DataEvent, 387
GetDataRoot
gdcm::Testing, 1259
GetDataSet
gdcm::CSAHeader, 352
gdcm::DataSetEvent, 403
gdcm::File, 550
GetDataSetPos
gdcm::network::ULEvent, 1364
GetDataSets
gdcm::network::ULBasicCallback, 1345
GetDataSetTransferSyntax
gdcm::FileMetaInformation, 576
GetDataValueRepresentation
gdcm::Curve, 367
GetDebugFlag
gdcm::Trace, 1265
GetDebugStream
gdcm::Trace, 1266
GetDecodeLength
gdcm::Base64, 224
GetDEEnd
gdcm::DataSet, 394
GetDefaultTransferSyntax
gdcm::PresentationContextGenerator, 975

GetDefs
gdcm::Global, 616
gdcm::TableReader, 1241
GetDES
gdcm::DataSet, 394
GetDescription
gdcm::CSAHeaderDictEntry, 359
gdcm::Exception, 537
gdcm::ModuleEntry, 831
gdcm::Overlay, 892
GetDescriptiveName
vtkGDCMImageReader, 1422
vtkGDCMImageReader2, 1436
vtkGDCMImageWriter, 1451
GetDict
gdcm::XMLDictReader, 1541
GetDictEntry
gdcm::Dict, 422
gdcm::Dicts, 437
gdcm::PrivateDict, 988
GetDictEntryByKeyword
gdcm::Dict, 422
GetDictEntryByName
gdcm::Dict, 423
GetDictName
gdcm::DictConverter, 426
GetDicts
gdcm::Global, 616, 617
GetDictVM
gdcm::Attribute< Group, Element, TVR, TVM >, 162
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 171
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 178
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 190
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 204
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 211
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
GetDictVR
gdcm::Attribute< Group, Element, TVR, TVM >, 162
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 178
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 191
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 204
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 211
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
GetDimension
gdcm::Bitmap, 255
GetDimensions
gdcm::Bitmap, 255
gdcm::Curve, 367
gdcm::ImageCodec, 661
GetDimensionsValue
gdcm::ImageHelper, 674
GetDimensionsValueForResolution
gdcm::StreamImageReader, 1157
GetDirectionCosines
gdcm::Image, 634
GetDirectionCosinesFromDataSet
gdcm::ImageHelper, 674
GetDirectionCosinesTolerance
gdcm::IPPSorter, 716
GetDirectionCosinesValue
gdcm::ImageHelper, 675
GetDirectories
gdcm::Directory, 447
GetElapsedTime
gdcm::network::ARTIMTimer, 154
GetElement
gdcm::Tag, 1247
GetElementTag
gdcm::Tag, 1247
GetEncodeLength
gdcm::Base64, 224
GetErrorCode
gdcm::Parser, 902
GetErrorFlag
gdcm::Trace, 1266
GetErrorStream
gdcm::Trace, 1266
GetErrorString
gdcm::Parser, 903
GetEvent
gdcm::network::ULEvent, 1364
GetEventName
gdcm::AnonymizeEvent, 134
gdcm::DataEvent, 387
gdcm::DataSetEvent, 403
gdcm::Event, 534
gdcm::FileNameEvent, 587

gdcm::ProgressEvent, 998
GetExtension
gdcm::Filename, 582
GetFactoryInstance
gdcm::CryptoFactory, 337
GetFile
gdcm::Anonymizer, 140
gdcm::Cleaner, 302
gdcm::DICOMDIRGenerator, 419
gdcm::FileDecompressLookupTable, 563
gdcm::FileDerivation, 566, 567
gdcm::FileExplicitFilter, 570
gdcm::IconImageFilter, 623
gdcm::PythonFilter, 1004
gdcm::Reader, 1027
gdcm::SplitMosaicFilter, 1151
gdcm::StreamImageReader, 1158
gdcm::StringFilter, 1192
gdcm::Writer, 1536
vtkGDCMMedicalImageProperties, 1458
GetFileExtensions
vtkGDCMImageReader, 1422
vtkGDCMImageReader2, 1436
vtkGDCMImageWriter, 1451
GetFileMetaInformationVersion
gdcm::FileMetaInformation, 576
GetFileName
gdcm::Filename, 582
gdcm::FileNameEvent, 587
gdcm::Testing, 1259
vtkGDCMImageWriter, 1451
vtkGDCMThreadedImageReader2, 1480
GetFilename
gdcm::FilenameGenerator, 589
gdcm::TableReader, 1241
GetFilenameFromPrivateTagToValue
gdcm::Scanner2, 1063
gdcm::StrictScanner2, 1181
GetFilenameFromPublicTagToValue
gdcm::Scanner2, 1063
gdcm::StrictScanner2, 1181
GetFilenameFromTagToValue
gdcm::Scanner, 1053
gdcm::StrictScanner, 1171
GetFileNames
gdcm::Testing, 1260
GetFilenames
gdcm::Directory, 447
gdcm::FilenameGenerator, 590
gdcm::Scanner, 1053
gdcm::Scanner2, 1063
gdcm::Sorter, 1143
gdcm::StrictScanner, 1172
gdcm::StrictScanner2, 1181
GetFilenamesFromSeriesUIDs
gdcm::DirectoryHelper, 449
GetFiles
gdcm::FileSet, 592
GetFiniteVolume
gdcm::Surface, 1203
GetFirstSingleSerieUIDFileSet
gdcm::SerieHelper, 1110
GetForcePixelSpacing
gdcm::ImageHelper, 675
GetForceRescaleInterceptSlope
gdcm::ImageHelper, 675
GetFormat
gdcm::CSAHeader, 353
GetFragBuffer
gdcm::SequenceOfFragments, 1094
GetFragment
gdcm::SequenceOfFragments, 1094
GetFragmentSizeMax
gdcm::ImageFragmentSplitter, 672
GetFrameOfReference
gdcm::DirectoryHelper, 449
GetFullLength
gdcm::FileMetaInformation, 576
GetGDCMDataRoot
vtkGDCMTesting, 1471
GetGDCMImplementationClassUID
gdcm::FileMetaInformation, 576
GetGDCMImplementationVersionName
gdcm::FileMetaInformation, 577
GetGDCMSourceApplicationEntityTitle
gdcm::FileMetaInformation, 577
GetGDCMUID
gdcm::UIDGenerator, 1283
GetGroup
gdcm::Curve, 367
gdcm::Overlay, 892
gdcm::Tag, 1248
GetHasExpired
gdcm::network::ARTIMTimer, 154
GetHeader
gdcm::File, 551
GetHeaderInfo
gdcm::ImageCodec, 661
gdcm::JPEG12Codec, 731
gdcm::JPEG16Codec, 736
gdcm::JPEG2000Codec, 742
gdcm::JPEG8Codec, 749
gdcm::JPEGCodec, 756
gdcm::JPEGLSCodec, 765
gdcm::PGXCodec, 925
gdcm::PNMCodec, 962
gdcm::RAWCodec, 1023
gdcm::RLECodec, 1044

GetHierachicalSearchTags
gdcm::QueryBase, 1006
gdcm::QueryImage, 1010
gdcm::QueryPatient, 1013
gdcm::QuerySeries, 1015
gdcm::QueryStudy, 1018
GetHighBit
gdcm::PixelFormat, 934
GetHostName
gdcm::System, 1232
GetIconImage
gdcm::IconImageFilter, 623
gdcm::IconImageGenerator, 626
gdcm::Pixmap, 943
vtkGDCMImageReader, 1422
vtkGDCMImageReader2, 1437
GetIconImagePort
vtkGDCMImageReader2, 1437
GetIE
gdcm::IODEntry, 709
GetImage
gdcm::ImageReader, 682
gdcm::ImageWriter, 695, 696
gdcm::PixmapWriter, 956
gdcm::SplitMosaicFilter, 1151
GetImplementationClassUID
gdcm::FileMetaInformation, 577
GetImplementationVersionName
gdcm::FileMetaInformation, 577
GetIndex
gdcm::SwapCode, 1226
gdcm::VM, 1401
GetInitialized
gdcm::CAPICryptographicMessageSyntax, 289
GetInput
gdcm::ImageToImageFilter, 691
gdcm::PixmapToPixmapFilter, 952
vtkImageColorViewer, 1488
GetInputFilename
gdcm::DictConverter, 426
GetInstance
gdcm::Global, 617
GetIntercept
gdcm::Image, 634
gdcm::Rescaler, 1037
GetInterfile
gdcm::CSAHeader, 353
GetInternal
gdcm::Preamble, 965
GetIOD
gdcm::IODs, 713
gdcm::SOPClassUIDToIOD, 1140
GetIODEntry
gdcm::IOD, 707
GetIODFromFile
gdcm::Defs, 409
GetIODFromSOPClassUID
gdcm::SOPClassUIDToIOD, 1140
GetIODNameFromMediaStorage
gdcm::Defs, 409
GetIODs
gdcm::Defs, 409
GetIsCommand
gdcm::network::PresentationDataValue, 980
GetIsLastFragment
gdcm::network::PresentationDataValue, 980
GetIStream
gdcm::network::ULEvent, 1364
GetItem
gdcm::SequenceOfItems, 1102, 1103
GetKey
gdcm::CSAElement, 343
GetKeys
gdcm::Scanner, 1054
gdcm::Scanner2, 1064
gdcm::StrictScanner, 1172
gdcm::StrictScanner2, 1182
GetKeyword
gdcm::DictEntry, 429
GetKeywordFromTag
gdcm::Dict, 423
GetLabel
gdcm::Orientation, 887
GetLastElement
gdcm::ParseException, 900
GetLastSystemError
gdcm::System, 1233
GetLength
gdcm::ByteValue, 280
gdcm::CP246ExplicitDataElement, 334
gdcm::DataElement, 374
gdcm::DataSet, 395
gdcm::Element< TVR, TVM >, 459
gdcm::Element< TVR, VM::VM1_2 >, 466
gdcm::Element< TVR, VM::VM1_n >, 471
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 484
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 496
gdcm::Element< TVR, VM::VM3_n >, 502
gdcm::Element< VR::AS, VM::VM5 >, 506
gdcm::Element< VR::OB, VM::VM1 >, 511
gdcm::Element< VR::OW, VM::VM1 >, 516
gdcm::ExplicitDataElement, 542
gdcm::ExplicitImplicitDataElement, 546
gdcm::Fragment, 613
gdcm::ImplicitDataElement, 703
gdcm::Item, 724

gdcm::Preamble, 965
gdcm::SequenceOfFragments, 1094
gdcm::SequenceOfItems, 1103
gdcm::Tag, 1248
gdcm::UNExplicitDataElement, 1371
gdcm::UNExplicitImplicitDataElement, 1375
gdcm::Value, 1389
gdcm::VL, 1395
gdcm::VM, 1401
gdcm::VR, 1407
gdcm::VR16ExplicitDataElement, 1413
GetLocaleCharset
gdcm::System, 1233
GetLossless
gdcm::JPEGCodec, 756
gdcm::JPEGLSCodec, 765
GetLossyFlag
gdcm::ImageCodec, 661
GetLossyFlagFromFile
gdcm::Testing, 1260
GetLUT
gdcm::Bitmap, 255
gdcm::ImageCodec, 661
gdcm::ImageHelper, 675
gdcm::LookupTable, 780
GetLUTDescriptor
gdcm::LookupTable, 780
GetLUTLength
gdcm::LookupTable, 780
GetMacro
gdcm::Macros, 790
GetMacroEntry
gdcm::Macro, 787
GetMacros
gdcm::Defs, 410
GetMajorAxisFromPatientRelativeDirectionCosine
gdcm::Orientation, 887
GetMajorVersion
gdcm::Version, 1393
GetManifold
gdcm::Surface, 1203
GetMapping
gdcm::Scanner, 1054
gdcm::StrictScanner, 1172
GetMappingFromPrivateTagToValue
gdcm::Scanner2, 1064
gdcm::StrictScanner2, 1182
GetMappingFromPublicTagToValue
gdcm::Scanner2, 1064
gdcm::StrictScanner2, 1182
GetMappingFromTagToValue
gdcm::Scanner, 1054
gdcm::StrictScanner, 1172
GetMappings
gdcm::Scanner, 1054
gdcm::StrictScanner, 1172
GetMax
gdcm::PixelFormat, 934
GetMaximumLength
gdcm::network::MaximumLengthSub, 791
GetMaximumLengthSub
gdcm::network::UserInformation, 1382
GetMaximumPointDistance
gdcm::Surface, 1203
GetMaxLength
gdcm::PersonName, 920
GetMaxPDULength
gdcm::network::ULConnectionInfo, 1354
GetMaxPDUSize
gdcm::network::ULConnection, 1348
GetMD5DataImage
gdcm::Testing, 1260
GetMD5DataImages
gdcm::Testing, 1260
GetMD5FromBrokenFile
gdcm::Testing, 1260
GetMD5FromFile
gdcm::Testing, 1260
GetMD5MetaImage
vtkGDCMTesting, 1471
GetMeanPointDistance
gdcm::Surface, 1203
GetMediaStorage
gdcm::DataSet, 395
gdcm::FileMetaInformation, 577
GetMediaStorageAsString
gdcm::FileMetaInformation, 577
GetMediaStorageDataFile
gdcm::Testing, 1261
GetMediaStorageDataFiles
gdcm::Testing, 1261
GetMediaStorageFromFile
gdcm::Testing, 1261
GetMeshPrimitive
gdcm::Surface, 1204
GetMessageHeader
gdcm::network::PresentationDataValue, 981
GetMetaInformationTS
gdcm::FileMetaInformation, 577
GetMHDMD5FromFile
vtkGDCMTesting, 1471
GetMin
gdcm::PixelFormat, 934
GetMinorVersion
gdcm::Version, 1393
GetModality
gdcm::MediaStorage, 801
GetModalityDimension

gdcm::MediaStorage, 801
GetModule
gdcm::Modules, 834
GetModuleEntry
gdcm::NestedModuleEntries, 856
GetModuleEntryInMacros
gdcm::Module, 827
GetModules
gdcm::Defs, 410
GetMPType
gdcm::MeshPrimitive, 814
GetMPTypeString
gdcm::MeshPrimitive, 814
GetMRImageSeriesUIDs
gdcm::DirectoryHelper, 450
GetMrProtocol
gdcm::CSAHeader, 353
GetMrProtocolByName
gdcm::MrProtocol, 844
GetMSString
gdcm::MediaStorage, 801
GetMSType
gdcm::MediaStorage, 801
GetMTime
vtkImageMapToColors16, 1499
GetName
gdcm::CSAElement, 344
gdcm::CSAHeaderDictEntry, 359
gdcm::DictEntry, 429
gdcm::Filename, 582
gdcm::GroupDict, 620
gdcm::IODEntry, 709
gdcm::Macro, 787
gdcm::Module, 827
gdcm::ModuleEntry, 831
gdcm::network::AbstractSyntax, 130
gdcm::network::ApplicationContext, 146
gdcm::network::TransferSyntaxSub, 1275
gdcm::PDBElement, 909
gdcm::QueryBase, 1007
gdcm::QueryImage, 1010
gdcm::QueryPatient, 1013
gdcm::QuerySeries, 1015
gdcm::QueryStudy, 1018
gdcm::UIDs, 1300
GetNeedByteSwap
gdcm::Bitmap, 255
gdcm::ImageCodec, 662
GetNegociatedType
gdcm::TransferSyntax, 1272
GetNestedDataSet
gdcm::Item, 724
GetNextSingleSerieUIDFileSet
gdcm::SerieHelper, 1110
GetNoOfItems
gdcm::CSAElement, 344
GetNumberOfComponents
gdcm::PersonName, 920
GetNumberOfContourReferencedFrameOfReferences
vtkRTStructSetProperties, 1521
GetNumberOfCurves
gdcm::Curve, 367
gdcm::Pixmap, 944
GetNumberOfDimensions
gdcm::Bitmap, 256
gdcm::ImageCodec, 662
GetNumberOfElementsFromArray
gdcm::VM, 1402
GetNumberOfFileNames
gdcm::Testing, 1261
GetNumberOfFilenames
gdcm::FilenameGenerator, 590
GetNumberOfFragments
gdcm::SequenceOfFragments, 1095
GetNumberOfIconImages
gdcm::IconImageFilter, 623
GetNumberOfImagesInMosaic
gdcm::SplitMosaicFilter, 1151
GetNumberOfIODs
gdcm::IOD, 707
GetNumberOfItems
gdcm::SequenceOfItems, 1103
GetNumberOfMD5DataImages
gdcm::Testing, 1261
GetNumberOfMD5MetaImages
vtkGDCMTesting, 1471
GetNumberOfMediaStorageDataFiles
gdcm::Testing, 1261
GetNumberOfModality
gdcm::MediaStorage, 802
GetNumberOfModuleEntries
gdcm::NestedModuleEntries, 857
GetNumberOfMSString
gdcm::MediaStorage, 802
GetNumberOfMSType
gdcm::MediaStorage, 802
GetNumberOfOverlays
gdcm::Pixmap, 944
GetNumberOfPoints
gdcm::Curve, 367
GetNumberOfPresentationContext
gdcm::network::AAssociateRQPDU, 125
GetNumberOfPresentationContextAC
gdcm::network::AAssociateACPDU, 118
GetNumberOfPresentationDataValues
gdcm::network::PDataTFPDU, 906
GetNumberOfPrimitivesData
gdcm::MeshPrimitive, 814

GetNumberOfReferencedFrameOfReferences
vtkRTStructSetProperties, 1522
GetNumberOfSegments
gdcm::SegmentWriter, 1088
GetNumberOfSOPClassToIOD
gdcm::SOPClassUIDToIOD, 1140
GetNumberOfStructureSetROIs
vtkRTStructSetProperties, 1522
GetNumberOfSurfacePoints
gdcm::Surface, 1204
GetNumberOfSurfaces
gdcm::SurfaceReader, 1218
gdcm::SurfaceWriter, 1223
GetNumberOfTransferSyntaxes
gdcm::network::PresentationContextRQ, 977
gdcm::PresentationContext, 969
GetNumberOfTransferSyntaxStrings
gdcm::UIDs, 1300
GetNumberOfValues
gdcm::Attribute< Group, Element, TVR, TVM >, 162
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 191
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 211
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
GetNumberOfVectors
gdcm::Surface, 1204
GetObliquityThresholdCosineValue
gdcm::Orientation, 887
GetOffScreenRendering
vtkImageColorViewer, 1488
GetOptionalTags
gdcm::QueryBase, 1007
gdcm::QueryImage, 1011
gdcm::QueryPatient, 1013
gdcm::QuerySeries, 1016
gdcm::QueryStudy, 1018
GetOrderedValues
gdcm::Scanner, 1054
gdcm::StrictScanner, 1172
GetOrigin
gdcm::Image, 634
gdcm::Overlay, 893
GetOriginValue
gdcm::ImageHelper, 675
GetOuput
gdcm::ImageConverter, 668
GetOutput
gdcm::BitmapToBitmapFilter, 266
gdcm::ImageToImageFilter, 691
gdcm::PixmapToPixmapFilter, 952
GetOutputAsBitmap
gdcm::BitmapToBitmapFilter, 266
GetOutputAsPixmap
gdcm::PixmapToPixmapFilter, 952
GetOutputFilename
gdcm::DictConverter, 426
GetOutputType
gdcm::DictConverter, 426
GetOverlay
gdcm::Pixmap, 944
vtkGDCMImageReader, 1422
vtkGDCMImageReader2, 1437
GetOverlayData
gdcm::Overlay, 893
GetOverlayPort
vtkGDCMImageReader2, 1437
GetOverlayTypeAsString
gdcm::Overlay, 893
GetOverlayTypeFromString
gdcm::Overlay, 893
GetOverlayVisibility
vtkImageColorViewer, 1489
GetOwner
gdcm::PrivateTag, 993
GetPath
gdcm::Filename, 582
GetPattern
gdcm::FilenameGenerator, 590
GetPDBEEnd
gdcm::PDBHeader, 912
GetPDBElementByName
gdcm::PDBHeader, 912
GetPDBInfoTag
gdcm::PDBHeader, 912
GetPDUs
gdcm::network::ULEvent, 1364
GetPDVs
gdcm::network::PDUFactory, 919
GetPermissions
gdcm::System, 1233
GetPhotometricInterpretation
gdcm::Bitmap, 256
gdcm::ImageChangePhotometricInterpretation, 643
gdcm::ImageCodec, 662
GetPhotometricInterpretationValue
gdcm::ImageHelper, 675

GetPIString
gdcm::PhotometricInterpretation, 928
GetPIType
gdcm::PhotometricInterpretation, 928
GetPixelFormat
gdcm::Bitmap, 256
gdcm::ImageCodec, 662
GetPixelFormatValue
gdcm::ImageHelper, 675
GetPixelRepresentation
gdcm::PixelFormat, 934
GetPixelSize
gdcm::PixelFormat, 934
GetPixelSpacingDataRoot
gdcm::Testing, 1261
GetPixmap
gdcm::FileDecompressLookupTable, 564
gdcm::IconImageGenerator, 626
gdcm::PixmapReader, 949
gdcm::PixmapWriter, 956
GetPlanarConfiguration
gdcm::Bitmap, 256
gdcm::ImageChangePlanarConfiguration, 648
gdcm::ImageCodec, 662
GetPlanarConfigurationValue
gdcm::ImageHelper, 676
GetPMSRescaleInterceptSlope
gdcm::ImageHelper, 676
GetPMTFInformationDataTag
gdcm::MEC_MR3, 793
GetPointCoordinatesData
gdcm::Surface, 1204
GetPointer
gdcm::ByteValue, 281
gdcm::LookupTable, 780
gdcm::SmartPointer< ObjectType >, 1136
vtkLookupTable16, 1516
GetPointerFromElement
gdcm::ImageHelper, 676
GetPointPositionAccuracy
gdcm::Surface, 1204
GetPointsBoundingBoxCoordinates
gdcm::Surface, 1204
GetPosition
vtkImageColorViewer, 1489
GetPreamble
gdcm::FileMetaInformation, 577
GetPrefix
gdcm::FilenameGenerator, 590
GetPresentationContext
gdcm::network::AAssociateRQPDU, 125
GetPresentationContextAC
gdcm::network::AAssociateACPDU, 118
GetPresentationContextACByID
gdcm::network::ULConnection, 1348
GetPresentationContextByAbstractSyntax
gdcm::network::AAssociateRQPDU, 125
GetPresentationContextByID
gdcm::network::AAssociateRQPDU, 125
GetPresentationContextID
gdcm::network::PresentationContextAC, 971
gdcm::network::PresentationContextRQ, 977
gdcm::network::PresentationDataValue, 981
gdcm::PresentationContext, 969
GetPresentationContextIDFromPresentationContext
gdcm::network::ULConnection, 1348
GetPresentationContextRQByID
gdcm::network::ULConnection, 1348
GetPresentationContexts
gdcm::network::AAssociateRQPDU, 125
gdcm::network::ULConnection, 1348
gdcm::PresentationContextGenerator, 975
GetPresentationDataValue
gdcm::network::PDataTFPDU, 906
GetPrettyPrint
gdcm::JSON, 767
GetPrimitiveData
gdcm::MeshPrimitive, 815
GetPrimitivesData
gdcm::MeshPrimitive, 815
GetPrimitiveType
gdcm::MeshPrimitive, 815
GetPrintStyle
gdcm::Printer, 985
gdcm::XMLPrinter, 1543
GetPrivateCreator
gdcm::DataSet, 395
gdcm::Tag, 1248
GetPrivateDict
gdcm::Dicts, 438
gdcm::XMLPrivateDictReader, 1547
GetPrivateMapping
gdcm::Scanner2, 1064
gdcm::StrictScanner2, 1182
GetPrivateMappings
gdcm::Scanner2, 1064
gdcm::StrictScanner2, 1182
GetPrivateOrderedValues
gdcm::Scanner2, 1064
gdcm::StrictScanner2, 1182
GetPrivateTag
gdcm::DataSet, 395
GetPrivateValue
gdcm::Scanner2, 1064
gdcm::StrictScanner2, 1182
GetPrivateValues
gdcm::Scanner2, 1065
gdcm::StrictScanner2, 1183

