GDCM

2.4.4

Generated by Doxygen 1.8.8

Sun Nov 23 2014 16:21:04

Contents

1 GDCM Documentation **1**

2 off-screen rendering of DICOM images **3**

2.1 SYNOPSIS . . . 3
2.2 DESCRIPTION . . . 3
2.3 PARAMETERS . . . 3
2.4 OPTIONS . . . 3
2.4.1 OPTIONS . . . 3
2.4.2 general options . . . 3
2.5 Simple usage . . . 4
2.6 SEE ALSO . . . 4
2.7 COPYRIGHT . . . 4

3 Convert a file supported by VTK into DICOM. **5**

3.1 SYNOPSIS . . . 5
3.2 DESCRIPTION . . . 5
3.3 PARAMETERS . . . 5
3.4 OPTIONS . . . 5
3.4.1 OPTIONS . . . 5
3.4.2 compression options . . . 6
3.4.3 general options . . . 6
3.4.4 environment variable . . . 6
3.5 DESCRIPTION . . . 6
3.5.1 CONVERT MetaImage (mhd, mha) . . . 6
3.5.2 CONVERT MHA/MHD . . . 7
3.5.3 CONVERT VTI . . . 7
3.5.4 CONVERT VTK . . . 7
3.6 CONVERT DICOM . . . 7
3.7 RoundTrip DICOM to MHD to DICOM . . . 7

3.8 gdcm2vtk notes . . . 7
3.9 SEE ALSO . . . 8
3.10 COPYRIGHT . . . 8

4 Tool to anonymize a DICOM file. 9
4.1 SYNOPSIS . . . 9
4.2 DESCRIPTION . . . 9
4.3 PARAMETERS . . . 9
4.4 OPTIONS . . . 10
4.4.1 Required parameters . . . 10
4.4.2 OPTIONS . . . 10
4.4.3 encryption options . . . 10
4.4.4 dumb mode options . . . 10
4.4.5 general options . . . 10
4.4.6 environment variable . . . 11
4.5 Typical usage . . . 11
4.5.1 De-identification (anonymization, encrypt) . . . 11
4.5.2 Re-identification (de-anonymization,decrypt) . . . 11
4.5.3 Multiple files caveat . . . 11
4.5.4 Dumb mode . . . 11
4.5.4.1 Irreversible Anonymization . . . 12
4.6 OpenSSL . . . 12
4.6.1 Generating a Private Key . . . 12
4.6.2 Generating a Certificate . . . 13
4.7 DICOM Standard: . . . 13
4.8 Warnings . . . 13
4.9 SEE ALSO . . . 13
4.10 COPYRIGHT . . . 13

5 Tool to convert DICOM to DICOM. 15
5.1 SYNOPSIS . . . 15
5.2 DESCRIPTION . . . 15
5.3 PARAMETERS . . . 15
5.4 OPTIONS . . . 15
5.4.1 PARAMETERS . . . 15
5.4.2 OPTIONS . . . 15
5.4.3 image options . . . 16
5.4.4 JPEG options . . . 16

5.4.5 JPEG-LS options . . . 16
5.4.6 J2K options . . . 16
5.4.7 general options . . . 16
5.4.8 special options . . . 16
5.4.9 environment variable . . . 17
5.5 Simple usage . . . 17
5.6 Typical usage . . . 17
5.6.1 File Meta Header . . . 17
5.6.2 Conversion to Explicit Transfer Syntax . . . 18
5.6.3 Compressing to lossless JPEG . . . 18
5.6.4 Compressing to lossy JPEG . . . 18
5.6.5 Compressing to lossless JPEG-LS . . . 18
5.6.6 Compressing to lossy JPEG-LS . . . 18
5.6.7 Compressing to lossless J2K . . . 18
5.6.8 Compressing to lossy J2K . . . 18
5.6.9 Compressing to lossless RLE . . . 19
5.6.10 Split encapsulated DICOM: . . . 19
5.6.11 Forcing (re)compression . . . 19
5.6.12 Decompressing a Compressed DICOM . . . 19
5.6.13 Compressing an uncompressed Icon . . . 19
5.6.14 Generating an Icon . . . 20
5.6.15 Changing the planar Configuration . . . 20
5.7 Lossless Conversion . . . 20
5.8 Quality Control . . . 20
5.8.1 DCMTK / dicom3tools . . . 20
5.8.2 VIM: vimdiff . . . 21
5.8.3 vbindiff . . . 21
5.9 SEE ALSO . . . 21
5.10 COPYRIGHT . . . 21