GetProcessingAlgorithm
gdcm::Surface, 1205
GetProgress
gdcm::ProgressEvent, 998
GetPropertyCategory
gdcm::Segment, 1072
GetPropertyType
gdcm::Segment, 1072
GetPropertyTypeModifiers
gdcm::Segment, 1072
GetProtocol
gdcm::network::ULConnection, 1349
GetPublicDict
gdcm::Dicts, 438
GetPublicMapping
gdcm::Scanner2, 1065
gdcm::StrictScanner2, 1183
GetPublicMappings
gdcm::Scanner2, 1065
gdcm::StrictScanner2, 1183
GetPublicOrderedValues
gdcm::Scanner2, 1065
gdcm::StrictScanner2, 1183
GetPublicValue
gdcm::Scanner2, 1065
gdcm::StrictScanner2, 1183
GetPublicValues
gdcm::Scanner2, 1065
gdcm::StrictScanner2, 1183
GetQuality
gdcm::JPEG2000Codec, 742
gdcm::JPEGCodec, 756
GetQueryDataSet
gdcm::BaseQuery, 235
GetQueryLevel
gdcm::QueryBase, 1007
gdcm::QueryImage, 1011
gdcm::QueryPatient, 1013
gdcm::QuerySeries, 1016
gdcm::QueryStudy, 1018
GetQueryLevelFromQueryRoot
gdcm::BaseRootQuery, 240
GetQueryLevelFromString
gdcm::BaseRootQuery, 240
GetQueryLevelString
gdcm::BaseRootQuery, 240
GetRate
gdcm::JPEG2000Codec, 742
GetRAWMD5FromFile
vtkGDCMTesting, 1471
GetRealWorldValueMappingContent
gdcm::ImageHelper, 676
GetReason
gdcm::network::PresentationContextAC, 971
GetRecommendedDisplayCIELabValue
gdcm::Surface, 1205
GetRecommendedDisplayGrayscaleValue
gdcm::Surface, 1205
GetRecommendedPresentationOpacity
gdcm::Surface, 1205
GetRecommendedPresentationType
gdcm::Surface, 1205
GetRef
gdcm::IODEntry, 710
GetReferencedFrameOfReferenceClassUID
vtkRTStructSetProperties, 1522
GetReferencedFrameOfReferenceInstanceUID
vtkRTStructSetProperties, 1522
GetRegion
gdcm::ImageRegionReader, 688
GetRequiredDataSet
gdcm::ModalityPerformedProcedureStepCreateQuery, 819
gdcm::ModalityPerformedProcedureStepSetQuery, 823
GetRequiredTags
gdcm::QueryBase, 1007
gdcm::QueryImage, 1011
gdcm::QueryPatient, 1013
gdcm::QuerySeries, 1016
gdcm::QueryStudy, 1018
GetRescaleInterceptSlopeValue
gdcm::ImageHelper, 676
GetReserved43_74
gdcm::network::AAssociateRQPDU, 125
GetResponses
gdcm::network::ULBasicCallback, 1345
GetRetired
gdcm::DictEntry, 430
GetRoot
gdcm::UIDGenerator, 1283
GetRows
gdcm::Bitmap, 257
gdcm::Overlay, 893
GetRTStructSeriesUIDs
gdcm::DirectoryHelper, 450
GetSamplesPerPixel
gdcm::PhotometricInterpretation, 928
gdcm::PixelFormat, 934
GetScalarType
gdcm::PixelFormat, 935
GetScalarTypeAsString
gdcm::PixelFormat, 935
GetScanner
gdcm::DICOMDIRGenerator, 419
GetSecondaryCaptureImagePlaneModule
gdcm::ImageHelper, 676
GetSegment

gdcm::SegmentWriter, 1088
GetSegmentAlgorithmName
gdcm::Segment, 1072
GetSegmentAlgorithmType
gdcm::Segment, 1073
GetSegmentDescription
gdcm::Segment, 1073
GetSegmentLabel
gdcm::Segment, 1073
GetSegmentNumber
gdcm::Segment, 1073
GetSegments
gdcm::SegmentReader, 1083
gdcm::SegmentWriter, 1088
GetSelectedPrivateGroupOffsetFromFile
gdcm::Testing, 1262
GetSelectedTagsOffsetFromFile
gdcm::Testing, 1262
GetSequenceOfFragments
gdcm::DataElement, 374
GetSeriesUIDsBySOPClassUID
gdcm::DirectoryHelper, 450
GetSize
gdcm::VR, 1408
vtkImageColorViewer, 1489
GetSizeof
gdcm::VR, 1408
GetSliceArray
gdcm::MrProtocol, 844
GetSliceMax
vtkImageColorViewer, 1489
GetSliceMin
vtkImageColorViewer, 1489
GetSliceRange
vtkImageColorViewer, 1489
GetSlope
gdcm::Image, 634
gdcm::Rescaler, 1037
GetSOPClassUID
gdcm::DirectoryHelper, 450
GetSOPClassUIDFromIOD
gdcm::SOPClassUIDToIOD, 1140
GetSOPClassUIDToIOD
gdcm::SOPClassUIDToIOD, 1140
GetSOPClassUIDToIODs
gdcm::SOPClassUIDToIOD, 1141
GetSOPInstanceUID
gdcm::BaseQuery, 235
GetSourceApplicationEntityTitle
gdcm::FileMetaInformation, 578
GetSourceDirectory
gdcm::Testing, 1262
GetSpacing
gdcm::Image, 634
GetSpacingTagFromMediaStorage
gdcm::ImageHelper, 677
GetSpacingValue
gdcm::ImageHelper, 677
GetStart
gdcm::ByteBuffer, 272
GetState
gdcm::network::ULConnection, 1349
GetStateIndex
gdcm::network, 108
GetSTATES
gdcm::Surface, 1205
GetSTATESString
gdcm::Surface, 1206
GetStream
gdcm::Trace, 1266
GetStreamCurrentPosition
gdcm::Reader, 1027
GetStreamOffsetFromFile
gdcm::Testing, 1262
GetStreamPtr
gdcm::Reader, 1028
gdcm::Writer, 1536
GetString
gdcm::MediaStorage, 802
gdcm::PhotometricInterpretation, 929
gdcm::TransferSyntax, 1272
gdcm::UIDs, 1301
GetStringValueFromTag
gdcm::DirectoryHelper, 450
GetStructureSetObservationNumber
vtkRTStructSetProperties, 1522
GetStructureSetROIDescription
vtkRTStructSetProperties, 1522
GetStructureSetROIGenerationAlgorithm
vtkRTStructSetProperties, 1522
GetStructureSetROIName
vtkRTStructSetProperties, 1522
GetStructureSetROINumber
vtkRTStructSetProperties, 1523
GetStructureSetROIObservationLabel
vtkRTStructSetProperties, 1523
GetStructureSetROIRefFrameRefUID
vtkRTStructSetProperties, 1523
GetStructureSetRTROIInterpretedType
vtkRTStructSetProperties, 1523
GetSurface
gdcm::Segment, 1073
GetSurfaceComments
gdcm::Surface, 1206
GetSurfaceCount
gdcm::Segment, 1073
GetSurfaceNumber
gdcm::Surface, 1206

GetSurfaceProcessing
gdcm::Surface, 1206
GetSurfaceProcessingDescription
gdcm::Surface, 1206
GetSurfaceProcessingRatio
gdcm::Surface, 1206
GetSurfaces
gdcm::Segment, 1073
GetSwapCode
gdcm::TransferSyntax, 1272
GetSwapCodeString
gdcm::SwapCode, 1226
GetSyngoDT
gdcm::CSAElement, 344
GetTable
gdcm::SequenceOfFragments, 1095
GetTableEntry
gdcm::Table, 1237
GetTag
gdcm::AnonymizeEvent, 134
gdcm::Attribute< Group, Element, TVR, TVM >, 162
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 191
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 211
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
gdcm::DataElement, 374, 375
GetTagListByLevel
gdcm::BaseRootQuery, 240
gdcm::FindPatientRootQuery, 604
gdcm::FindStudyRootQuery, 608
gdcm::MovePatientRootQuery, 837
gdcm::MoveStudyRootQuery, 841
gdcm::WLMFindQuery, 1532
GetTempDirectory
gdcm::Testing, 1262
GetTempDirectoryW
gdcm::Testing, 1262
GetTempFilename
gdcm::Testing, 1263
GetTempFilenameW
gdcm::Testing, 1263
GetTimeout
gdcm::network::ARTIMTimer, 155
gdcm::ServiceClassUser, 1117
GetTimer
gdcm::network::ULConnection, 1349
GetTimezoneOffsetFromUTC
gdcm::System, 1233
GetToplevel
gdcm::Directory, 447
GetToshibaMECMR3Tag
gdcm::MEC_MR3, 793
GetTransferSyntax
gdcm::Bitmap, 257
gdcm::ImageChangeTransferSyntax, 652
gdcm::network::PresentationContextAC, 971
gdcm::network::PresentationContextRQ, 978
gdcm::PresentationContext, 969
GetTransferSyntaxes
gdcm::network::PresentationContextRQ, 978
GetTransferSyntaxString
gdcm::UIDs, 1301
GetTransferSyntaxStrings
gdcm::UIDs, 1301
GetTSString
gdcm::TransferSyntax, 1272
GetTSType
gdcm::TransferSyntax, 1273
GetType
gdcm::ModuleEntry, 831
gdcm::Orientation, 887
gdcm::Overlay, 893
gdcm::PhotometricInterpretation, 929
GetTypeAsEnum
gdcm::Overlay, 893
GetTypeFromTag
gdcm::Defs, 410
gdcm::IOD, 707
GetTypeOfData
gdcm::Curve, 367
GetTypeOfDataDescription
gdcm::Curve, 368
GetTypeString
gdcm::Type, 1280
GetTypeType
gdcm::Type, 1280
GetUIDName
gdcm::UIDs, 1301
GetUIDString
gdcm::UIDs, 1301
GetUniqueTags
gdcm::QueryBase, 1007
gdcm::QueryImage, 1011
gdcm::QueryPatient, 1014
gdcm::QuerySeries, 1016
gdcm::QueryStudy, 1019

GetUnpackBuffer
gdcm::Overlay, 894
GetUnpackBufferLength
gdcm::Overlay, 894
GetUsage
gdcm::IODEntry, 710
GetUsageString
gdcm::Usage, 1379
GetUsageType
gdcm::IODEntry, 710
gdcm::Usage, 1379
GetUserData
gdcm::Parser, 903
GetUserInformation
gdcm::network::AAssociateACPDU, 118
gdcm::network::AAssociateRQPDU, 125
GetValidatedFile
gdcm::Validate, 1386
GetValidDataSet
gdcm::WLMFindQuery, 1532
GetValue
gdcm::Attribute< Group, Element, TVR, TVM >, 163
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 191
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 211
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
gdcm::CSAElement, 344
gdcm::DataElement, 375
gdcm::Element< TVR, TVM >, 460
gdcm::Element< TVR, VM::VM1_2 >, 466
gdcm::Element< TVR, VM::VM1_n >, 471
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 484
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 496
gdcm::Element< TVR, VM::VM3_n >, 502
gdcm::Element< VR::AS, VM::VM5 >, 506
gdcm::Element< VR::OB, VM::VM1 >, 511
gdcm::Element< VR::OW, VM::VM1 >, 516
gdcm::PDBElement, 909
gdcm::Scanner, 1054
gdcm::StrictScanner, 1173
GetValueAsSQ
gdcm::DataElement, 375
GetValues
gdcm::Attribute< Group, Element, TVR, TVM >, 163
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 191
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 211
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
gdcm::Element< TVR, TVM >, 460
gdcm::Element< TVR, VM::VM1_2 >, 466
gdcm::Element< TVR, VM::VM1_n >, 471
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 484
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 496
gdcm::Element< TVR, VM::VM3_n >, 502
gdcm::Element< VR::AS, VM::VM5 >, 506
gdcm::Element< VR::OB, VM::VM1 >, 511
gdcm::Element< VR::OW, VM::VM1 >, 516
gdcm::Scanner, 1055
gdcm::Scanner2, 1066
gdcm::StrictScanner, 1173
gdcm::StrictScanner2, 1184
GetVectorAccuracy
gdcm::Surface, 1206
GetVectorCoordinateData
gdcm::Surface, 1206
GetVectorDimensionality
gdcm::Surface, 1207
GetVersion
gdcm::MrProtocol, 844
gdcm::Version, 1393
GetVIEWType
gdcm::Surface, 1207
GetVIEWTypeString
gdcm::Surface, 1207
GetVL
gdcm::DataElement, 376
GetVL16Max
gdcm::VL, 1395
GetVL32Max
gdcm::VL, 1396

GetVM
gdcm::Attribute< Group, Element, TVR, TVM >, 163
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 192
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 217
gdcm::CSAElement, 345
gdcm::CSAHeaderDictEntry, 359
gdcm::DictEntry, 430
gdcm::Element< TVR, TVM >, 460
gdcm::Element< TVR, VM::VM1_2 >, 466
gdcm::Element< TVR, VM::VM1_n >, 471
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 484
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 496
gdcm::Element< TVR, VM::VM3_n >, 502
gdcm::Element< VR::AS, VM::VM5 >, 506
gdcm::Element< VR::OB, VM::VM1 >, 511
gdcm::Element< VR::OW, VM::VM1 >, 516
GetVMString
gdcm::VM, 1402
GetVMType
gdcm::VM, 1402
GetVMTypeFromLength
gdcm::VM, 1402
GetVoidPointer
gdcm::ByteValue, 281
GetVR
gdcm::Attribute< Group, Element, TVR, TVM >, 164
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 192
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
gdcm::CSAElement, 345
gdcm::CSAHeaderDictEntry, 359
gdcm::DataElement, 376
gdcm::DictEntry, 430
gdcm::Element< TVR, TVM >, 460
gdcm::Element< TVR, VM::VM1_2 >, 467
gdcm::Element< TVR, VM::VM1_n >, 471
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 485
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 497
gdcm::Element< TVR, VM::VM3_n >, 503
gdcm::Element< VR::AS, VM::VM5 >, 507
gdcm::Element< VR::OB, VM::VM1 >, 511
gdcm::Element< VR::OW, VM::VM1 >, 516
GetVRFromTag
gdcm, 92
GetVRString
gdcm::VR, 1408
GetVRStringFromFile
gdcm::VR, 1408
GetVRType
gdcm::VR, 1408
GetVRTypeFromFile
gdcm::VR, 1408
GetVTKDataRoot
vtkGDCMTesting, 1471
GetWarningFlag
gdcm::Trace, 1266
GetWarningStream
gdcm::Trace, 1266
GetWindowName
vtkImageColorViewer, 1489
GetXMax
gdcm::BoxRegion, 270
GetXMin
gdcm::BoxRegion, 270
GetYMax
gdcm::BoxRegion, 270
GetYMin
gdcm::BoxRegion, 270
GetZMax
gdcm::BoxRegion, 270
GetZMin
gdcm::BoxRegion, 271
GetZSpacing
gdcm::IPPSorter, 716
GetZSpacingTagFromMediaStorage
gdcm::ImageHelper, 677
GetZSpacingTolerance

gdcm::IPPSorter, 717
Global
gdcm::Defs, 411
gdcm::Dicts, 439
gdcm::Global, 616
GlobalInstance
gdcm, 102
GrabOverlayFromPixelData
gdcm::Overlay, 894
Graphics
gdcm::Overlay, 891
GRAY
gdcm::LookupTable, 778
GrayscaleSoftcopyPresentationStateStorageSOPClass
gdcm::MediaStorage, 799
GREEN
gdcm::LookupTable, 778
green
gdcm::terminal, 110
GroupDict
gdcm::GroupDict, 619
GroupStringVector
gdcm::GroupDict, 619
GuessFromModality
gdcm::MediaStorage, 802

HandleBulkData
gdcm::XMLPrinter, 1543
HandleDataSet
gdcm::network::ULBasicCallback, 1345
gdcm::network::ULConnectionCallback, 1352
gdcm::network::ULWritingCallback, 1367
HandleDescription
gdcm::XMLDictReader, 1541
gdcm::XMLPrivateDictReader, 1547
HandleEntry
gdcm::XMLDictReader, 1541
gdcm::XMLPrivateDictReader, 1548
HandleEvent
gdcm::network::ULTransitionTable, 1365
HandleIOD
gdcm::TableReader, 1241
HandleIODEntry
gdcm::TableReader, 1241
HandleMacro
gdcm::TableReader, 1241
HandleMacroEntry
gdcm::TableReader, 1241
HandleMacroEntryDescription
gdcm::TableReader, 1242
HandleModule
gdcm::TableReader, 1242
HandleModuleEntry
gdcm::TableReader, 1242
HandleModuleEntryDescription
gdcm::TableReader, 1242
HandleModuleInclude
gdcm::TableReader, 1242
HandleResponse
gdcm::network::ULBasicCallback, 1345
gdcm::network::ULConnectionCallback, 1352
gdcm::network::ULWritingCallback, 1367
HangingProtocolStorage
gdcm::MediaStorage, 799
HardcopyColorImageStorage
gdcm::MediaStorage, 800
HardcopyGrayscaleImageStorage
gdcm::MediaStorage, 799
HasObserver
gdcm::Subject, 1197
HemodynamicWaveformStorage
gdcm::MediaStorage, 798
hidden
gdcm::terminal, 110
HITACHI
gdcm::EquipmentManufacturer, 532
HSV
gdcm::PhotometricInterpretation, 928

Icon
gdcm::Pixmap, 945
IconDataScalarType
vtkGDCMImageReader, 1431
vtkGDCMImageReader2, 1446
IconImage
gdcm, 87
IconImageDataExtent
vtkGDCMImageReader, 1431
vtkGDCMImageReader2, 1446
IconImageFilter
gdcm::IconImageFilter, 622
IconImageGenerator
gdcm::IconImageGenerator, 625
IconNumberOfScalarComponents
vtkGDCMImageReader, 1431
vtkGDCMImageReader2, 1446
ID
gdcm::PresentationContext, 970
ignore_char
gdcm::ignore_char, 628
Image
gdcm::Image, 633
ImageActor
vtkImageColorViewer, 1496
ImageApplyLookupTable
gdcm::ImageApplyLookupTable, 639
ImageChangePhotometricInterpretation
gdcm::ImageChangePhotometricInterpretation, 643

gdcm::ImageCodec, 665
ImageChangePlanarConfiguration
gdcm::ImageChangePlanarConfiguration, 648
ImageChangeTransferSyntax
gdcm::Bitmap, 262
gdcm::ImageChangeTransferSyntax, 652
ImageCodec
gdcm::ImageCodec, 658
ImageConverter
gdcm::ImageConverter, 668
ImageFormat
vtkGDCMImageReader, 1431
vtkGDCMImageReader2, 1446
ImageFragmentSplitter
gdcm::ImageFragmentSplitter, 671
ImageNumberOrdering
gdcm::SerieHelper, 1110
ImageOrientationPatient
vtkGDCMImageReader, 1431
vtkGDCMImageReader2, 1446
ImagePositionPatient
vtkGDCMImageReader, 1431
vtkGDCMImageReader2, 1446
ImagePositionPatientOrdering
gdcm::SerieHelper, 1110
ImageReader
gdcm::ImageReader, 682
ImageRegionReader
gdcm::ImageRegionReader, 687
gdcm::JPEG2000Codec, 745
gdcm::JPEGCodec, 758
gdcm::JPEGLSCodec, 766
gdcm::RLECodec, 1046
ImageToImageFilter
gdcm::ImageToImageFilter, 691
ImageWriter
gdcm::ImageWriter, 695
ImplementationClassUIDSub
gdcm::network::ImplementationClassUIDSub, 697
ImplementationUIDSub
gdcm::network::ImplementationUIDSub, 698
ImplementationVersionNameSub
gdcm::network::ImplementationVersionNameSub, 699
Implicit
gdcm::TransferSyntax, 1271
ImplicitVRBigEndianACRNEMA
gdcm::TransferSyntax, 1271
ImplicitVRBigEndianPrivateGE
gdcm::TransferSyntax, 1271
ImplicitVRLittleEndian
gdcm::TransferSyntax, 1271
IncompleteLUT
gdcm::LookupTable, 783
InformationObjectDefinition Directory Reference, 61
InitFromRQ
gdcm::network::AAssociateACPDU, 118
Initialize
gdcm::network::ULConnectionInfo, 1354
InitializeBlueLUT
gdcm::LookupTable, 781
InitializeConnection
gdcm::network::ULConnection, 1349
gdcm::ServiceClassUser, 1117
Initialized
gdcm::LookupTable, 781
InitializeDataSet
gdcm::BaseRootQuery, 240
gdcm::FindPatientRootQuery, 605
gdcm::FindStudyRootQuery, 609
gdcm::MovePatientRootQuery, 838
gdcm::MoveStudyRootQuery, 842
gdcm::WLMFindQuery, 1532
InitializeGreenLUT
gdcm::LookupTable, 781
InitializeIncomingConnection
gdcm::network::ULConnection, 1349
InitializeLUT
gdcm::LookupTable, 781
InitializeRedLUT
gdcm::LookupTable, 781
InitializeRTStructSet
vtkGDCMPolyDataWriter, 1466
InitOpenSSL
gdcm::OpenSSLCryptoFactory, 876
Input
gdcm::BitmapToBitmapFilter, 266
Insert
gdcm::CommandDataSet, 323
gdcm::DataSet, 395
gdcm::FileMetaInformation, 578
gdcm::GroupDict, 620
InsertDataElement
gdcm::DataSet, 396
gdcm::Item, 724
InsertEntry
gdcm::Table, 1237
InstallPipeline
vtkImageColorViewer, 1490
INT12
gdcm::PixelFormat, 932
INT16
gdcm::PixelFormat, 932
INT32
gdcm::PixelFormat, 932
INT64
gdcm::PixelFormat, 933
INT8