6 dumps differences of two DICOM files 23
6.1 SYNOPSIS . . . 23
6.2 DESCRIPTION . . . 23
6.3 PARAMETERS . . . 23
6.4 OPTIONS . . . 23
6.4.1 OPTIONS . . . 23
6.4.2 general options . . . 23

6.5 Simple usage . . . 24
6.6 SEE ALSO . . . 24
6.7 COPYRIGHT . . . 24

7 dumps a DICOM file, it will display the structure and values contained in the specified DICOM file. 25
7.1 SYNOPSIS . . . 25
7.2 DESCRIPTION . . . 25
7.3 PARAMETERS . . . 25
7.4 OPTIONS . . . 25
7.4.1 OPTIONS . . . 25
7.4.2 general options . . . 26
7.4.3 special options . . . 26
7.5 Typical usage . . . 26
7.5.1 Printing Implicit Transfer Syntax . . . 26
7.5.2 Print Private Attributes . . . 27
7.5.3 SIEMENS CSA Header . . . 27
7.5.4 GEMS Protocol Data Block . . . 27
7.5.5 ELSCINT Protocol Information . . . 28
7.5.6 VEPRO Protocol Information . . . 28
7.5.7 Philips Private MR Series Data Storage (1.3.46.670589.11.0.0.12.2) . . . 29
7.5.8 Encapsulated ASN1 Structure . . . 30
7.6 SEE ALSO . . . 31
7.7 COPYRIGHT . . . 31

8 Tool to generate a DICOMDIR file from a File-Set. 33
8.1 SYNOPSIS . . . 33
8.2 DESCRIPTION . . . 33
8.3 PARAMETERS . . . 33
8.4 OPTIONS . . . 33
8.4.1 Parameters . . . 33
8.4.2 OPTIONS . . . 33
8.4.3 general options . . . 33
8.4.4 environment variable . . . 34
8.5 Typical usage . . . 34
8.6 NOTE . . . 34
8.7 SEE ALSO . . . 34
8.8 COPYRIGHT . . . 34

9 Manipulate DICOM image file. 35

9.1 SYNOPSIS . . . 35
9.2 DESCRIPTION . . . 35
9.3 PARAMETERS . . . 35
9.4 OPTIONS . . . 35
9.4.1 PARAMETERS . . . 35
9.4.2 OPTIONS . . . 36
9.4.3 fill options . . . 36
9.4.4 general options . . . 36
9.4.5 environment variable . . . 36
9.5 Supported File Format (appropriate file extension) gdcmimg . . . 36
9.6 Typical usage . . . 37
9.6.1 Remove a rectangular part of the image . . . 37
9.6.2 Convert RAW to DICOM . . . 37
9.6.3 Convert PGM/PNM/PPM to DICOM . . . 38
9.6.4 Convert RLE to DICOM . . . 38
9.6.5 Convert JPEG to DICOM . . . 38
9.6.6 Convert J2K to DICOM . . . 38
9.6.7 Specifying a SOP Class UID . . . 38
9.7 Multiple Files . . . 38
9.8 Start Offset . . . 39
9.9 Warning . . . 39
9.10 SEE ALSO . . . 39
9.11 COPYRIGHT . . . 39

10 Display meta info about the input DICOM file. 41

10.1 SYNOPSIS . . . 41
10.2 DESCRIPTION . . . 41
10.3 PARAMETERS . . . 41
10.4 OPTIONS . . . 41
10.4.1 OPTIONS . . . 41
10.4.2 general options . . . 41
10.4.3 environment variable . . . 42
10.5 Simple usage . . . 42
10.5.1 gdcmData . . . 42
10.5.2 Davie Clunie datasets: . . . 42
10.5.3 Checking the md5sum of the Pixel Data . . . 43