gdcm::PixelFormat, 932
Interactor
vtkImageColorViewer, 1496
InteractorStyle
vtkImageColorViewer, 1496
INTERFILE
gdcm::CSAHeader, 351
Internal
gdcm::ApplicationEntity, 149
gdcm::Attribute< Group, Element, TVR, TVM >, 167
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 175
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 181
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 187
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 201
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 207
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 214
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 219
gdcm::Element< TVR, TVM >, 463
gdcm::Element< TVR, VM::VM1_2 >, 468
gdcm::Element< TVR, VM::VM2_2n >, 481
gdcm::Element< TVR, VM::VM2_n >, 486
gdcm::Element< TVR, VM::VM3_3n >, 493
gdcm::Element< TVR, VM::VM3_4 >, 498
gdcm::Element< TVR, VM::VM3_n >, 504
gdcm::Element< VR::AS, VM::VM5 >, 508
gdcm::Element< VR::OB, VM::VM1 >, 513
gdcm::Element< VR::OW, VM::VM1 >, 518
gdcm::LookupTable, 783
gdcm::UI, 1281
InternalCode
gdcm::Coder, 311
gdcm::JPEG12Codec, 731
gdcm::JPEG16Codec, 736
gdcm::JPEG8Codec, 749
Internals
vtkRTStructSetProperties, 1527
INVALID
gdcm::VR, 1405
Invalid
gdcm::Overlay, 891
gdcm::Usage, 1378
InverseRescale
gdcm::Rescaler, 1037
InverseRescaleFunctionIntoBestFit
gdcm::Rescaler, 1037
InvokeEvent
gdcm::Subject, 1197
IOD
gdcm::IOD, 707
IODEntry
gdcm::IODEntry, 709
IODMapType
gdcm::IODs, 712
IODMapTypeConstIterator
gdcm::IODs, 712
IODName
gdcm::IODs, 712
IODs
gdcm::IODs, 712
IPPSorter
gdcm::IPPSorter, 716
IS
gdcm::VR, 1406
IsAETitleValid
gdcm::network::AAssociateRQPDU, 125
IsASCII
gdcm::VR, 1408
IsASCII2
gdcm::VR, 1409
IsBinary
gdcm::VR, 1409
IsBinary2
gdcm::VR, 1409
IsCompatible
gdcm::PixelFormat, 935
IsDual
gdcm::VR, 1409
IsEmpty
gdcm::Bitmap, 257
gdcm::ByteValue, 281
gdcm::CSAElement, 345
gdcm::CSAHeaderDict, 356
gdcm::Curve, 368
gdcm::DataElement, 377
gdcm::DataSet, 396
gdcm::Defs, 410
gdcm::Dict, 423
gdcm::Dicts, 438
gdcm::Filename, 583
gdcm::Macros, 790
gdcm::Modules, 834
gdcm::Overlay, 894
gdcm::Preamble, 965
gdcm::PrivateDict, 989
gdcm::SegmentHelper::BasicCodedEntry, 245
gdcm::SequenceOfItems, 1103
IsEncapsulated
gdcm::TransferSyntax, 1273
IsEncoded
gdcm::TransferSyntax, 1273
IsExplicit

gdcm::TransferSyntax, 1273
IsFrameEncoder
gdcm::ImageCodec, 662
gdcm::JPEG2000Codec, 743
gdcm::JPEGCodec, 757
gdcm::JPEGLSCodec, 765
gdcm::RLECodec, 1045
IsGroupLength
gdcm::Tag, 1248
IsGroupXX
gdcm::Tag, 1248
IsIdentical
gdcm::Filename, 583
IsIllegal
gdcm::Tag, 1249
IsImage
gdcm::MediaStorage, 802
IsImplicit
gdcm::TransferSyntax, 1273
IsInPixelData
gdcm::Overlay, 894
IsKey
gdcm::Scanner, 1055
gdcm::Scanner2, 1066
gdcm::StrictScanner, 1173
gdcm::StrictScanner2, 1184
IsLastFragment
gdcm::network::AAbortPDU, 114
gdcm::network::AAssociateACPDU, 118
gdcm::network::AAssociateRJPDU, 121
gdcm::network::AAssociateRQPDU, 126
gdcm::network::AReleaseRPPDU, 150
gdcm::network::AReleaseRQPDU, 153
gdcm::network::BasePDU, 231
gdcm::network::PDataTFPDU, 906
IsLossless
gdcm::PhotometricInterpretation, 929
gdcm::TransferSyntax, 1273
IsLossy
gdcm::Bitmap, 257
gdcm::ImageCodec, 663
gdcm::PhotometricInterpretation, 929
gdcm::TransferSyntax, 1273
IsOdd
gdcm::VL, 1396
IsPresentationContextAccepted
gdcm::ServiceClassUser, 1118
IsPrintable
gdcm::ByteValue, 282
IsPrivate
gdcm::Tag, 1249
IsPrivateCreator
gdcm::Tag, 1249
IsPublic
gdcm::Tag, 1249
IsRetired
gdcm::PhotometricInterpretation, 929
IsRGB8
gdcm::LookupTable, 781
IsRowEncoder
gdcm::ImageCodec, 663
gdcm::JPEG2000Codec, 743
gdcm::JPEGCodec, 757
gdcm::JPEGLSCodec, 765
gdcm::RLECodec, 1045
IsSameColorSpace
gdcm::PhotometricInterpretation, 929
IsStateSuspension
gdcm::JPEG12Codec, 731
gdcm::JPEG16Codec, 736
gdcm::JPEG8Codec, 749
gdcm::JPEGCodec, 757
IsSwap
gdcm::VR, 1409
IsTransferSyntaxCompatible
gdcm::Bitmap, 257
IsUndefined
gdcm::MediaStorage, 803
gdcm::VL, 1396
IsUndefinedLength
gdcm::DataElement, 377
gdcm::SequenceOfItems, 1103
IsUnique
gdcm::DictEntry, 430
IsValid
gdcm::ApplicationEntity, 148
gdcm::BoxRegion, 271
gdcm::CodeString, 315
gdcm::DirectionCosines, 443
gdcm::DPath, 452
gdcm::FileMetaInformation, 578
gdcm::ImageCodec, 663
gdcm::JPEGCodec, 757
gdcm::LO, 775
gdcm::PixelFormat, 935
gdcm::Preamble, 965
gdcm::Region, 1034
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1189
gdcm::TagPath, 1256
gdcm::TransferSyntax, 1274
gdcm::UIDGenerator, 1283
gdcm::UUIDGenerator, 1384
gdcm::VM, 1402
gdcm::VR, 1409
IsVRFile
gdcm::VR, 1409
IsZero

gdcm::Overlay, 895
Item
gdcm::Item, 723
Items
gdcm::SequenceOfItems, 1106
ItemVector
gdcm::SequenceOfItems, 1100
Iterator
gdcm::CSAHeaderDict, 355
gdcm::DataSet, 391
gdcm::Dict, 421
gdcm::SequenceOfFragments, 1092
gdcm::SequenceOfItems, 1100
iterator
gdcm::CodeString, 314
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1188
ItFileSetHt
gdcm::SerieHelper, 1111
IVOCTForPresentation
gdcm::MediaStorage, 800
IVOCTForProcessing
gdcm::MediaStorage, 800

Join
gdcm::Filename, 583
JPEG12Codec
gdcm::JPEG12Codec, 731
JPEG16Codec
gdcm::JPEG16Codec, 736
JPEG2000
gdcm::TransferSyntax, 1271
JPEG2000_COMPRESSION
vtkGDCMImageWriter, 1450
JPEG2000Codec
gdcm::JPEG2000Codec, 740
JPEG2000Lossless
gdcm::TransferSyntax, 1271
JPEG2000Part2
gdcm::TransferSyntax, 1271
JPEG2000Part2Lossless
gdcm::TransferSyntax, 1271
JPEG8Codec
gdcm::JPEG8Codec, 749
JPEG_COMPRESSION
vtkGDCMImageWriter, 1450
JPEGBaselineProcess1
gdcm::TransferSyntax, 1271
JPEGCodec
gdcm::JPEGCodec, 754
JPEGExtendedProcess2_4
gdcm::TransferSyntax, 1271
JPEGExtendedProcess3_5
gdcm::TransferSyntax, 1271
JPEGFullProgressionProcess10_12
gdcm::TransferSyntax, 1271
JPEGLosslessProcess14
gdcm::TransferSyntax, 1271
JPEGLosslessProcess14_1
gdcm::TransferSyntax, 1271
JPEGLS_COMPRESSION
vtkGDCMImageWriter, 1450
JPEGLSCodec
gdcm::JPEGLSCodec, 762
JPEGLSLossless
gdcm::TransferSyntax, 1271
JPEGLSNearLossless
gdcm::TransferSyntax, 1271
JPEGSpectralSelectionProcess6_8
gdcm::TransferSyntax, 1271
JPIPReferenced
gdcm::TransferSyntax, 1271
JSON
gdcm::JSON, 767
JunkAfterDocElementError
gdcm::Parser, 902

KAKADUCodec
gdcm::KAKADUCodec, 771
KeyField
gdcm::CSAElement, 348
KeyObjectSelectionDocument
gdcm::MediaStorage, 799
KeyValuePairArrayType
gdcm::CompositeNetworkFunctions, 326
KeyValuePairType
gdcm::CompositeNetworkFunctions, 326
KODAK
gdcm::EquipmentManufacturer, 532

LD_ALL
gdcm, 91
LD_NOSEQ
gdcm, 91
LD_NOSHADOW
gdcm, 91
LD_NOSHADOWSEQ
gdcm, 91
LeadECGWaveformStorage
gdcm::MediaStorage, 798
LegacyConvertedEnhancedCTImageStorage
gdcm::MediaStorage, 800
LegacyConvertedEnhancedMRImageStorage
gdcm::MediaStorage, 800
LegacyConvertedEnhancedPETImageStorage
gdcm::MediaStorage, 800
Level
vtkImageMapToWindowLevelColors2, 1507

LINE
gdcm::MeshPrimitive, 814
ListCharSets
gdcm::QueryFactory, 1008
LittleEndian
gdcm::SwapCode, 1226
LO
gdcm::LO, 775
gdcm::VR, 1406
Load
gdcm::Directory, 447
gdcm::MrProtocol, 844
LOADBULKDATA
gdcm::XMLPrinter, 1543
LoadDefault
gdcm::CSAHeaderDict, 356
gdcm::Dict, 423
gdcm::PrivateDict, 989
LoadDefaults
gdcm::Defs, 411
gdcm::Dicts, 438
LoadFromDataElement
gdcm::CSAHeader, 353
gdcm::PDBHeader, 912
LoadFromFile
gdcm::Defs, 411
LoadIconImage
vtkGDCMImageReader, 1432
vtkGDCMImageReader2, 1446
LoadImageFromFiles
gdcm::DirectoryHelper, 450
LoadOverlays
vtkGDCMImageReader, 1432
vtkGDCMImageReader2, 1447
LoadResourcesFiles
gdcm::Global, 617
LoadSingleFile
vtkGDCMImageReader, 1422
vtkGDCMImageReader2, 1437
Locate
gdcm::Global, 617
LOComp
gdcm, 87
LodModeType
gdcm, 90
LookupTable
gdcm::LookupTable, 778, 779
vtkImageMapToColors16, 1503
LookupTableType
gdcm::LookupTable, 778
LossyFlag
gdcm::Bitmap, 263
gdcm::ImageCodec, 666
vtkGDCMImageReader, 1432
vtkGDCMImageReader2, 1447
LT
gdcm::VR, 1406
LTComp
gdcm, 87
LUT
gdcm::Bitmap, 263
gdcm::ImageCodec, 666
LUTPtr
gdcm::Bitmap, 253
gdcm::ImageCodec, 658

m_char
gdcm::ignore_char, 628
m_ConstMemberFunction
gdcm::MemberCommand< T >, 810
m_DataSet
gdcm::DataSetEvent, 404
m_MemberFunction
gdcm::MemberCommand< T >, 810
gdcm::SimpleMemberCommand< T >, 1129
m_This
gdcm::MemberCommand< T >, 810
gdcm::SimpleMemberCommand< T >, 1129
Macro
gdcm::Macro, 787
MacroEntry
gdcm, 87
Macros
gdcm::Macros, 789
mAction
gdcm::network::Transition, 1278
magenta
gdcm::terminal, 110
MAGNIFIED
gdcm::Spacing, 1147
MakeDirectory
gdcm::System, 1233
MakeNew
gdcm::network::Transition, 1278
MakeObject
gdcm::AnonymizeEvent, 134
gdcm::DataEvent, 387
gdcm::DataSetEvent, 403
gdcm::Event, 535
gdcm::FileNameEvent, 587
gdcm::ProgressEvent, 998
MammographyCADSR
gdcm::MediaStorage, 799
Mandatory
gdcm::Usage, 1378
MANUAL
gdcm::Segment, 1071
MapCSAHeaderDictEntry

gdcm::CSAHeaderDict, 355
MapDictEntry
gdcm::Dict, 421
MapIODEntry
gdcm::IOD, 706
MapModuleEntry
gdcm::Macro, 786
gdcm::Module, 826
MappingType
gdcm::Scanner, 1051
gdcm::StrictScanner, 1169
MapScalarsThroughTable2
vtkLookupTable16, 1516
MapTableEntry
gdcm::Table, 1237
MARCONI
gdcm::EquipmentManufacturer, 532
Match
gdcm::DPath, 452
MaximumLengthSub
gdcm::network::MaximumLengthSub, 791
MaxLength
gdcm::ApplicationEntity, 149
gdcm::PersonName, 921
MaxNumberOfComponents
gdcm::ApplicationEntity, 149
gdcm::PersonName, 921
MaxPrintLength
gdcm::Printer, 987
mConnection
gdcm::network::ULConnectionManager, 1362
MD5DataImagesType
gdcm::Testing, 1258
MD5MetaImagesType
vtkGDCMTesting, 1470
mDataSet
gdcm::BaseQuery, 237
MediaStorage
gdcm::MediaStorage, 801
MediaStorageAndFileFormat Directory Reference, 62
MediaStorageDataFilesType
gdcm::Testing, 1258
MediaStorageDirectoryStorage
gdcm::MediaStorage, 798
MedicalImageProperties
vtkGDCMImageReader, 1432
vtkGDCMPolyDataReader, 1463
vtkGDCMPolyDataWriter, 1468
mElementOffsets
gdcm::StreamImageWriter, 1164
mElementOffsets1
gdcm::StreamImageWriter, 1164
MemberCommand
gdcm::MemberCommand< T >, 808
mEnd
gdcm::network::Transition, 1278
MeshPrimitive
gdcm::MeshPrimitive, 814
MessageExchangeDefinition Directory Reference, 65
MessageID
gdcm::network::CEchoRQ, 292
MetaInformationTS
gdcm::FileMetaInformation, 580
mHelpDescription
gdcm::BaseRootQuery, 241
mImage
gdcm::BaseRootQuery, 241
mImplicit
gdcm::network::ULConnectionCallback, 1353
ModalityPerformedProcedureStepCreateQuery
gdcm::ModalityPerformedProcedureStepCreateQuery, 819
ModalityPerformedProcedureStepSetQuery
gdcm::ModalityPerformedProcedureStepSetQuery, 822
ModalityPerformedProcedureStepSOPClass
gdcm::MediaStorage, 799
Mode
gdcm::terminal, 110
Module
gdcm::Module, 826
ModuleEntry
gdcm::ModuleEntry, 830
ModuleMapType
gdcm::Macros, 789
gdcm::Modules, 833
Modules
gdcm::Modules, 833
MONOCHROME1
gdcm::PhotometricInterpretation, 927
MONOCHROME2
gdcm::PhotometricInterpretation, 927
MovePatientRootQuery
gdcm::MovePatientRootQuery, 837
MoveStudyRootQuery
gdcm::MoveStudyRootQuery, 841
mPatient
gdcm::BaseRootQuery, 242
MPEG2MainProfile
gdcm::TransferSyntax, 1271
MPEG2MainProfileHighLevel
gdcm::TransferSyntax, 1271
MPEG4AVCH264BDcompatibleHighProfileLevel4_1
gdcm::TransferSyntax, 1272
MPEG4AVCH264HighProfileLevel4_1
gdcm::TransferSyntax, 1271
MPType
gdcm::MeshPrimitive, 813

MPType_END
gdcm::MeshPrimitive, 814
MRImageStorage
gdcm::MediaStorage, 798
mRootType
gdcm::BaseRootQuery, 242
MrProtocol
gdcm::MrProtocol, 843
MRSpectroscopyStorage
gdcm::MediaStorage, 798
MS_END
gdcm::MediaStorage, 800
mSecondaryConnection
gdcm::network::ULConnectionManager, 1362
mSeries
gdcm::BaseRootQuery, 242
mSopInstanceUID
gdcm::BaseQuery, 237
mspFile
gdcm::StreamImageWriter, 1164
mStudy
gdcm::BaseRootQuery, 242
MSType
gdcm::MediaStorage, 798
mTransitions
gdcm::network::ULConnectionManager, 1362
MultiframeGrayscaleByteSecondaryCaptureImageStorage
gdcm::MediaStorage, 798
MultiframeGrayscaleWordSecondaryCaptureImageStorage
gdcm::MediaStorage, 798
MultiframeSingleBitSecondaryCaptureImageStorage
gdcm::MediaStorage, 798
MultiframeTrueColorSecondaryCaptureImageStorage
gdcm::MediaStorage, 798
mWriter
gdcm::StreamImageWriter, 1165
mXMax
gdcm::StreamImageWriter, 1165
mXMin
gdcm::StreamImageWriter, 1165
mYMax
gdcm::StreamImageWriter, 1165
mYMin
gdcm::StreamImageWriter, 1165
mZMax
gdcm::StreamImageWriter, 1165
mZMin
gdcm::StreamImageWriter, 1165

N_ACTION_RQ
gdcm::network::DIMSE, 441
N_ACTION_RSP
gdcm::network::DIMSE, 441
N_CREATE_RQ
gdcm::network::DIMSE, 441
N_CREATE_RSP
gdcm::network::DIMSE, 441
N_DELETE_RQ
gdcm::network::DIMSE, 441
N_DELETE_RSP
gdcm::network::DIMSE, 441
N_EVENT_REPORT_RQ
gdcm::network::DIMSE, 441
N_EVENT_REPORT_RSP
gdcm::network::DIMSE, 441
N_GET_RQ
gdcm::network::DIMSE, 441
N_GET_RSP
gdcm::network::DIMSE, 441
N_SET_RQ
gdcm::network::DIMSE, 441
N_SET_RSP
gdcm::network::DIMSE, 441
NAction
gdcm::NormalizedNetworkFunctions, 866
Name
gdcm::ModuleEntry, 832
NameField
gdcm::CSAElement, 348
gdcm::PDBElement, 910
NCreate
gdcm::NormalizedNetworkFunctions, 867
NDelete
gdcm::NormalizedNetworkFunctions, 867
NeedByteSwap
gdcm::Bitmap, 263
gdcm::ImageCodec, 666
NeedOverlayCleanup
gdcm::ImageCodec, 666
NegociatedType
gdcm::TransferSyntax, 1270
NestedMacroEntries
gdcm, 87
NestedModuleEntries
gdcm::NestedModuleEntries, 856
NEventReport
gdcm::NormalizedNetworkFunctions, 867
New
gdcm::Anonymizer, 141
gdcm::Cleaner, 302
gdcm::FileChangeTransferSyntax, 560
gdcm::FileStreamer, 597
gdcm::MemberCommand< T >, 809
gdcm::Scanner, 1055
gdcm::Scanner2, 1066
gdcm::SequenceOfFragments, 1095
gdcm::SequenceOfItems, 1104
gdcm::ServiceClassUser, 1118

gdcm::SimpleMemberCommand< T >, 1128
gdcm::StrictScanner, 1173
gdcm::StrictScanner2, 1184
vtkGDCMImageReader, 1422
vtkGDCMImageReader2, 1437
vtkGDCMImageWriter, 1451
vtkGDCMMedicalImageProperties, 1458
vtkGDCMPolyDataReader, 1461
vtkGDCMPolyDataWriter, 1466
vtkGDCMTesting, 1471
vtkGDCMThreadedImageReader, 1476
vtkGDCMThreadedImageReader2, 1480
vtkImageColorViewer, 1490
vtkImageMapToColors16, 1499
vtkImageMapToWindowLevelColors2, 1505
vtkImagePlanarComponentsToComponents, 1509
vtkImageRGBToYBR, 1511
vtkImageYBRToRGB, 1513
vtkLookupTable16, 1516
vtkRTStructSetProperties, 1523
NGet
gdcm::NormalizedNetworkFunctions, 867
NO
gdcm::Surface, 1201
NO_COMPRESSION
vtkGDCMImageWriter, 1450
NoElementsError
gdcm::Parser, 902
NoError
gdcm::Parser, 902
NOMAGIC
gdcm::CSAHeader, 351
NoMemoryError
gdcm::Parser, 902
NoObject
gdcm::MediaStorage, 800
NoOfItemsField
gdcm::CSAElement, 348
Norm
gdcm::DirectionCosines, 444
Normal
gdcm::MrProtocol::Slice, 1133
Normalize
gdcm::DirectionCosines, 444
NSet
gdcm::NormalizedNetworkFunctions, 867
NuclearMedicineImageStorage
gdcm::MediaStorage, 799
NuclearMedicineImageStorageRetired
gdcm::MediaStorage, 798
NumberOfDimensions
gdcm::Bitmap, 263
gdcm::ImageCodec, 666
NumberOfIconImages
vtkGDCMImageReader, 1432
vtkGDCMImageReader2, 1447
NumberOfOverlays
vtkGDCMImageReader, 1432
vtkGDCMImageReader2, 1447
NumberOfSurfaces
gdcm::SurfaceWriter, 1224

OB
gdcm::VR, 1406
OB_OW
gdcm::VR, 1406
Object
gdcm::Object, 872
ObjectEnd
gdcm::MediaStorage, 801
ObjectType
gdcm::MediaStorage, 800
OBLIQUE
gdcm::Orientation, 886
OD
gdcm::VR, 1406
OF
gdcm::VR, 1406
Ofstream
gdcm::Writer, 1538
OL
gdcm::VR, 1406
OnlyUUID
gdcm::XMLPrinter, 1543
OPENSSL
gdcm::CryptoFactory, 336
OpenSSLCryptoFactory
gdcm::OpenSSLCryptoFactory, 875
OpenSSLCryptographicMessageSyntax
gdcm::OpenSSLCryptographicMessageSyntax, 878
OPENSSLP7
gdcm::CryptoFactory, 336
OpenSSLP7CryptoFactory
gdcm::OpenSSLP7CryptoFactory, 881
OpenSSLP7CryptographicMessageSyntax
gdcm::OpenSSLP7CryptographicMessageSyntax, 883
operator const char *
gdcm::ConstCharWrapper, 331
gdcm::Filename, 583
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1189
operator const double *
gdcm::DirectionCosines, 444
operator const std::vector< char > &
gdcm::ByteValue, 282
operator MSType
gdcm::MediaStorage, 803

operator ObjectType ∗
gdcm::SmartPointer< ObjectType >, 1136
operator PIType
gdcm::PhotometricInterpretation, 929
operator ScalarType
gdcm::PixelFormat, 935
operator SwapCode::SwapCodeType
gdcm::SwapCode, 1226
operator TSType
gdcm::TransferSyntax, 1274
gdcm::UIDs, 1301
operator TypeType
gdcm::Type, 1280
operator uint32_t
gdcm::VL, 1396
operator UsageType
gdcm::Usage, 1379
operator VMType
gdcm::VM, 1402
operator VRType
gdcm::VR, 1410
operator!=
gdcm, 92
gdcm::Attribute< Group, Element, TVR, TVM >, 164
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 192
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 199
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
gdcm::CodeString, 316
gdcm::PixelFormat, 936
gdcm::PrivateTag, 993
gdcm::Tag, 1250
operator<
gdcm::Attribute< Group, Element, TVR, TVM >, 164
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 192
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
gdcm::CSAElement, 345
gdcm::CSAHeaderDictEntry, 359
gdcm::DataElement, 377
gdcm::DPath, 452
gdcm::PrivateTag, 994
gdcm::Tag, 1250
operator<<
gdcm, 92–100
gdcm::BasicOffsetTable, 249
gdcm::CodeString, 316
gdcm::CommandDataSet, 324
gdcm::CSAElement, 347
gdcm::CSAHeader, 354
gdcm::CSAHeaderDict, 357
gdcm::CSAHeaderDictEntry, 360
gdcm::DataElement, 382
gdcm::DataSet, 400
gdcm::Dict, 424
gdcm::DictEntry, 432
gdcm::Dicts, 439
gdcm::Directory, 448
gdcm::DPath, 453
gdcm::File, 552
gdcm::FileMetaInformation, 580
gdcm::FileSet, 593
gdcm::Fragment, 614
gdcm::Global, 618
gdcm::GroupDict, 621
gdcm::IOD, 708
gdcm::IODEntry, 711
gdcm::IODs, 713
gdcm::Item, 725
gdcm::Macro, 788
gdcm::Macros, 790
gdcm::MediaStorage, 804
gdcm::Module, 828
gdcm::ModuleEntry, 832
gdcm::Modules, 834
gdcm::MrProtocol, 845
gdcm::NestedModuleEntries, 857
gdcm::Object, 873
gdcm::Orientation, 888
gdcm::PDBElement, 910
gdcm::PDBHeader, 913
gdcm::PhotometricInterpretation, 930
gdcm::PixelFormat, 938

gdcm::Preamble, 966
gdcm::PrivateDict, 989
gdcm::PrivateTag, 995
gdcm::Scanner, 1057
gdcm::Scanner2, 1067
gdcm::Sorter, 1145
gdcm::StrictScanner, 1175
gdcm::StrictScanner2, 1185
gdcm::SwapCode, 1227
gdcm::Table, 1238
gdcm::Tag, 1254
gdcm::TransferSyntax, 1274
gdcm::Type, 1280
gdcm::UI, 1281
gdcm::Usage, 1379
gdcm::Version, 1393
gdcm::VL, 1398
gdcm::VM, 1403
gdcm::VR, 1410
operator<=
gdcm::Tag, 1250
operator>>
gdcm, 101
gdcm::Tag, 1254
operator()
gdcm::DataSet, 396
gdcm::Scanner2::ltstr, 784
gdcm::Scanner::ltstr, 784
gdcm::StrictScanner2::ltstr, 785
gdcm::StrictScanner::ltstr, 785
operator++
gdcm::VL, 1396
operator+=
gdcm::VL, 1396
operator->
gdcm::SmartPointer< ObjectType >, 1137
operator=
gdcm::AnonymizeEvent, 134
gdcm::ASN1, 156
gdcm::Base64, 225
gdcm::BoxRegion, 271
gdcm::ByteSwapFilter, 276
gdcm::ByteValue, 282
gdcm::Command, 319
gdcm::CryptographicMessageSyntax, 339
gdcm::CSAElement, 345
gdcm::CSAHeaderDict, 356
gdcm::DataElement, 378
gdcm::DataEvent, 387
gdcm::DataSet, 396
gdcm::DataSetEvent, 404
gdcm::Defs, 411
gdcm::Dict, 423
gdcm::Dicts, 439
gdcm::Element< TVR, VM::VM1_n >, 472
gdcm::Event, 535
gdcm::FileMetaInformation, 578
gdcm::FileNameEvent, 587
gdcm::Global, 618
gdcm::MemberCommand< T >, 809
gdcm::network::ULAction, 1305
gdcm::network::ULConnection, 1349
gdcm::network::UserInformation, 1382
gdcm::Object, 873
gdcm::Overlay, 895
gdcm::ParseException, 900
gdcm::Preamble, 965
gdcm::PrivateTag, 994
gdcm::ProgressEvent, 998
gdcm::SequenceOfItems, 1104
gdcm::ServiceClassUser, 1118
gdcm::SHA1, 1123
gdcm::SimpleMemberCommand< T >, 1128
gdcm::SimpleSubjectWatcher, 1130
gdcm::SmartPointer< ObjectType >, 1137
gdcm::Table, 1238
gdcm::Tag, 1250
operator==
gdcm, 101
gdcm::Attribute< Group, Element, TVR, TVM >, 164
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 179
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 192
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 205
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
gdcm::ByteValue, 282
gdcm::CodeString, 316
gdcm::CSAElement, 346
gdcm::DataElement, 378
gdcm::network::AbstractSyntax, 130
gdcm::network::PresentationContextRQ, 978
gdcm::network::TransferSyntaxSub, 1275
gdcm::PDBElement, 909
gdcm::PixelFormat, 936
gdcm::PresentationContext, 969
gdcm::PrivateTag, 994
gdcm::SequenceOfFragments, 1095

gdcm::SequenceOfItems, 1104
gdcm::Tag, 1250
gdcm::Value, 1389
operator[]
gdcm::Attribute< Group, Element, TVR, TVM >, 164, 165
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 192
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
gdcm::DataSet, 397
gdcm::Element< TVR, TVM >, 461
gdcm::Element< TVR, VM::VM1_2 >, 467
gdcm::Element< TVR, VM::VM1_n >, 472
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 485
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 497
gdcm::Element< TVR, VM::VM3_n >, 503
gdcm::Element< VR::AS, VM::VM5 >, 507
gdcm::Element< VR::OB, VM::VM1 >, 511
gdcm::Element< VR::OW, VM::VM1 >, 516
gdcm::Tag, 1250, 1251
operator∗
gdcm::SmartPointer< ObjectType >, 1137
OphthalmicPhotography16BitImageStorage
gdcm::MediaStorage, 800
OphthalmicPhotography8BitImageStorage
gdcm::MediaStorage, 800
OphthalmicTomographyImageStorage
gdcm::MediaStorage, 800
OrderFileList
gdcm::SerieHelper, 1110
Orientation
gdcm::Orientation, 886
OrientationType
gdcm::Orientation, 886
Output
gdcm::BitmapToBitmapFilter, 266
OutputFormat
vtkImageMapToColors16, 1503
OutputTypes
gdcm::DictConverter, 425
OV
gdcm::VR, 1406
Overlay
gdcm::Overlay, 891, 892
OverlayImageActor
vtkImageColorViewer, 1496
Overlays
gdcm::Pixmap, 946
OverlayType
gdcm::Overlay, 891
OW
gdcm::VR, 1406

Pack
gdcm::Unpacker12Bits, 1376
Padding
gdcm::ApplicationEntity, 149
gdcm::PersonName, 921
PALETTE_COLOR
gdcm::PhotometricInterpretation, 928
Parent
gdcm::Element< TVR, VM::VM1_2 >, 466
gdcm::Element< TVR, VM::VM2_2n >, 478
gdcm::Element< TVR, VM::VM2_n >, 484
gdcm::Element< TVR, VM::VM3_3n >, 490
gdcm::Element< TVR, VM::VM3_4 >, 496
gdcm::Element< TVR, VM::VM3_n >, 502
Parse
gdcm::Parser, 903
ParseBuffer
gdcm::Parser, 903
ParseCertificateFile
gdcm::CAPICryptographicMessageSyntax, 289
gdcm::CryptographicMessageSyntax, 340
gdcm::OpenSSLCryptographicMessageSyntax, 878
gdcm::OpenSSLP7CryptographicMessageSyntax, 884
ParseDateTime
gdcm::System, 1233, 1234
ParseDump
gdcm::ASN1, 156
ParseDumpFile
gdcm::ASN1, 156
ParseException
gdcm::ParseException, 899
ParseKeyFile
gdcm::CAPICryptographicMessageSyntax, 289
gdcm::CryptographicMessageSyntax, 340
gdcm::OpenSSLCryptographicMessageSyntax, 879
gdcm::OpenSSLP7CryptographicMessageSyntax, 884
Parser
gdcm::Parser, 902

PassAlphaToOutput
vtkImageMapToColors16, 1503
Patient
gdcm::Patient, 904
PDataTFPDU
gdcm::network::PDataTFPDU, 906
PDBElement
gdcm::PDBElement, 909
PDBHeader
gdcm::PDBHeader, 912
PDF
gdcm::MediaStorage, 801
PDFCodec
gdcm::PDFCodec, 915
PerformAction
gdcm::network::ULAction, 1305
gdcm::network::ULActionAA1, 1307
gdcm::network::ULActionAA2, 1308
gdcm::network::ULActionAA3, 1309
gdcm::network::ULActionAA4, 1311
gdcm::network::ULActionAA5, 1312
gdcm::network::ULActionAA6, 1313
gdcm::network::ULActionAA7, 1315
gdcm::network::ULActionAA8, 1316
gdcm::network::ULActionAE1, 1317
gdcm::network::ULActionAE2, 1319
gdcm::network::ULActionAE3, 1320
gdcm::network::ULActionAE4, 1321
gdcm::network::ULActionAE5, 1323
gdcm::network::ULActionAE6, 1324
gdcm::network::ULActionAE7, 1325
gdcm::network::ULActionAE8, 1327
gdcm::network::ULActionAR1, 1328
gdcm::network::ULActionAR10, 1329
gdcm::network::ULActionAR2, 1331
gdcm::network::ULActionAR3, 1332
gdcm::network::ULActionAR4, 1333
gdcm::network::ULActionAR5, 1335
gdcm::network::ULActionAR6, 1336
gdcm::network::ULActionAR7, 1337
gdcm::network::ULActionAR8, 1339
gdcm::network::ULActionAR9, 1340
gdcm::network::ULActionDT1, 1341
gdcm::network::ULActionDT2, 1343
PETImageStorage
gdcm::MediaStorage, 799
PF
gdcm::Bitmap, 263
gdcm::ImageCodec, 666
PGXCodec
gdcm::PGXCodec, 925
PHILIPS
gdcm::Dicts, 436
Philips3D
gdcm::MediaStorage, 799
PhilipsPrivateMRSyntheticImageStorage
gdcm::MediaStorage, 799
PhotometricInterpretation
gdcm::PhotometricInterpretation, 928
PI
gdcm::Bitmap, 263
gdcm::ImageCodec, 667
PI_END
gdcm::PhotometricInterpretation, 928
PIType
gdcm::PhotometricInterpretation, 927
PixelData
gdcm::Bitmap, 264
gdcm::PixmapReader, 950
gdcm::PixmapWriter, 958
PixelFormat
gdcm::PixelFormat, 933
Pixmap
gdcm::Pixmap, 943
PixmapReader
gdcm::Bitmap, 262
gdcm::PixmapReader, 949
PixmapToPixmapFilter
gdcm::PixmapToPixmapFilter, 952
PixmapWriter
gdcm::PixmapWriter, 956
PlanarConfiguration
gdcm::Bitmap, 264
gdcm::ImageCodec, 667
vtkGDCMImageReader, 1432
vtkGDCMImageReader2, 1447
PMS
gdcm::EquipmentManufacturer, 532
PN
gdcm::VR, 1406
PNComp
gdcm, 87
PNMCodec
gdcm::PNMCodec, 961
pointer
gdcm::CodeString, 314
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1188
POINTS
gdcm::Surface, 1202
Position
gdcm::MrProtocol::Slice, 1133
Preamble
gdcm::Preamble, 964
PrepareWrite
gdcm::PixmapWriter, 957
gdcm::SegmentWriter, 1088

gdcm::SurfaceWriter, 1224
PrepareWritePointMacro
gdcm::SurfaceWriter, 1224
Prepend
gdcm::Global, 618
PresentationContext
gdcm::PresentationContext, 968
PresentationContextAC
gdcm::network::PresentationContextAC, 971
PresentationContextArrayType
gdcm::network::AAssociateRQPDU, 124
gdcm::PresentationContextGenerator, 974
PresentationContextGenerator
gdcm::PresentationContextGenerator, 974
PresentationContextRQ
gdcm::network::PresentationContextRQ, 977
PresentationDataValue
gdcm::network::PresentationDataValue, 980
Preserve
gdcm::Cleaner, 302
PrettyPrintOff
gdcm::JSON, 767
PrettyPrintOn
gdcm::JSON, 768
PrimitiveData
gdcm::MeshPrimitive, 816
PrimitivesData
gdcm::MeshPrimitive, 813
PrimitiveType
gdcm::MeshPrimitive, 816
Print
gdcm::ApplicationEntity, 148
gdcm::Attribute< Group, Element, TVR, TVM >, 165
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 193
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
gdcm::BaseQuery, 235
gdcm::Bitmap, 257
gdcm::BoxRegion, 271
gdcm::ByteValue, 282
gdcm::CSAHeader, 353
gdcm::Curve, 368
gdcm::DataSet, 397
gdcm::DictPrinter, 435
gdcm::DirectionCosines, 444
gdcm::Directory, 448
gdcm::DPath, 453
gdcm::Element< TVR, TVM >, 461
gdcm::Element< TVR, VM::VM1_2 >, 467
gdcm::Element< TVR, VM::VM1_n >, 472
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 485
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 497
gdcm::Element< TVR, VM::VM3_n >, 503
gdcm::Element< VR::AS, VM::VM5 >, 507
gdcm::Element< VR::OB, VM::VM1 >, 512
gdcm::Element< VR::OW, VM::VM1 >, 517
gdcm::Event, 535
gdcm::Image, 635
gdcm::LookupTable, 782
gdcm::MEC_MR3, 794
gdcm::MrProtocol, 844
gdcm::network::AAbortPDU, 114
gdcm::network::AAssociateACPDU, 118
gdcm::network::AAssociateRJPDU, 121
gdcm::network::AAssociateRQPDU, 126
gdcm::network::AbstractSyntax, 130
gdcm::network::ApplicationContext, 146
gdcm::network::AReleaseRPPDU, 150
gdcm::network::AReleaseRQPDU, 153
gdcm::network::AsynchronousOperationsWindowSub, 158
gdcm::network::BasePDU, 231
gdcm::network::ImplementationClassUIDSub, 697
gdcm::network::ImplementationVersionNameSub, 699
gdcm::network::MaximumLengthSub, 791
gdcm::network::PDataTFPDU, 906
gdcm::network::PresentationContextAC, 971
gdcm::network::PresentationContextRQ, 978
gdcm::network::PresentationDataValue, 981
gdcm::network::RoleSelectionSub, 1047
gdcm::network::ServiceClassApplicationInformation, 1113
gdcm::network::SOPClassExtendedNegociationSub, 1138
gdcm::network::TransferSyntaxSub, 1275
gdcm::network::UserInformation, 1383
gdcm::Object, 873
gdcm::Orientation, 887
gdcm::Overlay, 895
gdcm::PDBHeader, 913
gdcm::PersonName, 920
gdcm::PixelFormat, 936

gdcm::Pixmap, 944
gdcm::Preamble, 965
gdcm::PresentationContext, 969
gdcm::Printer, 985
gdcm::Region, 1034
gdcm::Scanner, 1056
gdcm::Scanner2, 1066
gdcm::SegmentedPaletteColorLookupTable, 1080
gdcm::SequenceOfFragments, 1095
gdcm::SequenceOfItems, 1104
gdcm::Sorter, 1143
gdcm::StrictScanner, 1174
gdcm::StrictScanner2, 1184
gdcm::TagPath, 1256
gdcm::Testing, 1263
gdcm::Version, 1393
gdcm::XMLPrinter, 1543
PrintASCII
gdcm::ByteValue, 283
PrintASCIIXML
gdcm::ByteValue, 283
PrintAsContinuousString
gdcm::Tag, 1251
PrintAsContinuousUpperCaseString
gdcm::Tag, 1251
PrintAsPipeSeparatedString
gdcm::Tag, 1251
PrintDataElement
gdcm::Printer, 985
gdcm::XMLPrinter, 1544
PrintDataElement2
gdcm::DictPrinter, 435
PrintDataSet
gdcm::Printer, 986
gdcm::XMLPrinter, 1544
PrintDataSet2
gdcm::DictPrinter, 435
Printer
gdcm::Printer, 985
PrintGroupLength
gdcm::ByteValue, 283
PrintHex
gdcm::ByteValue, 283
PrintHexXML
gdcm::ByteValue, 283
PrintPNXML
gdcm::ByteValue, 283
PrintSelf
vtkGDCMImageReader, 1423
vtkGDCMImageReader2, 1437
vtkGDCMImageWriter, 1451
vtkGDCMMedicalImageProperties, 1458
vtkGDCMPolyDataReader, 1461
vtkGDCMPolyDataWriter, 1466
vtkGDCMTesting, 1472
vtkGDCMThreadedImageReader, 1476
vtkGDCMThreadedImageReader2, 1480
vtkImageColorViewer, 1490
vtkImageMapToColors16, 1499
vtkImageMapToWindowLevelColors2, 1505
vtkImagePlanarComponentsToComponents, 1509
vtkImageRGBToYBR, 1511
vtkImageYBRToRGB, 1513
vtkLookupTable16, 1516
vtkRTStructSetProperties, 1523
PrintSQ
gdcm::Printer, 986
gdcm::XMLPrinter, 1544
PrintStyle
gdcm::Printer, 987
gdcm::XMLPrinter, 1545
PrintStyles
gdcm::Printer, 985
gdcm::XMLPrinter, 1543
PrintTable
gdcm::network::ULTransitionTable, 1365
gdcm::Scanner, 1056
gdcm::Scanner2, 1066
gdcm::StrictScanner, 1174
gdcm::StrictScanner2, 1184
PrintXML
gdcm::PrivateDict, 989
PrivateBegin
gdcm::Scanner2, 1066
gdcm::StrictScanner2, 1184
PrivateConstIterator
gdcm::Scanner2, 1061
gdcm::StrictScanner2, 1179
PrivateDict
gdcm::PrivateDict, 988
PrivateEnd
gdcm::Scanner2, 1067
gdcm::StrictScanner2, 1185
PrivateMappingType
gdcm::Scanner2, 1061
gdcm::StrictScanner2, 1179
PrivateTag
gdcm::PrivateTag, 993
PrivateTagToValue
gdcm::Scanner2, 1061
gdcm::StrictScanner2, 1179
PrivateTagToValueValueType
gdcm::Scanner2, 1061
gdcm::StrictScanner2, 1179
Process
gdcm::Parser, 903
ProcessDataSet
gdcm::FileExplicitFilter, 570

ProcessPrivateTag
gdcm::Scanner2, 1067
gdcm::StrictScanner2, 1185
ProcessPublicTag
gdcm::Scanner, 1056
gdcm::Scanner2, 1067
gdcm::StrictScanner, 1174
gdcm::StrictScanner2, 1185
ProcessRequest
vtkGDCMImageReader2, 1438
ProduceCharacterSetDataElement
gdcm::QueryFactory, 1008
ProduceQuery
gdcm::QueryFactory, 1009
ProgressEvent
gdcm::ProgressEvent, 997
PropertyCategory
gdcm::Segment, 1075
PropertyType
gdcm::Segment, 1075
PropertyTypeModifiers
gdcm::Segment, 1075
PublicConstIterator
gdcm::Scanner2, 1061
gdcm::StrictScanner2, 1179
PublicMappingType
gdcm::Scanner2, 1061
gdcm::StrictScanner2, 1179
PublicTagToValue
gdcm::Scanner2, 1061
gdcm::StrictScanner2, 1179
PublicTagToValueValueType
gdcm::Scanner2, 1061
gdcm::StrictScanner2, 1179
Push
gdcm::TagPath, 1256
PushBackFile
vtkGDCMMedicalImageProperties, 1458
PVRGCodec
gdcm::PVRGCodec, 1002
Python Directory Reference, 67
PythonFilter
gdcm::PythonFilter, 1004

Quality
gdcm::JPEGCodec, 759
QueryFactory
gdcm::BaseQuery, 236
gdcm::BaseRootQuery, 241
gdcm::FindPatientRootQuery, 605
gdcm::FindStudyRootQuery, 609
gdcm::ModalityPerformedProcedureStepCreateQuery, 820
gdcm::ModalityPerformedProcedureStepSetQuery, 823
gdcm::MovePatientRootQuery, 838
gdcm::MoveStudyRootQuery, 842
gdcm::WLMFindQuery, 1533

RAWCodec
gdcm::RAWCodec, 1022
RawDataStorage
gdcm::MediaStorage, 799
Read
gdcm::BasicOffsetTable, 249
gdcm::ByteValue, 283, 284
gdcm::CommandDataSet, 323
gdcm::CP246ExplicitDataElement, 334
gdcm::DataElement, 378
gdcm::DataSet, 397
gdcm::Element< TVR, TVM >, 461
gdcm::Element< TVR, VM::VM1_2 >, 467
gdcm::Element< TVR, VM::VM1_n >, 472
gdcm::Element< TVR, VM::VM2_2n >, 479
gdcm::Element< TVR, VM::VM2_n >, 485
gdcm::Element< TVR, VM::VM3_3n >, 491
gdcm::Element< TVR, VM::VM3_4 >, 497
gdcm::Element< TVR, VM::VM3_n >, 503
gdcm::Element< VR::AS, VM::VM5 >, 507
gdcm::Element< VR::OB, VM::VM1 >, 512
gdcm::Element< VR::OW, VM::VM1 >, 517
gdcm::EncodingImplementation< VR::VRASCII >, 526
gdcm::EncodingImplementation< VR::VRBINARY >, 528
gdcm::ExplicitDataElement, 542
gdcm::ExplicitImplicitDataElement, 546
gdcm::File, 551
gdcm::FileMetaInformation, 578
gdcm::Fragment, 613
gdcm::ImageReader, 683
gdcm::ImageRegionReader, 688
gdcm::ImplicitDataElement, 703
gdcm::Item, 724
gdcm::network::AAbortPDU, 114
gdcm::network::AAssociateACPDU, 118
gdcm::network::AAssociateRJPDU, 121
gdcm::network::AAssociateRQPDU, 126
gdcm::network::AbstractSyntax, 130
gdcm::network::ApplicationContext, 146
gdcm::network::AReleaseRPPDU, 151
gdcm::network::AReleaseRQPDU, 153
gdcm::network::AsynchronousOperationsWindowSub, 158
gdcm::network::BasePDU, 231
gdcm::network::ImplementationClassUIDSub, 697

gdcm::network::ImplementationVersionNameSub, 699
gdcm::network::MaximumLengthSub, 791
gdcm::network::PDataTFPDU, 906
gdcm::network::PresentationContextAC, 972
gdcm::network::PresentationContextRQ, 978
gdcm::network::PresentationDataValue, 981
gdcm::network::RoleSelectionSub, 1047
gdcm::network::ServiceClassApplicationInformation, 1113
gdcm::network::SOPClassExtendedNegociationSub, 1138
gdcm::network::TransferSyntaxSub, 1275
gdcm::network::UserInformation, 1383
gdcm::PGXCodec, 926
gdcm::PixmapReader, 949
gdcm::PNMCodec, 962
gdcm::Preamble, 965
gdcm::Reader, 1028
gdcm::SegmentReader, 1083
gdcm::SequenceOfFragments, 1096
gdcm::SequenceOfItems, 1104
gdcm::StreamImageReader, 1158
gdcm::SurfaceReader, 1218
gdcm::TableReader, 1242
gdcm::Tag, 1251
gdcm::UNExplicitDataElement, 1371
gdcm::UNExplicitImplicitDataElement, 1375
gdcm::ValueIO< TDE, TSwap, TType >, 1390
gdcm::VL, 1397
gdcm::VR, 1410
gdcm::VR16ExplicitDataElement, 1413
gdcm::VRVLSize< 0 >, 1416
gdcm::VRVLSize< 1 >, 1418
Read16
gdcm::VL, 1397
ReadACRNEMAImage
gdcm::ImageReader, 683
gdcm::PixmapReader, 949
ReadBacktrack
gdcm::Fragment, 613
ReadCompat
gdcm::FileMetaInformation, 578
ReadCompatInternal
gdcm::FileMetaInformation, 578
ReadComputeLength
gdcm::EncodingImplementation< VR::VRASCII >, 526
gdcm::EncodingImplementation< VR::VRBINARY >, 528
ReadDataSet
gdcm::Reader, 1028
Reader
gdcm::Reader, 1027
ReadFiles
vtkGDCMThreadedImageReader, 1476
ReadFromCommaSeparatedString
gdcm::PrivateTag, 994
gdcm::Tag, 1251
ReadFromContinuousString
gdcm::Tag, 1252
ReadFromPipeSeparatedString
gdcm::Tag, 1252
ReadImage
gdcm::ImageReader, 683
gdcm::PixmapReader, 950
ReadImageInformation
gdcm::StreamImageReader, 1158
ReadImageInternal
gdcm::PixmapReader, 950
ReadInformation
gdcm::ImageRegionReader, 688
ReadInto
gdcm::network::PDataTFPDU, 907
gdcm::network::PresentationDataValue, 981
ReadIntoBuffer
gdcm::ImageRegionReader, 688
README.txt, 1549
ReadMetaInformation
gdcm::Reader, 1028
ReadNested
gdcm::DataSet, 397
ReadNoSwap
gdcm::EncodingImplementation< VR::VRASCII >, 526
gdcm::EncodingImplementation< VR::VRBINARY >, 528
ReadOrSkip
gdcm::DataElement, 378
ReadPointMacro
gdcm::SurfaceReader, 1219
ReadPreamble
gdcm::Reader, 1028
ReadPreValue
gdcm::CP246ExplicitDataElement, 334
gdcm::DataElement, 378
gdcm::ExplicitDataElement, 542
gdcm::ExplicitImplicitDataElement, 546
gdcm::Fragment, 613
gdcm::ImplicitDataElement, 703
gdcm::SequenceOfFragments, 1096
gdcm::UNExplicitDataElement, 1371
gdcm::UNExplicitImplicitDataElement, 1375
gdcm::VR16ExplicitDataElement, 1414
ReadSegment
gdcm::SegmentReader, 1084
ReadSegments
gdcm::SegmentReader, 1084

ReadSelectedPrivateTags
gdcm::DataSet, 397
gdcm::Reader, 1029
ReadSelectedPrivateTagsWithLength
gdcm::DataSet, 397
ReadSelectedTags
gdcm::DataSet, 398
gdcm::Reader, 1029
ReadSelectedTagsWithLength
gdcm::DataSet, 398
ReadSurface
gdcm::SurfaceReader, 1219
ReadSurfaces
gdcm::SurfaceReader, 1219
Readuint16
gdcm::DictConverter, 427
ReadUpToTag
gdcm::DataSet, 398
gdcm::Reader, 1029
ReadUpToTagWithLength
gdcm::DataSet, 398
ReadValue
gdcm::CP246ExplicitDataElement, 334
gdcm::DataElement, 379
gdcm::ExplicitDataElement, 543
gdcm::ExplicitImplicitDataElement, 547
gdcm::Fragment, 614
gdcm::ImplicitDataElement, 703
gdcm::SequenceOfFragments, 1096
gdcm::UNExplicitDataElement, 1371
gdcm::UNExplicitImplicitDataElement, 1375
gdcm::VR16ExplicitDataElement, 1414
ReadValueWithLength
gdcm::DataElement, 379
gdcm::ImplicitDataElement, 703
ReadVM
gdcm::DictConverter, 427
ReadVR
gdcm::DictConverter, 427
ReadWithLength
gdcm::CP246ExplicitDataElement, 334
gdcm::DataElement, 379
gdcm::DataSet, 398
gdcm::ExplicitDataElement, 543
gdcm::ExplicitImplicitDataElement, 547
gdcm::ImplicitDataElement, 703
gdcm::UNExplicitDataElement, 1371
gdcm::VR16ExplicitDataElement, 1414
RealWorldValueIntercept
gdcm::RealWorldValueMappingContent, 1032
RealWorldValueSlope
gdcm::RealWorldValueMappingContent, 1032
RecommendedDisplayCIELabToRGB
gdcm::SurfaceHelper, 1212
RecurseDataSet
gdcm::Anonymizer, 141
RED
gdcm::LookupTable, 778
red
gdcm::terminal, 110
reference
gdcm::CodeString, 314
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1188
ReferenceFrameOfReferenceUID
vtkRTStructSetProperties, 1527
ReferenceSeriesInstanceUID
vtkRTStructSetProperties, 1527
Region
gdcm::Region, 1033
Register
gdcm::Object, 873
Remove
gdcm::Anonymizer, 141
gdcm::Cleaner, 302, 303
gdcm::DataSet, 399
gdcm::FileAnonymizer, 555
gdcm::Preamble, 966
RemoveAllGroupLength
gdcm::Cleaner, 303
RemoveAllIllegal
gdcm::Cleaner, 303
RemoveAllMissingPrivateCreator
gdcm::Cleaner, 303
RemoveAllObservers
gdcm::Subject, 1198
RemoveDictEntry
gdcm::PrivateDict, 989
RemoveFile
gdcm::System, 1234
RemoveGroupLength
gdcm::Anonymizer, 141
RemoveItemByIndex
gdcm::SequenceOfItems, 1105
RemoveMissingPrivateCreator
gdcm::Cleaner, 303
RemoveObserver
gdcm::Subject, 1198
RemoveOverlay
gdcm::Pixmap, 944
RemovePrivateTags
gdcm::Anonymizer, 141
RemoveRetired
gdcm::Anonymizer, 142
Render
vtkImageColorViewer, 1490
Renderer

vtkImageColorViewer, 1496
RenderWindow
vtkImageColorViewer, 1496
Replace
gdcm::Anonymizer, 142
gdcm::CommandDataSet, 323
gdcm::DataSet, 399
gdcm::FileAnonymizer, 556
gdcm::FileMetaInformation, 579
ReplaceEmpty
gdcm::DataSet, 399
RequestData
vtkGDCMImageReader2, 1438
vtkGDCMPolyDataReader, 1462
vtkImageMapToColors16, 1499
vtkImageMapToWindowLevelColors2, 1505
vtkImagePlanarComponentsToComponents, 1509
RequestData_HemodynamicWaveformStorage
vtkGDCMPolyDataReader, 1462
RequestData_RTStructureSetStorage
vtkGDCMPolyDataReader, 1462
RequestDataCompat
vtkGDCMImageReader, 1423
vtkGDCMImageReader2, 1438
vtkGDCMThreadedImageReader, 1476
RequestInformation
vtkGDCMImageReader2, 1438
vtkGDCMPolyDataReader, 1462
vtkGDCMThreadedImageReader2, 1480
vtkImageMapToColors16, 1500
vtkImageMapToWindowLevelColors2, 1505
RequestInformation_HemodynamicWaveformStorage
vtkGDCMPolyDataReader, 1462
RequestInformation_RTStructureSetStorage
vtkGDCMPolyDataReader, 1462
RequestInformationCompat
vtkGDCMImageReader, 1423
vtkGDCMImageReader2, 1438
RequestPaddedCompositePixelCode
gdcm::ImageCodec, 667
RequestPlanarConfiguration
gdcm::ImageCodec, 667
Rescale
gdcm::Rescaler, 1037
RescaleFunctionIntoBestFit
gdcm::Rescaler, 1037
Rescaler
gdcm::Rescaler, 1036
ReserveDataElement
gdcm::FileStreamer, 597
ReserveGroupDataElement
gdcm::FileStreamer, 597
reset
gdcm::terminal, 110
ResetHandledDataSet
gdcm::network::ULConnectionCallback, 1352
RetrieveSOPInstanceUIDFromIndex
gdcm::DirectoryHelper, 450
RetrieveSOPInstanceUIDFromZPosition
gdcm::DirectoryHelper, 451
reverse
gdcm::terminal, 110
reverse_iterator
gdcm::CodeString, 314
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1188
RGB
gdcm::PhotometricInterpretation, 928
RGB2YBR
gdcm::ImageChangePhotometricInterpretation, 644
RGBPixelsToRGBPlanes
gdcm::ImageChangePlanarConfiguration, 648
RGBPlanesToRGBPixels
gdcm::ImageChangePlanarConfiguration, 648
RGBToRecommendedDisplayCIELab
gdcm::SurfaceHelper, 1213
RGBToRecommendedDisplayGrayscale
gdcm::SurfaceHelper, 1214
RLE_COMPRESSION
vtkGDCMImageWriter, 1450
RLECodec
gdcm::RLECodec, 1042
RLELossless
gdcm::TransferSyntax, 1271
ROI
gdcm::Overlay, 891
RoleSelectionSub
gdcm::network::RoleSelectionSub, 1046
Round
gdcm, 101
roundat
gdcm, 101
RTDoseStorage
gdcm::MediaStorage, 799
RTImageStorage
gdcm::MediaStorage, 799
RTIonBeamsTreatmentRecordStorage
gdcm::MediaStorage, 800
RTIonPlanStorage
gdcm::MediaStorage, 799
RTPlanStorage
gdcm::MediaStorage, 799
RTStructSetProperties
vtkGDCMPolyDataReader, 1464
vtkGDCMPolyDataWriter, 1468
RTStructureSetStorage
gdcm::MediaStorage, 799

RTTreatmentSummaryRecordStorage
gdcm::MediaStorage, 800
Rule
gdcm::SerieHelper, 1108
RunEventLoop
gdcm::network::ULConnectionManager, 1359
RunMoveEventLoop
gdcm::network::ULConnectionManager, 1359

SAGITTAL
gdcm::Orientation, 886
SAMSUNG
gdcm::EquipmentManufacturer, 532
ScalarType
gdcm::PixelFormat, 932
Scale
vtkGDCMImageReader, 1433
vtkGDCMImageReader2, 1447
Scan
gdcm::Scanner, 1056
gdcm::Scanner2, 1067
gdcm::StrictScanner, 1174
gdcm::StrictScanner2, 1185
Scanner
gdcm::Scanner, 1052
Scanner2
gdcm::Scanner2, 1062
Scrub
gdcm::Cleaner, 304
SecondaryCaptureImageStorage
gdcm::MediaStorage, 798
Segment
gdcm::Segment, 1071
SegmentAlgorithmName
gdcm::Segment, 1076
SegmentAlgorithmType
gdcm::Segment, 1076
Segmentation
gdcm::MediaStorage, 801
SegmentationStorage
gdcm::MediaStorage, 799
SegmentDescription
gdcm::Segment, 1076
SegmentedPaletteColorLookupTable
gdcm::SegmentedPaletteColorLookupTable, 1079
SegmentLabel
gdcm::Segment, 1076
SegmentMap
gdcm::SegmentReader, 1083
SegmentNumber
gdcm::Segment, 1076
SegmentReader
gdcm::SegmentReader, 1083
Segments
gdcm::SegmentReader, 1084
gdcm::SegmentWriter, 1089
SegmentVector
gdcm::SegmentReader, 1083
gdcm::SegmentWriter, 1088
SegmentWriter
gdcm::SegmentWriter, 1088
Selection
gdcm::Sorter, 1145
SelectionMap
gdcm::Sorter, 1143
Self
gdcm::AnonymizeEvent, 133
gdcm::DataEvent, 386
gdcm::DataSetEvent, 402
gdcm::FileNameEvent, 586
gdcm::MemberCommand< T >, 808
gdcm::ProgressEvent, 997
gdcm::SimpleMemberCommand< T >, 1127
SEMIAUTOMATIC
gdcm::Segment, 1071
SendEcho
gdcm::network::ULConnectionManager, 1359
gdcm::ServiceClassUser, 1118
SendFind
gdcm::network::ULConnectionManager, 1359, 1360
gdcm::ServiceClassUser, 1118
SendMove
gdcm::network::ULConnectionManager, 1360
gdcm::ServiceClassUser, 1118, 1119
SendNAction
gdcm::network::ULConnectionManager, 1360
SendNCreate
gdcm::network::ULConnectionManager, 1360
SendNDelete
gdcm::network::ULConnectionManager, 1361
SendNEventReport
gdcm::network::ULConnectionManager, 1361
SendNGet
gdcm::network::ULConnectionManager, 1361
SendNSet
gdcm::network::ULConnectionManager, 1361
SendStore
gdcm::network::ULConnectionManager, 1362
gdcm::ServiceClassUser, 1119
Separator
gdcm::ApplicationEntity, 149
gdcm::PersonName, 921
SequenceLengthField
gdcm::SequenceOfItems, 1106
SequenceOfFragments
gdcm::SequenceOfFragments, 1093
SequenceOfItems
gdcm::SequenceOfItems, 1101

SerieHelper
gdcm::SerieHelper, 1109
SerieRestrictions
gdcm::SerieHelper, 1108
Series
gdcm::Series, 1112
SeriesInstanceUID
vtkRTStructSetProperties, 1527
ServiceClassApplicationInformation
gdcm::network::ServiceClassApplicationInformation, 1112
ServiceClassUser
gdcm::ServiceClassUser, 1117
Set
gdcm::Attribute< Group, Element, TVR, TVM >, 165
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 185
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 193
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 212
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
gdcm::Element< TVR, TVM >, 461
gdcm::Element< TVR, VM::VM1_2 >, 467
gdcm::Element< TVR, VM::VM1_n >, 472
gdcm::Element< TVR, VM::VM2_2n >, 480
gdcm::Element< TVR, VM::VM2_n >, 485
gdcm::Element< TVR, VM::VM3_3n >, 492
gdcm::Element< TVR, VM::VM3_4 >, 497
gdcm::Element< TVR, VM::VM3_n >, 503
gdcm::Element< VR::AS, VM::VM5 >, 507
gdcm::Element< VR::OB, VM::VM1 >, 512
gdcm::Element< VR::OW, VM::VM1 >, 517
SetAbstractSyntax
gdcm::network::PresentationContextRQ, 978
gdcm::PresentationContext, 969
SetAETitle
gdcm::ServiceClassUser, 1120
SetAlgorithmFamily
gdcm::Surface, 1207
SetAlgorithmName
gdcm::Surface, 1207
SetAlgorithmVersion
gdcm::Surface, 1207
SetAnatomicRegion
gdcm::Segment, 1073
SetAnatomicRegionModifiers
gdcm::Segment, 1074
SetAppendDerivationHistory
gdcm::FileDerivation, 567
SetArray
gdcm::Element< TVR, VM::VM1_n >, 473
setattribute
gdcm::terminal, 111
SetAxisOfRotation
gdcm::Surface, 1207
setbgcolor
gdcm::terminal, 111
SetBitPosition
gdcm::Overlay, 895
SetBitsAllocated
gdcm::Overlay, 895
gdcm::PixelFormat, 936
SetBitSample
gdcm::JPEGCodec, 757
SetBitsStored
gdcm::PixelFormat, 937
SetBlob
gdcm::ApplicationEntity, 148
gdcm::network::PresentationDataValue, 981
gdcm::PersonName, 920
SetBlueLUT
gdcm::LookupTable, 782
SetBufferLength
gdcm::JPEGLSCodec, 765
gdcm::PNMCodec, 962
gdcm::RLECodec, 1045
SetByteSwapTag
gdcm::ByteSwapFilter, 276
SetByteValue
gdcm::Attribute< Group, Element, TVR, TVM >, 165
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 193
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 213
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 218
gdcm::CSAElement, 346
gdcm::DataElement, 379

SetByteValueNoSwap
gdcm::Attribute< Group, Element, TVR, TVM >, 165
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 193
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 213
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 219
SetCallbackFunction
gdcm::MemberCommand< T >, 809, 810
gdcm::SimpleMemberCommand< T >, 1128
SetCalledAETitle
gdcm::network::AAssociateACPDU, 119
gdcm::network::AAssociateRQPDU, 126
gdcm::ServiceClassUser, 1120
SetCallingAETitle
gdcm::network::AAssociateACPDU, 119
gdcm::network::AAssociateRQPDU, 126
SetCenterOfRotation
gdcm::Surface, 1207
SetChangePrivateTags
gdcm::FileExplicitFilter, 570
SetCheckFileMetaInformation
gdcm::Writer, 1537
SetCipherType
gdcm::CAPICryptographicMessageSyntax, 289
gdcm::CryptographicMessageSyntax, 340
gdcm::OpenSSLCryptographicMessageSyntax, 879
gdcm::OpenSSLP7CryptographicMessageSyntax, 884
SetColor
gdcm::Printer, 986
SetColorLevel
vtkImageColorViewer, 1490
SetColorWindow
vtkImageColorViewer, 1490
SetColumns
gdcm::Bitmap, 258
gdcm::Overlay, 895
SetCommand
gdcm::network::PresentationDataValue, 981
SetComponents
gdcm::PersonName, 920, 921
SetCompressIconImage
gdcm::ImageChangeTransferSyntax, 653
SetComputeZSpacing
gdcm::IPPSorter, 717
SetCoordinateStartValue
gdcm::Curve, 368
SetCoordinateStepValue
gdcm::Curve, 368
SetCryptographicMessageSyntax
gdcm::Anonymizer, 143
SetCurve
gdcm::Curve, 368
vtkGDCMImageReader, 1423
vtkGDCMImageReader2, 1439
SetCurveDataDescriptor
gdcm::Curve, 368
SetCurveDescription
gdcm::Curve, 368
SetData
gdcm::DataEvent, 388
SetDataElement
gdcm::Bitmap, 258
SetDataSet
gdcm::File, 551
gdcm::network::PresentationDataValue, 981
SetDataSetTransferSyntax
gdcm::FileMetaInformation, 579
SetDataValueRepresentation
gdcm::Curve, 369
SetDebug
gdcm::Trace, 1266
SetDebugStream
gdcm::Trace, 1266
SetDefaultTransferSyntax
gdcm::PresentationContextGenerator, 975
SetDerivationCodeSequenceCodeValue
gdcm::FileDerivation, 567
SetDerivationDescription
gdcm::FileDerivation, 567
SetDescription
gdcm::CSAHeaderDictEntry, 360
gdcm::ModuleEntry, 831
gdcm::Overlay, 896
SetDescriptor
gdcm::DICOMDIRGenerator, 419
SetDictName
gdcm::DictConverter, 427
SetDicts
gdcm::PythonFilter, 1004
gdcm::StringFilter, 1192
SetDimension
gdcm::Bitmap, 258
SetDimensions
gdcm::Bitmap, 258
gdcm::Curve, 369

gdcm::ImageCodec, 663
SetDimensionsValue
gdcm::ImageHelper, 677
SetDirectionCosines
gdcm::Image, 635
vtkGDCMImageWriter, 1451
SetDirectionCosinesFromImageOrientationPatient
vtkGDCMImageWriter, 1451
SetDirectionCosinesTolerance
gdcm::IPPSorter, 717
SetDirectionCosinesValue
gdcm::ImageHelper, 677
SetDirectory
gdcm::network::ULWritingCallback, 1367
gdcm::SerieHelper, 1110
SetDisplayId
vtkImageColorViewer, 1490
SetDomain
gdcm::BoxRegion, 271
SetDropDuplicatePositions
gdcm::IPPSorter, 718
SetElement
gdcm::Tag, 1252
SetElementHandler
gdcm::Parser, 903
SetElementTag
gdcm::Tag, 1252, 1253
SetElementXX
gdcm::DictEntry, 430
SetError
gdcm::Trace, 1267
SetErrorStream
gdcm::Trace, 1267
SetEvent
gdcm::network::ULEvent, 1364
setfgcolor
gdcm::terminal, 111
SetFile
gdcm::Anonymizer, 143
gdcm::Cleaner, 304
gdcm::DICOMDIRGenerator, 419
gdcm::FileDecompressLookupTable, 564
gdcm::FileDerivation, 567
gdcm::FileExplicitFilter, 570
gdcm::IconImageFilter, 624
gdcm::Printer, 986
gdcm::PythonFilter, 1004
gdcm::Reader, 1029
gdcm::SplitMosaicFilter, 1151
gdcm::StreamImageWriter, 1162
gdcm::StringFilter, 1192
gdcm::Validate, 1386
gdcm::Writer, 1537
gdcm::XMLPrinter, 1544
SetFileName
gdcm::FileNameEvent, 587
gdcm::Reader, 1029
gdcm::StreamImageReader, 1158
gdcm::StreamImageWriter, 1163
gdcm::Writer, 1537
vtkGDCMThreadedImageReader2, 1480
SetFilename
gdcm::TableReader, 1242
SetFileNames
vtkGDCMImageReader, 1423
vtkGDCMImageWriter, 1452
vtkGDCMThreadedImageReader2, 1480
SetFilenames
gdcm::DICOMDIRGenerator, 419
SetFilePattern
vtkGDCMImageReader, 1424
vtkGDCMImageReader2, 1439
SetFilePrefix
vtkGDCMImageReader, 1424
vtkGDCMImageReader2, 1439
SetFiles
gdcm::FileSet, 593
SetFiniteVolume
gdcm::Surface, 1208
SetForce
gdcm::ImageChangeTransferSyntax, 653
gdcm::ImageFragmentSplitter, 672
SetForcePixelSpacing
gdcm::ImageHelper, 677
SetForceRescaleInterceptSlope
gdcm::ImageHelper, 677
SetFragmentSizeMax
gdcm::ImageFragmentSplitter, 672
SetFrameOrigin
gdcm::Overlay, 896
SetFromDataElement
gdcm::Attribute< Group, Element, TVR, TVM >, 166
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 193
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 200
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 213
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 219

gdcm::Element< TVR, TVM >, 461
gdcm::Element< TVR, VM::VM1_2 >, 467
gdcm::Element< TVR, VM::VM1_n >, 473
gdcm::Element< TVR, VM::VM2_2n >, 480
gdcm::Element< TVR, VM::VM2_n >, 485
gdcm::Element< TVR, VM::VM3_3n >, 492
gdcm::Element< TVR, VM::VM3_4 >, 497
gdcm::Element< TVR, VM::VM3_n >, 503
gdcm::Element< VR::AS, VM::VM5 >, 507
gdcm::Element< VR::OB, VM::VM1 >, 512
gdcm::Element< VR::OW, VM::VM1 >, 517
SetFromDataSet
gdcm::Attribute< Group, Element, TVR, TVM >, 166
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 193
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 201
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 213
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 219
gdcm::MediaStorage, 803
SetFromFile
gdcm::MediaStorage, 803
SetFromHeader
gdcm::MediaStorage, 803
SetFromModality
gdcm::MediaStorage, 804
SetFromSourceImageSequence
gdcm::MediaStorage, 804
SetFromString
gdcm::DirectionCosines, 444
SetFromUID
gdcm::UIDs, 1301
SetGreenLUT
gdcm::LookupTable, 782
SetGroup
gdcm::Curve, 369
gdcm::Overlay, 896
gdcm::Tag, 1253
SetGroupXX
gdcm::DictEntry, 431
SetHeader
gdcm::File, 552
SetHighBit
gdcm::PixelFormat, 937
SetHostname
gdcm::ServiceClassUser, 1120
SetIconImage
gdcm::Pixmap, 945
SetIE
gdcm::IODEntry, 710
SetImage
gdcm::PixmapWriter, 957
gdcm::SplitMosaicFilter, 1151
SetImplementationClassUID
gdcm::FileMetaInformation, 579
SetImplementationVersionName
gdcm::FileMetaInformation, 579
SetImplicitFlag
gdcm::network::ULConnectionCallback, 1352
SetInput
gdcm::BitmapToBitmapFilter, 266
gdcm::ImageConverter, 668
vtkImageColorViewer, 1491
SetInputConnection
vtkImageColorViewer, 1491
SetInputDirectory
gdcm::EmptyMaskGenerator, 522
SetInputFileName
gdcm::DictConverter, 427
gdcm::FileAnonymizer, 556
gdcm::FileChangeTransferSyntax, 560
SetIntercept
gdcm::Image, 635
gdcm::Rescaler, 1038
SetKey
gdcm::CSAElement, 346
SetKeyword
gdcm::DictEntry, 431
SetLastElement
gdcm::ParseException, 900
SetLastFragment
gdcm::network::PresentationDataValue, 982
SetLength
gdcm::ByteValue, 284
gdcm::Element< TVR, VM::VM1_2 >, 467
gdcm::Element< TVR, VM::VM1_n >, 473
gdcm::Element< TVR, VM::VM2_2n >, 480
gdcm::Element< TVR, VM::VM2_n >, 485
gdcm::Element< TVR, VM::VM3_3n >, 492
gdcm::Element< TVR, VM::VM3_4 >, 497
gdcm::Element< TVR, VM::VM3_n >, 503
gdcm::RLECodec, 1045
gdcm::SequenceOfFragments, 1096
gdcm::SequenceOfItems, 1105
gdcm::Value, 1389
SetLengthOnly
gdcm::ByteValue, 284
gdcm::Value, 1389

SetLengthToUndefined
gdcm::SequenceOfItems, 1105
SetLoadMode
gdcm::SerieHelper, 1111
SetLookupTable
vtkImageMapToColors16, 1500
SetLossless
gdcm::JPEGCodec, 757
gdcm::JPEGLSCodec, 765
SetLossyError
gdcm::JPEGLSCodec, 765
SetLossyFlag
gdcm::Bitmap, 259
gdcm::ImageCodec, 663
gdcm::PVRGCodec, 1003
SetLUT
gdcm::Bitmap, 259
gdcm::ImageCodec, 663
gdcm::LookupTable, 782
gdcm::SegmentedPaletteColorLookupTable, 1080
SetManifold
gdcm::Surface, 1208
SetMaximumLength
gdcm::network::MaximumLengthSub, 791
SetMaximumPointDistance
gdcm::Surface, 1208
SetMaxPDULength
gdcm::network::ULConnectionInfo, 1354
SetMaxPDUSize
gdcm::network::ULConnection, 1349
SetMCT
gdcm::JPEG2000Codec, 743
SetMeanPointDistance
gdcm::Surface, 1208
SetMedicalImageProperties
vtkGDCMImageReader, 1424
vtkGDCMImageReader2, 1439
vtkGDCMImageWriter, 1452
vtkGDCMPolyDataWriter, 1467
SetMergeModeToAbstractSyntax
gdcm::PresentationContextGenerator, 975
SetMergeModeToTransferSyntax
gdcm::PresentationContextGenerator, 975
SetMeshPrimitive
gdcm::Surface, 1208
SetMessageHeader
gdcm::network::PresentationDataValue, 982
SetMinMaxForPixelType
gdcm::Rescaler, 1038
setmode
gdcm::terminal, 111
SetName
gdcm::CSAElement, 346
gdcm::CSAHeaderDictEntry, 360
gdcm::DictEntry, 431
gdcm::IODEntry, 710
gdcm::Macro, 787
gdcm::Module, 827
gdcm::ModuleEntry, 831
gdcm::network::AbstractSyntax, 130
gdcm::network::ApplicationContext, 146
gdcm::network::TransferSyntaxSub, 1275
gdcm::PDBElement, 909
SetNameFromUID
gdcm::network::AbstractSyntax, 130
gdcm::network::TransferSyntaxSub, 1276
SetNeedByteSwap
gdcm::Bitmap, 259
gdcm::ImageCodec, 664
SetNeedOverlayCleanup
gdcm::ImageCodec, 664
SetNestedDataSet
gdcm::Item, 725
SetNoOfItems
gdcm::CSAElement, 346
SetNoSwap
gdcm::Element< TVR, TVM >, 462
gdcm::Element< TVR, VM::VM1_2 >, 467
gdcm::Element< TVR, VM::VM1_n >, 473
gdcm::Element< TVR, VM::VM2_2n >, 480
gdcm::Element< TVR, VM::VM2_n >, 485
gdcm::Element< TVR, VM::VM3_3n >, 492
gdcm::Element< TVR, VM::VM3_4 >, 497
gdcm::Element< TVR, VM::VM3_n >, 503
gdcm::Element< VR::AS, VM::VM5 >, 507
gdcm::Element< VR::OB, VM::VM1 >, 512
gdcm::Element< VR::OW, VM::VM1 >, 517
SetNumberOfCurves
gdcm::Pixmap, 945
SetNumberOfDimensions
gdcm::Bitmap, 259
gdcm::ImageCodec, 664
SetNumberOfFilenames
gdcm::FilenameGenerator, 590
SetNumberOfFrames
gdcm::Overlay, 896
SetNumberOfInputPorts
vtkGDCMPolyDataWriter, 1467
SetNumberOfItems
gdcm::SequenceOfItems, 1105
SetNumberOfOverlays
gdcm::Pixmap, 945
SetNumberOfPoints
gdcm::Curve, 369
SetNumberOfResolutions
gdcm::JPEG2000Codec, 743
SetNumberOfSegments
gdcm::SegmentWriter, 1089

SetNumberOfSurfacePoints
gdcm::Surface, 1208
SetNumberOfSurfaces
gdcm::SurfaceWriter, 1224
SetNumberOfTableValues
vtkLookupTable16, 1516
SetNumberOfThreadsForDecompression
gdcm::JPEG2000Codec, 743
SetNumberOfValues
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 194
SetNumberOfVectors
gdcm::Surface, 1208
SetObliquityThresholdCosineValue
gdcm::Orientation, 887
SetOffScreenRendering
vtkImageColorViewer, 1491
SetOrigin
gdcm::Image, 635, 636
gdcm::Overlay, 896
SetOriginValue
gdcm::ImageHelper, 678
SetOutputDimensions
gdcm::IconImageGenerator, 626
SetOutputDirectory
gdcm::EmptyMaskGenerator, 522
SetOutputFileName
gdcm::DictConverter, 427
gdcm::FileAnonymizer, 556
gdcm::FileChangeTransferSyntax, 560
gdcm::FileStreamer, 597
SetOutputFormatToLuminance
vtkImageMapToColors16, 1500
SetOutputFormatToLuminanceAlpha
vtkImageMapToColors16, 1500
SetOutputFormatToRGB
vtkImageMapToColors16, 1500
SetOutputFormatToRGBA
vtkImageMapToColors16, 1500
SetOutputType
gdcm::DictConverter, 427
SetOutsideValuePixel
gdcm::IconImageGenerator, 626
SetOverlay
gdcm::Overlay, 896
SetOverlayVisibility
vtkImageColorViewer, 1491
SetOwner
gdcm::PrivateTag, 994
SetParentId
vtkImageColorViewer, 1491
SetPassword
gdcm::CAPICryptographicMessageSyntax, 289
gdcm::CryptographicMessageSyntax, 340
gdcm::OpenSSLCryptographicMessageSyntax, 879
gdcm::OpenSSLP7CryptographicMessageSyntax, 884
SetPattern
gdcm::FilenameGenerator, 590
SetPDU
gdcm::network::ULEvent, 1364
SetPermissions
gdcm::System, 1234
SetPhotometricInterpretation
gdcm::Bitmap, 259
gdcm::ImageChangePhotometricInterpretation, 644
gdcm::ImageCodec, 664
SetPixelFormat
gdcm::Bitmap, 260
gdcm::ImageCodec, 664
gdcm::JPEGCodec, 757
gdcm::Rescaler, 1038
SetPixelMinMax
gdcm::IconImageGenerator, 627
SetPixelRepresentation
gdcm::PixelFormat, 937
SetPixmap
gdcm::FileDecompressLookupTable, 564
gdcm::IconImageGenerator, 627
gdcm::PixmapWriter, 957
SetPlanarConfiguration
gdcm::Bitmap, 260
gdcm::ImageChangePlanarConfiguration, 649
gdcm::ImageCodec, 665
SetPMSRescaleInterceptSlope
gdcm::ImageHelper, 678
SetPointCoordinatesData
gdcm::Surface, 1208
SetPointPositionAccuracy
gdcm::Surface, 1209
SetPointsBoundingBoxCoordinates
gdcm::Surface, 1209
SetPort
gdcm::ServiceClassUser, 1120
SetPortSCP
gdcm::ServiceClassUser, 1120
SetPosition
vtkImageColorViewer, 1491
SetPreamble
gdcm::FileMetaInformation, 579
SetPrefix
gdcm::FilenameGenerator, 591
SetPresentationContextID
gdcm::network::PresentationContextAC, 972
gdcm::network::PresentationContextRQ, 978
gdcm::network::PresentationDataValue, 982
gdcm::PresentationContext, 970
SetPresentationContexts

gdcm::network::ULConnection, 1349, 1350
gdcm::ServiceClassUser, 1121
SetPrettyPrint
gdcm::JSON, 768
SetPrimitiveData
gdcm::MeshPrimitive, 815
SetPrimitivesData
gdcm::MeshPrimitive, 816
SetPrimitiveType
gdcm::MeshPrimitive, 816
SetPrivateCreator
gdcm::Tag, 1253
SetProcessingAlgorithm
gdcm::Surface, 1209
SetProgress
gdcm::ProgressEvent, 998
SetPropertyCategory
gdcm::Segment, 1074
SetPropertyType
gdcm::Segment, 1074
SetPropertyTypeModifiers
gdcm::Segment, 1074
SetPurposeOfReferenceCodeSequenceCodeValue
gdcm::FileDerivation, 568
SetQuality
gdcm::JPEG2000Codec, 743
gdcm::JPEGCodec, 758
SetRate
gdcm::JPEG2000Codec, 744
SetReason
gdcm::network::AAbortPDU, 115
gdcm::network::PresentationContextAC, 972
SetRecommendedDisplayCIELabValue
gdcm::Surface, 1209
SetRecommendedDisplayGrayscaleValue
gdcm::Surface, 1209
SetRecommendedPresentationOpacity
gdcm::Surface, 1209
SetRecommendedPresentationType
gdcm::Surface, 1210
SetRecomputeItemLength
gdcm::FileExplicitFilter, 570
SetRecomputeSequenceLength
gdcm::FileExplicitFilter, 570
SetRedLUT
gdcm::LookupTable, 782
SetRef
gdcm::IODEntry, 710
SetRegion
gdcm::ImageRegionReader, 689
SetRenderer
vtkImageColorViewer, 1492
SetRenderWindow
vtkImageColorViewer, 1492
SetRescaleInterceptSlopeValue
gdcm::ImageHelper, 678
SetRetired
gdcm::DictEntry, 431
SetReversible
gdcm::JPEG2000Codec, 744
SetRGB8
gdcm::ImageApplyLookupTable, 640
SetRoot
gdcm::UIDGenerator, 1283
SetRootDirectory
gdcm::DICOMDIRGenerator, 420
SetRows
gdcm::Bitmap, 260
gdcm::Overlay, 897
SetRTStructSetProperties
vtkGDCMPolyDataWriter, 1467
SetSamplesPerPixel
gdcm::PixelFormat, 937
SetScalarType
gdcm::PixelFormat, 937
SetSearchParameter
gdcm::BaseQuery, 235
SetSecondaryCaptureImagePlaneModule
gdcm::ImageHelper, 678
SetSegmentAlgorithmName
gdcm::Segment, 1074
SetSegmentAlgorithmType
gdcm::Segment, 1074
SetSegmentDescription
gdcm::Segment, 1074
SetSegmentLabel
gdcm::Segment, 1075
SetSegmentNumber
gdcm::Segment, 1075
SetSegments
gdcm::SegmentWriter, 1089
SetSize
vtkImageColorViewer, 1492
SetSlice
vtkImageColorViewer, 1492
SetSliceOrientation
vtkImageColorViewer, 1492
SetSliceOrientationToXY
vtkImageColorViewer, 1493
SetSliceOrientationToXZ
vtkImageColorViewer, 1493
SetSliceOrientationToYZ
vtkImageColorViewer, 1493
SetSlope
gdcm::Image, 636
gdcm::Rescaler, 1038
SetSOPClassUIDMode
gdcm::EmptyMaskGenerator, 523

SetSOPInstanceUID
gdcm::BaseQuery, 236
SetSortFunction
gdcm::Sorter, 1144
SetSource
gdcm::network::AAbortPDU, 115
SetSourceApplicationEntityTitle
gdcm::FileMetaInformation, 579
SetSpacing
gdcm::Image, 636
SetSpacingValue
gdcm::ImageHelper, 678
SetState
gdcm::network::ULConnection, 1350
SetStream
gdcm::Reader, 1030
gdcm::StreamImageReader, 1159
gdcm::StreamImageWriter, 1163
gdcm::Trace, 1267
gdcm::Writer, 1537
SetStreamToFile
gdcm::Trace, 1267
SetStyle
gdcm::Printer, 986
gdcm::XMLPrinter, 1544
SetSurfaceComments
gdcm::Surface, 1210
SetSurfaceCount
gdcm::Segment, 1075
SetSurfaceNumber
gdcm::Surface, 1210
SetSurfaceProcessing
gdcm::Surface, 1210
SetSurfaceProcessingDescription
gdcm::Surface, 1210
SetSurfaceProcessingRatio
gdcm::Surface, 1210
SetSyngoDT
gdcm::CSAElement, 346
SetTag
gdcm::AnonymizeEvent, 134
gdcm::DataElement, 380
SetTagsToRead
gdcm::Sorter, 1144
SetTargetPixelType
gdcm::Rescaler, 1038
SetTemplateFileName
gdcm::FileStreamer, 597
SetTileSize
gdcm::JPEG2000Codec, 744
SetTimeout
gdcm::network::ARTIMTimer, 155
gdcm::ServiceClassUser, 1121
SetToUndefined
gdcm::VL, 1397
SetTransferSyntax
gdcm::Bitmap, 260
gdcm::FileChangeTransferSyntax, 561
gdcm::ImageChangeTransferSyntax, 653
gdcm::network::PresentationContextAC, 972
SetTuple
gdcm::network::RoleSelectionSub, 1047
gdcm::network::ServiceClassApplicationInformation, 1113
gdcm::network::SOPClassExtendedNegociationSub, 1138
SetType
gdcm::ModuleEntry, 831
gdcm::Overlay, 897
SetTypeOfData
gdcm::Curve, 369
SetupInteractor
vtkImageColorViewer, 1493
SetUsage
gdcm::IODEntry, 710
SetUserCodec
gdcm::ImageChangeTransferSyntax, 653
SetUserData
gdcm::Parser, 903
SetUserInformation
gdcm::network::AAssociateRQPDU, 126
SetUseSeriesDetails
gdcm::SerieHelper, 1111
SetUseTargetPixelType
gdcm::Rescaler, 1039
SetUseVRUN
gdcm::FileExplicitFilter, 570
SetValue
gdcm::Attribute< Group, Element, TVR, TVM >, 166
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 175
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 194
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 201
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 206
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 213
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 219
gdcm::CSAElement, 347
gdcm::DataElement, 380
gdcm::Element< TVR, TVM >, 462

gdcm::Element< TVR, VM::VM1_2 >, 468
gdcm::Element< TVR, VM::VM1_n >, 473
gdcm::Element< TVR, VM::VM2_2n >, 480
gdcm::Element< TVR, VM::VM2_n >, 486
gdcm::Element< TVR, VM::VM3_3n >, 492
gdcm::Element< TVR, VM::VM3_4 >, 498
gdcm::Element< TVR, VM::VM3_n >, 504
gdcm::Element< VR::AS, VM::VM5 >, 507
gdcm::Element< VR::OB, VM::VM1 >, 512
gdcm::Element< VR::OW, VM::VM1 >, 517
gdcm::PDBElement, 910
SetValueFieldLength
gdcm::DataElement, 381
SetValues
gdcm::Attribute< Group, Element, TVR, TVM >, 167
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 175
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 181
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 186
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 194
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >, 201
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, 207
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >, 213
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, 219
SetVectorAccuracy
gdcm::Surface, 1210
SetVectorCoordinateData
gdcm::Surface, 1210
SetVectorDimensionality
gdcm::Surface, 1211
SetVL
gdcm::DataElement, 381
SetVLToUndefined
gdcm::DataElement, 381
SetVM
gdcm::CSAElement, 347
gdcm::CSAHeaderDictEntry, 360
gdcm::DictEntry, 431
SetVR
gdcm::CSAElement, 347
gdcm::CSAHeaderDictEntry, 360
gdcm::DataElement, 381
gdcm::DictEntry, 431
SetWarning
gdcm::Trace, 1267
SetWarningStream
gdcm::Trace, 1268
SetWindowId
vtkImageColorViewer, 1493
SetWriteDataSetOnly
gdcm::Writer, 1538
SetZSpacingTolerance
gdcm::IPPSorter, 718
SH
gdcm::VR, 1406
SHA1
gdcm::SHA1, 1123
SHComp
gdcm, 88
Shift
vtkGDCMImageReader, 1433
vtkGDCMImageReader2, 1447
ShiftEnd
gdcm::ByteBuffer, 272
ShowAbort
gdcm::SimpleSubjectWatcher, 1130
ShowAnonymization
gdcm::SimpleSubjectWatcher, 1131
ShowData
gdcm::SimpleSubjectWatcher, 1131
ShowDataSet
gdcm::SimpleSubjectWatcher, 1131
ShowFileName
gdcm::SimpleSubjectWatcher, 1131
ShowIteration
gdcm::SimpleSubjectWatcher, 1131
ShowProgress
gdcm::SimpleSubjectWatcher, 1131
SIEMENS
gdcm::Dicts, 437
gdcm::EquipmentManufacturer, 532
SimpleMemberCommand
gdcm::SimpleMemberCommand< T >, 1127
SimpleSubjectWatcher
gdcm::SimpleSubjectWatcher, 1130
SINGLEBIT
gdcm::PixelFormat, 933
SingleSerieUIDFileSetHT
gdcm::SerieHelper, 1111
SingleSerieUIDFileSetmap
gdcm::SerieHelper, 1108
Size
gdcm::CodeString, 315
gdcm::DataSet, 399
gdcm::GroupDict, 620
gdcm::network::AAbortPDU, 115
gdcm::network::AAssociateACPDU, 119
gdcm::network::AAssociateRJPDU, 121
gdcm::network::AAssociateRQPDU, 127
gdcm::network::AbstractSyntax, 130
gdcm::network::ApplicationContext, 146

gdcm::network::AReleaseRPPDU, 151
gdcm::network::AReleaseRQPDU, 153
gdcm::network::AsynchronousOperationsWindowSub, 158
gdcm::network::BasePDU, 231
gdcm::network::ImplementationClassUIDSub, 697
gdcm::network::ImplementationVersionNameSub, 699
gdcm::network::MaximumLengthSub, 791
gdcm::network::PDataTFPDU, 907
gdcm::network::PresentationContextAC, 972
gdcm::network::PresentationContextRQ, 979
gdcm::network::PresentationDataValue, 982
gdcm::network::RoleSelectionSub, 1047
gdcm::network::ServiceClassApplicationInformation, 1113
gdcm::network::SOPClassExtendedNegociationSub, 1139
gdcm::network::TransferSyntaxSub, 1276
gdcm::network::UserInformation, 1383
size_type
gdcm::CodeString, 314
gdcm::LO, 774
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1188
SizeType
gdcm::DataSet, 391
gdcm::FilenameGenerator, 589
gdcm::IOD, 706
gdcm::NestedModuleEntries, 856
gdcm::network::AAssociateACPDU, 117
gdcm::network::AAssociateRQPDU, 124
gdcm::network::PDataTFPDU, 906
gdcm::network::PresentationContextRQ, 976
gdcm::PresentationContext, 968
gdcm::PresentationContextGenerator, 974
gdcm::SequenceOfFragments, 1092
gdcm::SequenceOfItems, 1101
SL
gdcm::VR, 1406
Slice
vtkImageColorViewer, 1496
SLICE_ORIENTATION_XY
vtkImageColorViewer, 1488
SLICE_ORIENTATION_XZ
vtkImageColorViewer, 1488
SLICE_ORIENTATION_YZ
vtkImageColorViewer, 1488
SliceOrientation
vtkImageColorViewer, 1497
Slices
gdcm::MrProtocol::SliceArray, 1134
SmartPointer
gdcm::Object, 873
gdcm::SmartPointer< ObjectType >, 1136
SOPClassExtendedNegociationSub
gdcm::network::SOPClassExtendedNegociationSub, 1138
SOPClassUIDMode
gdcm::EmptyMaskGenerator, 522
SOPInstanceUID
vtkRTStructSetProperties, 1527
Sort
gdcm::IPPSorter, 718
gdcm::Sorter, 1144
Sorter
gdcm::Sorter, 1143
SortFunc
gdcm::Sorter, 1145
SortFunction
gdcm::Sorter, 1143
Source Directory Reference, 67
SpacialFiducialsStorage
gdcm::MediaStorage, 799
SpacialRegistrationStorage
gdcm::MediaStorage, 799
Spacing
gdcm::Spacing, 1147
SpacingType
gdcm::Spacing, 1147
Spectroscopy
gdcm::Spectroscopy, 1148
Split
gdcm::ImageFragmentSplitter, 672
gdcm::SplitMosaicFilter, 1151
SplitExtent
vtkGDCMThreadedImageReader2, 1480
SplitMosaicFilter
gdcm::SplitMosaicFilter, 1150
SQ
gdcm::VR, 1406
Squeeze
gdcm::ApplicationEntity, 148
SS
gdcm::VR, 1406
ST
gdcm::VR, 1406
StableSort
gdcm::Sorter, 1144
StandaloneCurveStorage
gdcm::MediaStorage, 798
StandaloneModalityLUTStorage
gdcm::MediaStorage, 798
StandaloneOverlayStorage
gdcm::MediaStorage, 798
StandaloneVOILUTStorage
gdcm::MediaStorage, 798
Start

gdcm::network::ARTIMTimer, 155
StartAssociation
gdcm::ServiceClassUser, 1121
StartDataElement
gdcm::FileStreamer, 598
StartElement
gdcm::TableReader, 1242
gdcm::XMLDictReader, 1541
gdcm::XMLPrivateDictReader, 1548
StartElementHandler
gdcm::Parser, 901
StartEncode
gdcm::ImageCodec, 665
gdcm::JPEG2000Codec, 744
gdcm::JPEGCodec, 758
gdcm::JPEGLSCodec, 765
gdcm::RLECodec, 1045
StartFilter
gdcm::SimpleSubjectWatcher, 1131
StartGroupDataElement
gdcm::FileStreamer, 598
STATES
gdcm::Surface, 1201
STATES_END
gdcm::Surface, 1202
STComp
gdcm, 88
Stop
gdcm::network::ARTIMTimer, 155
StopAssociation
gdcm::ServiceClassUser, 1121
StopDataElement
gdcm::FileStreamer, 598
StopEncode
gdcm::ImageCodec, 665
gdcm::JPEG2000Codec, 744
gdcm::JPEGCodec, 758
gdcm::JPEGLSCodec, 766
gdcm::RLECodec, 1045
StopGroupDataElement
gdcm::FileStreamer, 598
StopProtocol
gdcm::network::ULConnection, 1350
StrCaseCmp
gdcm::System, 1234
Stream
gdcm::Writer, 1538
StreamImageReader
gdcm::Reader, 1030
gdcm::StreamImageReader, 1156
StreamImageWriter
gdcm::StreamImageWriter, 1161
gdcm::Writer, 1538
StrictScanner
gdcm::StrictScanner, 1170
StrictScanner2
gdcm::StrictScanner2, 1180
String
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1189
StringFilter
gdcm::StringFilter, 1191
StrNCaseCmp
gdcm::System, 1234
StrSep
gdcm::System, 1235
StrTokR
gdcm::System, 1235
StructureSetDate
vtkRTStructSetProperties, 1527
StructureSetLabel
vtkRTStructSetProperties, 1527
StructureSetName
vtkRTStructSetProperties, 1528
StructureSetTime
vtkRTStructSetProperties, 1528
Study
gdcm::Study, 1194
StudyComponentManagementSOPClass
gdcm::MediaStorage, 799
StudyInstanceUID
vtkRTStructSetProperties, 1528
Subject
gdcm::Subject, 1196
Superclass
gdcm::AnonymizeEvent, 133
gdcm::DataEvent, 386
gdcm::DataSetEvent, 402
gdcm::FileNameEvent, 586
gdcm::LO, 774
gdcm::ProgressEvent, 997
SURFACE
gdcm::Surface, 1202
Surface
gdcm::Surface, 1202
SurfaceCount
gdcm::Segment, 1076
SurfaceReader
gdcm::SurfaceReader, 1218
Surfaces
gdcm::Segment, 1076
SurfaceSegmentationStorage
gdcm::MediaStorage, 800
SurfaceVector
gdcm::Segment, 1070
SurfaceWriter
gdcm::SurfaceWriter, 1223
SV

gdcm::VR, 1406
SV10
gdcm::CSAHeader, 351
Swap
gdcm::ByteSwap< T >, 273
gdcm::SwapperDoOp, 1227
gdcm::SwapperNoOp, 1228
SwapArray
gdcm::SwapperDoOp, 1227
gdcm::SwapperNoOp, 1228
SwapCode
gdcm::SwapCode, 1226
SwapCodeType
gdcm::SwapCode, 1225
SwapFromSwapCodeIntoSystem
gdcm::ByteSwap< T >, 273
SwapRange
gdcm::ByteSwap< T >, 274
SwapRangeFromSwapCodeIntoSystem
gdcm::ByteSwap< T >, 274
SyngoDTField
gdcm::CSAElement, 348
SyntaxError
gdcm::Parser, 902
SystemIsBigEndian
gdcm::ByteSwap< T >, 274
SystemIsLittleEndian
gdcm::ByteSwap< T >, 274

T1
gdcm::Type, 1279
T1C
gdcm::Type, 1279
T2
gdcm::Type, 1279
T2C
gdcm::Type, 1280
T3
gdcm::Type, 1280
Table
gdcm::Table, 1237
Table16
vtkLookupTable16, 1517
TableEntry
gdcm::TableEntry, 1239
TableInternal
gdcm::Table, 1238
TableReader
gdcm::TableReader, 1240
TableRow
gdcm::network::TableRow, 1244
Tag
gdcm::Tag, 1247
tag
gdcm::Tag, 1254
TagField
gdcm::DataElement, 383
TagMismatchError
gdcm::Parser, 902
TagPath
gdcm::TagPath, 1255
tags
gdcm::Tag, 1254
TagsToRead
gdcm::Sorter, 1145
TagToValue
gdcm::Scanner, 1051
gdcm::StrictScanner, 1169
TagToValueValueType
gdcm::Scanner, 1051
gdcm::StrictScanner, 1170
TConstMemberFunctionPointer
gdcm::MemberCommand< T >, 808
TestAbortOff
gdcm::SimpleSubjectWatcher, 1132
TestAbortOn
gdcm::SimpleSubjectWatcher, 1132
Testing
gdcm::Testing, 1258
TestPBKDF2
gdcm::ASN1, 157
TestsList.txt, 1549
ThreadedExecute
vtkImageRGBToYBR, 1511
vtkImageYBRToRGB, 1513
ThreadedRequestData
vtkGDCMThreadedImageReader2, 1480
vtkImageMapToColors16, 1500
vtkImageMapToWindowLevelColors2, 1505
TM
gdcm::VR, 1406
TMComp
gdcm, 88
TMemberFunctionPointer
gdcm::MemberCommand< T >, 808
gdcm::SimpleMemberCommand< T >, 1127
Todo List, 3
ToPyObject
gdcm::PythonFilter, 1005
TOSHIBA
gdcm::EquipmentManufacturer, 532
ToshibaPrivateDataStorage
gdcm::MediaStorage, 799
ToString
gdcm::StringFilter, 1193
ToStringPair
gdcm::StringFilter, 1193, 1194
ToUnixSlashes

gdcm::Filename, 583
ToWindowsSlashes
gdcm::Filename, 583
Trace
gdcm::Trace, 1265
TransferSyntax
gdcm::TransferSyntax, 1272
TransferSyntaxArrayType
gdcm::PresentationContext, 968
TransferSyntaxes
gdcm::PresentationContext, 970
TransferSyntaxStringsType
gdcm::UIDs, 1300
TransferSyntaxSub
gdcm::network::TransferSyntaxSub, 1275
Transition
gdcm::network::Transition, 1277
transitions
gdcm::network::TableRow, 1244
TRIANGLE
gdcm::MeshPrimitive, 814
TRIANGLE_FAN
gdcm::MeshPrimitive, 814
TRIANGLE_STRIP
gdcm::MeshPrimitive, 814
Trim
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1190
TrimInternal
gdcm::CodeString, 315
Truncate
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1190
TryJPEG2000Codec
gdcm::Bitmap, 261
gdcm::ImageChangeTransferSyntax, 654
TryJPEG2000Codec2
gdcm::Bitmap, 261
TryJPEGCodec
gdcm::Bitmap, 261
gdcm::ImageChangeTransferSyntax, 654
TryJPEGCodec2
gdcm::Bitmap, 261
TryJPEGLSCodec
gdcm::Bitmap, 261
gdcm::ImageChangeTransferSyntax, 654
TryKAKADUCodec
gdcm::Bitmap, 261
TryPVRGCodec
gdcm::Bitmap, 262
TryRAWCodec
gdcm::Bitmap, 262
gdcm::ImageChangeTransferSyntax, 654
TryRLECodec
gdcm::Bitmap, 262
gdcm::ImageChangeTransferSyntax, 654
TS
gdcm::Bitmap, 264
TS_END
gdcm::TransferSyntax, 1272
TSName
gdcm::UIDs, 1300
TSType
gdcm::TransferSyntax, 1271
gdcm::UIDs, 1300
Type
gdcm::Element< TVR, TVM >, 459
gdcm::Element< TVR, VM::VM1_2 >, 466
gdcm::Element< TVR, VM::VM1_n >, 470
gdcm::Element< TVR, VM::VM2_2n >, 478
gdcm::Element< TVR, VM::VM2_n >, 484
gdcm::Element< TVR, VM::VM3_3n >, 490
gdcm::Element< TVR, VM::VM3_4 >, 496
gdcm::Element< TVR, VM::VM3_n >, 502
gdcm::Element< VR::AS, VM::VM5 >, 506
gdcm::Element< VR::OB, VM::VM1 >, 511
gdcm::Element< VR::OW, VM::VM1 >, 516
gdcm::EquipmentManufacturer, 532
gdcm::Type, 1280
gdcm::VL, 1395
TYPETOENCODING
gdcmVR.h, 1800
TYPETOLENGTH
gdcmVM.h, 1796
TypeToString
gdcm::EquipmentManufacturer, 532
TypeType
gdcm::Type, 1279

UC
gdcm::VR, 1406
UCComp
gdcm, 88
UI
gdcm::VR, 1406
UIComp
gdcm, 88
UIDGenerator
gdcm::UIDGenerator, 1282
UIH
gdcm::EquipmentManufacturer, 532
UINT12
gdcm::PixelFormat, 932
UINT16
gdcm::PixelFormat, 932
UINT32
gdcm::PixelFormat, 932
UINT64

gdcm::PixelFormat, 932
UINT8
gdcm::PixelFormat, 932
UL
gdcm::VR, 1406
ULAction
gdcm::network::ULAction, 1304
ULActionAE6
gdcm::network::ULConnection, 1350
ULBasicCallback
gdcm::network::ULBasicCallback, 1345
ULConnection
gdcm::network::ULConnection, 1347
ULConnectionCallback
gdcm::network::ULConnectionCallback, 1352
ULConnectionInfo
gdcm::network::ULConnectionInfo, 1354
ULConnectionManager
gdcm::network::ULConnection, 1350
gdcm::network::ULConnectionManager, 1358
ULEvent
gdcm::network::ULEvent, 1363
ULTransitionTable
gdcm::network::ULTransitionTable, 1365
UltrasoundImageStorage
gdcm::MediaStorage, 798
UltrasoundImageStorageRetired
gdcm::MediaStorage, 798
UltrasoundMultiFrameImageStorage
gdcm::MediaStorage, 798
UltrasoundMultiFrameImageStorageRetired
gdcm::MediaStorage, 798
ULWritingCallback
gdcm::network::ULWritingCallback, 1367
UN
gdcm::VR, 1406
UndefinedEntityError
gdcm::Parser, 902
underline
gdcm::terminal, 110
UnexpectedStateError
gdcm::Parser, 902
UnInstallPipeline
vtkImageColorViewer, 1493
UNKNOWN
gdcm::CSAHeader, 351
gdcm::EquipmentManufacturer, 532
gdcm::LookupTable, 778
gdcm::Orientation, 886
gdcm::PhotometricInterpretation, 927
gdcm::PixelFormat, 933
gdcm::Spacing, 1147
gdcm::Surface, 1202
gdcm::Type, 1280
Unknown
gdcm::SwapCode, 1226
gdcm::TransferSyntax, 1270
Unpack
gdcm::Unpacker12Bits, 1376
UnRegister
gdcm::Object, 873
UnusedBitsPresentInPixelData
gdcm::Bitmap, 262
gdcm::Pixmap, 945
Update
gdcm::Curve, 369
gdcm::Overlay, 897
UpdateDisplayExtent
vtkImageColorViewer, 1493
UpdateOrientation
vtkImageColorViewer, 1494
UpdatePosition
gdcm::ByteBuffer, 273
UR
gdcm::VR, 1406
URComp
gdcm, 88
URI
gdcm::MediaStorage, 801
US
gdcm::VR, 1406
US_OW
gdcm::VR, 1406
US_SS
gdcm::VR, 1406
US_SS_OW
gdcm::VR, 1406
Usage
gdcm::Usage, 1378
UsageType
gdcm::Usage, 1378
UseDictAlways
gdcm::PythonFilter, 1005
gdcm::StringFilter, 1194
UseGrayscaleSecondaryImageStorage
gdcm::EmptyMaskGenerator, 522
UseOriginalSOPClassUID
gdcm::EmptyMaskGenerator, 522
UserInformation
gdcm::network::UserInformation, 1382
UserOption
gdcm::Usage, 1378
UserOrdering
gdcm::SerieHelper, 1111
UT
gdcm::VR, 1406
UTComp
gdcm, 88

Utilities Directory Reference, 68
UV
gdcm::VR, 1406

V
gdcm::Validate, 1386
Valid
gdcm::Preamble, 966
Validate
gdcm::PixelFormat, 938
gdcm::Validate, 1385
ValidateQuery
gdcm::BaseQuery, 236
gdcm::BaseRootQuery, 241
gdcm::FindPatientRootQuery, 605
gdcm::FindStudyRootQuery, 609
gdcm::ModalityPerformedProcedureStepCreateQuery, 819
gdcm::ModalityPerformedProcedureStepSetQuery, 823
gdcm::MovePatientRootQuery, 838
gdcm::MoveStudyRootQuery, 842
gdcm::WLMFindQuery, 1532
Validation
gdcm::Validate, 1386
ValidDataSet
gdcm::BaseQuery, 236
Value
gdcm::Value, 1388
value
gdcm::STATIC_ASSERTION_FAILURE< true >, 1155
value_type
gdcm::CodeString, 314
gdcm::LO, 775
gdcm::String< TDelimiter, TMaxLength, TPadChar >, 1188
ValueField
gdcm::DataElement, 383
gdcm::PDBElement, 910
ValueLengthField
gdcm::DataElement, 383
ValueMultiplicityField
gdcm::CSAElement, 348
ValuePtr
gdcm::DataElement, 373
ValuesType
gdcm::Scanner, 1052
gdcm::Scanner2, 1061
gdcm::StrictScanner, 1170
gdcm::StrictScanner2, 1179
VERBOSE_STYLE
gdcm::Printer, 985
Verify
gdcm::Defs, 411
gdcm::Macro, 788
gdcm::Module, 827
Version
gdcm::Version, 1392
VERTEX
gdcm::MeshPrimitive, 814
Video
gdcm::MediaStorage, 801
VideoEndoscopicImageStorage
gdcm::MediaStorage, 799
VideoMicroscopicImageStorage
gdcm::MediaStorage, 800
VideoPhotographicImageStorage
gdcm::MediaStorage, 800
VIEWType
gdcm::Surface, 1202
VIEWType_END
gdcm::Surface, 1202
VL
gdcm::VL, 1395
VL16
gdcm::VR, 1406
VL32
gdcm::VR, 1406
VLEndoscopicImageStorage
gdcm::MediaStorage, 800
VLMicroscopicImageStorage
gdcm::MediaStorage, 800
VLPhotographicImageStorage
gdcm::MediaStorage, 799
VLWholeSlideMicroscopyImageStorage
gdcm::MediaStorage, 800
VM
gdcm::VM, 1401
VM0
gdcm::VM, 1400
VM1
gdcm::VM, 1400
VM10
gdcm::VM, 1400
VM12
gdcm::VM, 1400
VM16
gdcm::VM, 1400
VM18
gdcm::VM, 1400
VM1_2
gdcm::VM, 1400
VM1_3
gdcm::VM, 1400
VM1_32
gdcm::VM, 1400
VM1_4

gdcm::VM, 1400
VM1_5
gdcm::VM, 1400
VM1_8
gdcm::VM, 1400
VM1_99
gdcm::VM, 1400
VM1_n
gdcm::VM, 1400
VM2
gdcm::VM, 1400
VM24
gdcm::VM, 1400
VM256
gdcm::VM, 1400
VM28
gdcm::VM, 1400
VM2_2n
gdcm::VM, 1400
VM2_n
gdcm::VM, 1401
VM3
gdcm::VM, 1400
VM30_30n
gdcm::VM, 1401
VM32
gdcm::VM, 1400
VM35
gdcm::VM, 1400
VM3_3n
gdcm::VM, 1401
VM3_4
gdcm::VM, 1401
VM3_n
gdcm::VM, 1401
VM4
gdcm::VM, 1400
VM47_47n
gdcm::VM, 1401
VM4_4n
gdcm::VM, 1401
VM5
gdcm::VM, 1400
VM6
gdcm::VM, 1400
VM6_6n
gdcm::VM, 1401
VM6_n
gdcm::VM, 1401
VM7_7n
gdcm::VM, 1401
VM8
gdcm::VM, 1400
VM9
gdcm::VM, 1400
VM99
gdcm::VM, 1400
VM_END
gdcm::VM, 1401
VMType
gdcm::Attribute< Group, Element, TVR, TVM >, 161
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 171
gdcm::VM, 1400
VR
gdcm::VR, 1407
VR_END
gdcm::VR, 1407
VR_VM1
gdcm::VR, 1407
VRALL
gdcm::VR, 1407
VRASCII
gdcm::VR, 1406
VRBINARY
gdcm::VR, 1406
VRField
gdcm::CSAElement, 348
gdcm::DataElement, 383
VRType
gdcm::VR, 1405
VRTypeTemplateCase
gdcmVR.h, 1800
VT100
gdcm::terminal, 110
VTK Directory Reference, 68
VTK_CMYK
vtkGDCMImageReader.h, 2188
vtkGDCMImageReader2.h, 2193
VTK_INVERSE_LUMINANCE
vtkGDCMImageReader.h, 2188
vtkGDCMImageReader2.h, 2193
VTK_LOOKUP_TABLE
vtkGDCMImageReader.h, 2188
vtkGDCMImageReader2.h, 2193
VTK_YBR
vtkGDCMImageReader.h, 2188
vtkGDCMImageReader2.h, 2193
vtkBooleanMacro
vtkGDCMImageReader, 1424, 1425
vtkGDCMImageReader2, 1439, 1440
vtkGDCMImageWriter, 1452
vtkGDCMThreadedImageReader, 1476
vtkGDCMThreadedImageReader2, 1481
vtkImageColorViewer, 1494
vtkImageMapToColors16, 1501
vtkGDCMImageReader, 1418
~vtkGDCMImageReader, 1421

ApplyInverseVideo, 1430
ApplyLookupTable, 1430
ApplyPlanarConfiguration, 1430
ApplyShiftScale, 1430
ApplyYBRToRGB, 1430
CanReadFile, 1421
Curve, 1430
DirectionCosines, 1430
ExecuteData, 1421
ExecuteInformation, 1422
FileNames, 1431
FillMedicalImageInformation, 1422
ForceRescale, 1431
GetDescriptiveName, 1422
GetFileExtensions, 1422
GetIconImage, 1422
GetOverlay, 1422
IconDataScalarType, 1431
IconImageDataExtent, 1431
IconNumberOfScalarComponents, 1431
ImageFormat, 1431
ImageOrientationPatient, 1431
ImagePositionPatient, 1431
LoadIconImage, 1432
LoadOverlays, 1432
LoadSingleFile, 1422
LossyFlag, 1432
MedicalImageProperties, 1432
New, 1422
NumberOfIconImages, 1432
NumberOfOverlays, 1432
PlanarConfiguration, 1432
PrintSelf, 1423
RequestDataCompat, 1423
RequestInformationCompat, 1423
Scale, 1433
SetCurve, 1423
SetFileNames, 1423
SetFilePattern, 1424
SetFilePrefix, 1424
SetMedicalImageProperties, 1424
Shift, 1433
vtkBooleanMacro, 1424, 1425
vtkGDCMImageReader, 1421
vtkGDCMMedicalImageProperties, 1459
vtkGetMacro, 1425–1427
vtkGetObjectMacro, 1427, 1428
vtkGetStringMacro, 1428
vtkGetVector3Macro, 1428
vtkGetVector6Macro, 1428
vtkSetMacro, 1428, 1429
vtkSetVector6Macro, 1429
vtkTypeMacro, 1429
vtkGDCMImageReader.h, 2186, 2188
VTK_CMYK, 2188
VTK_INVERSE_LUMINANCE, 2188
VTK_LOOKUP_TABLE, 2188
VTK_YBR, 2188
vtkGDCMImageReader2, 1433
~vtkGDCMImageReader2, 1436
ApplyInverseVideo, 1445
ApplyLookupTable, 1445
ApplyPlanarConfiguration, 1445
ApplyShiftScale, 1445
ApplyYBRToRGB, 1445
CanReadFile, 1436
Curve, 1445
DirectionCosines, 1445
FillMedicalImageInformation, 1436
ForceRescale, 1446
GetDescriptiveName, 1436
GetFileExtensions, 1436
GetIconImage, 1437
GetIconImagePort, 1437
GetOverlay, 1437
GetOverlayPort, 1437
IconDataScalarType, 1446
IconImageDataExtent, 1446
IconNumberOfScalarComponents, 1446
ImageFormat, 1446
ImageOrientationPatient, 1446
ImagePositionPatient, 1446
LoadIconImage, 1446
LoadOverlays, 1447
LoadSingleFile, 1437
LossyFlag, 1447
New, 1437
NumberOfIconImages, 1447
NumberOfOverlays, 1447
PlanarConfiguration, 1447
PrintSelf, 1437
ProcessRequest, 1438
RequestData, 1438
RequestDataCompat, 1438
RequestInformation, 1438
RequestInformationCompat, 1438
Scale, 1447
SetCurve, 1439
SetFilePattern, 1439
SetFilePrefix, 1439
SetMedicalImageProperties, 1439
Shift, 1447
vtkBooleanMacro, 1439, 1440
vtkGDCMImageReader2, 1436
vtkGDCMMedicalImageProperties, 1459
vtkGetMacro, 1440–1442
vtkGetObjectMacro, 1442
vtkGetStringMacro, 1443

vtkGetVector3Macro, 1443
vtkGetVector6Macro, 1443
vtkSetMacro, 1443, 1444
vtkSetVector6Macro, 1444
vtkTypeMacro, 1444
vtkGDCMImageReader2.h, 2192, 2194
VTK_CMYK, 2193
VTK_INVERSE_LUMINANCE, 2193
VTK_LOOKUP_TABLE, 2193
VTK_YBR, 2193
vtkGDCMImageWriter, 1448
~vtkGDCMImageWriter, 1450
CompressionTypes, 1450
GetDescriptiveName, 1451
GetFileExtensions, 1451
GetFileName, 1451
JPEG2000_COMPRESSION, 1450
JPEG_COMPRESSION, 1450
JPEGLS_COMPRESSION, 1450
New, 1451
NO_COMPRESSION, 1450
PrintSelf, 1451
RLE_COMPRESSION, 1450
SetDirectionCosines, 1451
SetDirectionCosinesFromImageOrientationPatient, 1451
SetFileNames, 1452
SetMedicalImageProperties, 1452
vtkBooleanMacro, 1452
vtkGDCMImageWriter, 1450
vtkGDCMMedicalImageProperties, 1459
vtkGetMacro, 1452, 1453
vtkGetObjectMacro, 1453, 1454
vtkGetStringMacro, 1454
vtkSetMacro, 1454, 1455
vtkSetStringMacro, 1455
vtkTypeMacro, 1455
Write, 1455
WriteGDCMData, 1456
WriteSlice, 1456
vtkGDCMImageWriter.h, 2197, 2198
vtkGDCMMedicalImageProperties, 1456
~vtkGDCMMedicalImageProperties, 1457
Clear, 1458
GetFile, 1458
New, 1458
PrintSelf, 1458
PushBackFile, 1458
vtkGDCMImageReader, 1459
vtkGDCMImageReader2, 1459
vtkGDCMImageWriter, 1459
vtkGDCMMedicalImageProperties, 1457
vtkTypeMacro, 1458
vtkGDCMMedicalImageProperties.h, 2200, 2201
vtkGDCMPolyDataReader, 1459
~vtkGDCMPolyDataReader, 1461
FileName, 1463
FillMedicalImageInformation, 1461
MedicalImageProperties, 1463
New, 1461
PrintSelf, 1461
RequestData, 1462
RequestData_HemodynamicWaveformStorage, 1462
RequestData_RTStructureSetStorage, 1462
RequestInformation, 1462
RequestInformation_HemodynamicWaveformStorage, 1462
RequestInformation_RTStructureSetStorage, 1462
RTStructSetProperties, 1464
vtkGDCMPolyDataReader, 1461
vtkGetObjectMacro, 1462, 1463
vtkGetStringMacro, 1463
vtkSetStringMacro, 1463
vtkTypeMacro, 1463
vtkGDCMPolyDataReader.h, 2206
vtkGDCMPolyDataWriter, 1464
~vtkGDCMPolyDataWriter, 1466
InitializeRTStructSet, 1466
MedicalImageProperties, 1468
New, 1466
PrintSelf, 1466
RTStructSetProperties, 1468
SetMedicalImageProperties, 1467
SetNumberOfInputPorts, 1467
SetRTStructSetProperties, 1467
vtkGDCMPolyDataWriter, 1466
vtkTypeMacro, 1467
WriteData, 1467
WriteRTSTRUCTData, 1468
WriteRTSTRUCTInfo, 1468
vtkGDCMPolyDataWriter.h, 2207, 2208
vtkGDCMTesting, 1469
~vtkGDCMTesting, 1470
GetGDCMDataRoot, 1471
GetMD5MetaImage, 1471
GetMHDMD5FromFile, 1471
GetNumberOfMD5MetaImages, 1471
GetRAWMD5FromFile, 1471
GetVTKDataRoot, 1471
MD5MetaImagesType, 1470
New, 1471
PrintSelf, 1472
vtkGDCMTesting, 1470
vtkTypeMacro, 1472
vtkGDCMTesting.h, 2209, 2210
vtkGDCMThreadedImageReader, 1472
~vtkGDCMThreadedImageReader, 1475

ExecuteData, 1475
ExecuteInformation, 1475
New, 1476
PrintSelf, 1476
ReadFiles, 1476
RequestDataCompat, 1476
vtkBooleanMacro, 1476
vtkGDCMThreadedImageReader, 1475
vtkGetMacro, 1476
vtkSetMacro, 1476, 1477
vtkTypeMacro, 1477
vtkGDCMThreadedImageReader.h, 2211
vtkGDCMThreadedImageReader2, 1478
~vtkGDCMThreadedImageReader2, 1479
GetFileName, 1480
New, 1480
PrintSelf, 1480
RequestInformation, 1480
SetFileName, 1480
SetFileNames, 1480
SplitExtent, 1480
ThreadedRequestData, 1480
vtkBooleanMacro, 1481
vtkGDCMThreadedImageReader2, 1479
vtkGetMacro, 1481, 1482
vtkGetObjectMacro, 1482
vtkGetVector3Macro, 1482, 1483
vtkGetVector6Macro, 1483
vtkSetMacro, 1483, 1484
vtkSetVector3Macro, 1484
vtkSetVector6Macro, 1484
vtkTypeMacro, 1484
vtkGDCMThreadedImageReader2.h, 2213
vtkGetMacro
vtkGDCMImageReader, 1425–1427
vtkGDCMImageReader2, 1440–1442
vtkGDCMImageWriter, 1452, 1453
vtkGDCMThreadedImageReader, 1476
vtkGDCMThreadedImageReader2, 1481, 1482
vtkImageColorViewer, 1494
vtkImageMapToColors16, 1501
vtkImageMapToWindowLevelColors2, 1506
vtkGetObjectMacro
vtkGDCMImageReader, 1427, 1428
vtkGDCMImageReader2, 1442
vtkGDCMImageWriter, 1453, 1454
vtkGDCMPolyDataReader, 1462, 1463
vtkGDCMThreadedImageReader2, 1482
vtkImageColorViewer, 1494, 1495
vtkImageMapToColors16, 1501
vtkGetStringMacro
vtkGDCMImageReader, 1428
vtkGDCMImageReader2, 1443
vtkGDCMImageWriter, 1454
vtkGDCMPolyDataReader, 1463
vtkRTStructSetProperties, 1523–1525
vtkGetVector3Macro
vtkGDCMImageReader, 1428
vtkGDCMImageReader2, 1443
vtkGDCMThreadedImageReader2, 1482, 1483
vtkGetVector6Macro
vtkGDCMImageReader, 1428
vtkGDCMImageReader2, 1443
vtkGDCMThreadedImageReader2, 1483
vtkImageColorViewer, 1485
~vtkImageColorViewer, 1488
AddInput, 1488
AddInputConnection, 1488
FirstRender, 1496
GetColorLevel, 1488
GetColorWindow, 1488
GetInput, 1488
GetOffScreenRendering, 1488
GetOverlayVisibility, 1489
GetPosition, 1489
GetSize, 1489
GetSliceMax, 1489
GetSliceMin, 1489
GetSliceRange, 1489
GetWindowName, 1489
ImageActor, 1496
InstallPipeline, 1490
Interactor, 1496
InteractorStyle, 1496
New, 1490
OverlayImageActor, 1496
PrintSelf, 1490
Render, 1490
Renderer, 1496
RenderWindow, 1496
SetColorLevel, 1490
SetColorWindow, 1490
SetDisplayId, 1490
SetInput, 1491
SetInputConnection, 1491
SetOffScreenRendering, 1491
SetOverlayVisibility, 1491
SetParentId, 1491
SetPosition, 1491
SetRenderer, 1492
SetRenderWindow, 1492
SetSize, 1492
SetSlice, 1492
SetSliceOrientation, 1492
SetSliceOrientationToXY, 1493
SetSliceOrientationToXZ, 1493
SetSliceOrientationToYZ, 1493
SetupInteractor, 1493

SetWindowId, 1493
Slice, 1496
SLICE_ORIENTATION_XY, 1488
SLICE_ORIENTATION_XZ, 1488
SLICE_ORIENTATION_YZ, 1488
SliceOrientation, 1497
UnInstallPipeline, 1493
UpdateDisplayExtent, 1493
UpdateOrientation, 1494
vtkBooleanMacro, 1494
vtkGetMacro, 1494
vtkGetObjectMacro, 1494, 1495
vtkImageColorViewer, 1488
vtkImageColorViewerCallback, 1495
vtkTypeMacro, 1495
WindowLevel, 1497
vtkImageColorViewer.h, 2215, 2216
vtkImageColorViewerCallback
vtkImageColorViewer, 1495
vtkImageMapToColors16, 1497
~vtkImageMapToColors16, 1499
ActiveComponent, 1502
DataWasPassed, 1502
GetMTime, 1499
LookupTable, 1503
New, 1499
OutputFormat, 1503
PassAlphaToOutput, 1503
PrintSelf, 1499
RequestData, 1499
RequestInformation, 1500
SetLookupTable, 1500
SetOutputFormatToLuminance, 1500
SetOutputFormatToLuminanceAlpha, 1500
SetOutputFormatToRGB, 1500
SetOutputFormatToRGBA, 1500
ThreadedRequestData, 1500
vtkBooleanMacro, 1501
vtkGetMacro, 1501
vtkGetObjectMacro, 1501
vtkImageMapToColors16, 1499
vtkSetMacro, 1502
vtkTypeMacro, 1502
vtkImageMapToColors16.h, 2219
vtkImageMapToWindowLevelColors2, 1503
~vtkImageMapToWindowLevelColors2, 1505
Level, 1507
New, 1505
PrintSelf, 1505
RequestData, 1505
RequestInformation, 1505
ThreadedRequestData, 1505
vtkGetMacro, 1506
vtkImageMapToWindowLevelColors2, 1505
vtkSetMacro, 1506
vtkTypeMacro, 1506
Window, 1507
vtkImageMapToWindowLevelColors2.h, 2221
vtkImagePlanarComponentsToComponents, 1507
~vtkImagePlanarComponentsToComponents, 1508
New, 1509
PrintSelf, 1509
RequestData, 1509
vtkImagePlanarComponentsToComponents, 1508
vtkTypeMacro, 1509
vtkImagePlanarComponentsToComponents.h, 2223
vtkImageRGBToYBR, 1510
~vtkImageRGBToYBR, 1511
New, 1511
PrintSelf, 1511
ThreadedExecute, 1511
vtkImageRGBToYBR, 1511
vtkTypeMacro, 1511
vtkImageRGBToYBR.h, 2224, 2225
vtkImageYBRToRGB, 1512
~vtkImageYBRToRGB, 1513
New, 1513
PrintSelf, 1513
ThreadedExecute, 1513
vtkImageYBRToRGB, 1513
vtkTypeMacro, 1514
vtkImageYBRToRGB.h, 2226
vtkLookupTable16, 1514
~vtkLookupTable16, 1515
Build, 1516
GetPointer, 1516
MapScalarsThroughTable2, 1516
New, 1516
PrintSelf, 1516
SetNumberOfTableValues, 1516
Table16, 1517
vtkLookupTable16, 1515
vtkTypeMacro, 1517
WritePointer, 1517
vtkLookupTable16.h, 2227, 2228
vtkRTStructSetProperties, 1518
~vtkRTStructSetProperties, 1520
AddContourReferencedFrameOfReference, 1520
AddReferencedFrameOfReference, 1520
AddStructureSetROI, 1520
AddStructureSetROIObservation, 1521
Clear, 1521
DeepCopy, 1521
GetContourReferencedFrameOfReferenceClassUID, 1521
GetContourReferencedFrameOfReferenceInstanceUID, 1521

GetNumberOfContourReferencedFrameOfRefer-
ences, 1521
GetNumberOfReferencedFrameOfReferences, 1522
GetNumberOfStructureSetROIs, 1522
GetReferencedFrameOfReferenceClassUID, 1522
GetReferencedFrameOfReferenceInstanceUID, 1522
GetStructureSetObservationNumber, 1522
GetStructureSetROIDescription, 1522
GetStructureSetROIGenerationAlgorithm, 1522
GetStructureSetROIName, 1522
GetStructureSetROINumber, 1523
GetStructureSetROIObservationLabel, 1523
GetStructureSetROIRefFrameRefUID, 1523
GetStructureSetRTROIInterpretedType, 1523
Internals, 1527
New, 1523
PrintSelf, 1523
ReferenceFrameOfReferenceUID, 1527
ReferenceSeriesInstanceUID, 1527
SeriesInstanceUID, 1527
SOPInstanceUID, 1527
StructureSetDate, 1527
StructureSetLabel, 1527
StructureSetName, 1528
StructureSetTime, 1528
StudyInstanceUID, 1528
vtkGetStringMacro, 1523–1525
vtkRTStructSetProperties, 1520
vtkSetStringMacro, 1525, 1526
vtkTypeMacro, 1526
vtkRTStructSetProperties.h, 2229, 2230
vtkSetMacro
vtkGDCMImageReader, 1428, 1429
vtkGDCMImageReader2, 1443, 1444
vtkGDCMImageWriter, 1454, 1455
vtkGDCMThreadedImageReader, 1476, 1477
vtkGDCMThreadedImageReader2, 1483, 1484
vtkImageMapToColors16, 1502
vtkImageMapToWindowLevelColors2, 1506
vtkSetStringMacro
vtkGDCMImageWriter, 1455
vtkGDCMPolyDataReader, 1463
vtkRTStructSetProperties, 1525, 1526
vtkSetVector3Macro
vtkGDCMThreadedImageReader2, 1484
vtkSetVector6Macro
vtkGDCMImageReader, 1429
vtkGDCMImageReader2, 1444
vtkGDCMThreadedImageReader2, 1484
vtkTypeMacro
vtkGDCMImageReader, 1429
vtkGDCMImageReader2, 1444
vtkGDCMImageWriter, 1455
vtkGDCMMedicalImageProperties, 1458
vtkGDCMPolyDataReader, 1463
vtkGDCMPolyDataWriter, 1467
vtkGDCMTesting, 1472
vtkGDCMThreadedImageReader, 1477
vtkGDCMThreadedImageReader2, 1484
vtkImageColorViewer, 1495
vtkImageMapToColors16, 1502
vtkImageMapToWindowLevelColors2, 1506
vtkImagePlanarComponentsToComponents, 1509
vtkImageRGBToYBR, 1511
vtkImageYBRToRGB, 1514
vtkLookupTable16, 1517
vtkRTStructSetProperties, 1526

WarningOff
gdcm::Trace, 1268
WarningOn
gdcm::Trace, 1268
Waveform
gdcm::MediaStorage, 801
gdcm::Waveform, 1529
WeirdPapryus
gdcm::TransferSyntax, 1271
what
gdcm::Exception, 537
white
gdcm::terminal, 110
Window
vtkImageMapToWindowLevelColors2, 1507
WindowLevel
vtkImageColorViewer, 1497
WIREFRAME
gdcm::Surface, 1202
WLMFindQuery
gdcm::WLMFindQuery, 1532
Wrapping Directory Reference, 69
Write
gdcm::ByteValue, 284
gdcm::CommandDataSet, 323
gdcm::DataElement, 382
gdcm::DataSet, 400
gdcm::Element< TVR, TVM >, 462
gdcm::Element< TVR, VM::VM1_2 >, 468
gdcm::Element< TVR, VM::VM1_n >, 474
gdcm::Element< TVR, VM::VM2_2n >, 480
gdcm::Element< TVR, VM::VM2_n >, 486
gdcm::Element< TVR, VM::VM3_3n >, 492
gdcm::Element< TVR, VM::VM3_4 >, 498
gdcm::Element< TVR, VM::VM3_n >, 504
gdcm::Element< VR::AS, VM::VM5 >, 508
gdcm::Element< VR::OB, VM::VM1 >, 512
gdcm::Element< VR::OW, VM::VM1 >, 517
gdcm::EncodingImplementation< VR::VRASCII >,
526, 527

gdcm::EncodingImplementation< VR::VRBINARY >, 529
gdcm::ExplicitDataElement, 543
gdcm::File, 552
gdcm::FileAnonymizer, 557
gdcm::FileMetaInformation, 580
gdcm::Fragment, 614
gdcm::ImageWriter, 696
gdcm::ImplicitDataElement, 703
gdcm::Item, 725
gdcm::network::AAbortPDU, 115
gdcm::network::AAssociateACPDU, 119
gdcm::network::AAssociateRJPDU, 121
gdcm::network::AAssociateRQPDU, 127
gdcm::network::AbstractSyntax, 130
gdcm::network::ApplicationContext, 146
gdcm::network::AReleaseRPPDU, 151
gdcm::network::AReleaseRQPDU, 153
gdcm::network::AsynchronousOperationsWindowSub, 158
gdcm::network::BasePDU, 232
gdcm::network::ImplementationClassUIDSub, 697
gdcm::network::ImplementationUIDSub, 698
gdcm::network::ImplementationVersionNameSub, 699
gdcm::network::MaximumLengthSub, 791
gdcm::network::PDataTFPDU, 907
gdcm::network::PresentationContextAC, 972
gdcm::network::PresentationContextRQ, 979
gdcm::network::PresentationDataValue, 982
gdcm::network::RoleSelectionSub, 1047
gdcm::network::ServiceClassApplicationInformation, 1113
gdcm::network::SOPClassExtendedNegociationSub, 1139
gdcm::network::TransferSyntaxSub, 1276
gdcm::network::UserInformation, 1383
gdcm::PGXCodec, 926
gdcm::PixmapWriter, 957
gdcm::PNMCodec, 962
gdcm::Preamble, 966
gdcm::SegmentWriter, 1089
gdcm::SequenceOfFragments, 1096
gdcm::SequenceOfItems, 1105
gdcm::StreamImageWriter, 1163
gdcm::SurfaceWriter, 1224
gdcm::Tag, 1253
gdcm::ValueIO< TDE, TSwap, TType >, 1390
gdcm::VL, 1397
gdcm::VR, 1410
gdcm::VRVLSize< 0 >, 1416
gdcm::VRVLSize< 1 >, 1418
gdcm::Writer, 1538
vtkGDCMImageWriter, 1455
Write16
gdcm::VL, 1397
WriteASCII
gdcm::Element< TVR, VM::VM1_n >, 474
WriteBuffer
gdcm::ByteValue, 284
gdcm::SequenceOfFragments, 1097
WriteBufferAsRGBA
gdcm::LookupTable, 782
WriteData
vtkGDCMPolyDataWriter, 1467
WriteFooter
gdcm::DictConverter, 427
WriteGDCMData
vtkGDCMImageWriter, 1456
WriteHeader
gdcm::DictConverter, 428
WriteHelpFile
gdcm::BaseQuery, 236
WriteImageInformation
gdcm::StreamImageWriter, 1163
WriteImageSubregionRAW
gdcm::StreamImageWriter, 1164
WritePointer
vtkLookupTable16, 1517
WriteQuery
gdcm::BaseQuery, 236
Writer
gdcm::Writer, 1536
WriteRawHeader
gdcm::StreamImageWriter, 1164
WriteRTSTRUCTData
vtkGDCMPolyDataWriter, 1468
WriteRTSTRUCTInfo
vtkGDCMPolyDataWriter, 1468
WriteSlice
vtkGDCMImageWriter, 1456

x16printf
gdcm, 102
XML
gdcm::Printer, 985
XMLDictReader
gdcm::XMLDictReader, 1540
XMLPrinter
gdcm::XMLPrinter, 1543
XMLPrivateDictReader
gdcm::XMLPrivateDictReader, 1547
XRay3DAngiographicImageStorage
gdcm::MediaStorage, 799
XRay3DCraniofacialImageStorage
gdcm::MediaStorage, 800
XRayAngiographicBiPlaneImageStorageRetired
gdcm::MediaStorage, 799

XRayAngiographicImageStorage
gdcm::MediaStorage, 799
XRayRadiationDoseSR
gdcm::MediaStorage, 800
XRayRadiofluoroscopingImageStorage
gdcm::MediaStorage, 799

YBR2RGB
gdcm::ImageChangePhotometricInterpretation, 644
YBR_FULL
gdcm::PhotometricInterpretation, 928
YBR_FULL_422
gdcm::PhotometricInterpretation, 928
YBR_ICT
gdcm::PhotometricInterpretation, 928
YBR_PARTIAL_420
gdcm::PhotometricInterpretation, 928
YBR_PARTIAL_422
gdcm::PhotometricInterpretation, 928
YBR_RCT
gdcm::PhotometricInterpretation, 928
yellow
gdcm::terminal, 110
YES
gdcm::Surface, 1202

ZEROED_OUT
gdcm::CSAHeader, 351
ZSpacing
gdcm::IPPSorter, 719
ZTolerance
gdcm::IPPSorter, 719