10.5.4 Checking if Pixel Data is lossless . . . 43
10.6 SEE ALSO . . . 43
10.7 COPYRIGHT . . . 43

11 Tool to convert PAPYRUS 3.0 to DICOM. 45
11.1 SYNOPSIS . . . 45
11.2 DESCRIPTION . . . 45
11.3 PARAMETERS . . . 45
11.4 OPTIONS . . . 45
11.4.1 PARAMETERS . . . 45
11.4.2 OPTIONS . . . 45
11.4.3 general options . . . 45
11.4.4 environment variable . . . 46
11.5 Simple usage . . . 46
11.6 SEE ALSO . . . 46
11.7 COPYRIGHT . . . 46

12 Tool to convert PDF to PDF/DICOM. 47
12.1 SYNOPSIS . . . 47
12.2 DESCRIPTION . . . 47
12.3 PARAMETERS . . . 47
12.4 OPTIONS . . . 47
12.4.1 general options . . . 47
12.5 Usage Example . . . 48
12.6 PDF Info Mapping . . . 48
12.7 SEE ALSO . . . 49
12.8 COPYRIGHT . . . 49

13 Extract Data Element Value Field. 51
13.1 SYNOPSIS . . . 51
13.2 DESCRIPTION . . . 51
13.3 PARAMETERS . . . 51
13.4 OPTIONS . . . 51
13.4.1 PARAMETERS . . . 51
13.4.2 OPTIONS . . . 51
13.4.3 general options . . . 51
13.5 Typical usage . . . 52
13.5.1 Copy Attribute Value to file . . . 52

13.5.2 Extract Pixel Data . . . 52
13.5.3 Encapsulated Syntax . . . 52
13.5.4 Extract fragments as single file . . . 53
13.6 Footnote about JPEG files . . . 54
13.7 SEE ALSO . . . 54
13.8 COPYRIGHT . . . 54

14 Scan a directory containing DICOM files. 55
14.1 SYNOPSIS . . . 55
14.2 DESCRIPTION . . . 55
14.2.1 PARAMETERS . . . 55
14.2.2 OPTIONS . . . 55
14.2.3 general options . . . 55
14.3 Typical usage . . . 56
14.4 Simple usage . . . 56
14.5 Complex usage . . . 56
14.6 SEE ALSO . . . 56
14.7 COPYRIGHT . . . 56

15 Tool to execute a DICOM Query/Retrieve operation 57
15.1 SYNOPSIS . . . 57
15.2 DESCRIPTION . . . 57
15.3 PARAMETERS . . . 57
15.4 OPTIONS . . . 57
15.4.1 OPTIONS . . . 57
15.4.2 mode options . . . 57
15.4.3 C-STORE options . . . 58
15.4.4 C-FIND/C-MOVE options . . . 58
15.4.5 C-MOVE options . . . 58
15.4.6 general options . . . 58
15.4.7 environment variable . . . 58
15.5 C-ECHO usage . . . 59
15.6 C-STORE usage . . . 59
15.7 C-FIND usage . . . 59
15.8 C-MOVE usage . . . 60
15.9 patientroot notes . . . 60
15.10 Debugging . . . 60
15.11 Port Warning . . . 60

15.12C-STORE Warnings . . . 61
15.13C-MOVE Warnings . . . 61
15.14C-FIND IMAGE level (Composite Object Instance) . . . 61
15.15Storing the Query . . . 61
15.16DICOM Public Servers . . . 62
15.17SEE ALSO . . . 62
15.18COPYRIGHT . . . 62

16 Concatenate/Extract DICOM files. 63
16.1 SYNOPSIS . . . 63
16.2 DESCRIPTION . . . 63
16.3 PARAMETERS . . . 63
16.4 OPTIONS . . . 63
16.4.1 OPTIONS . . . 63
16.4.2 general options . . . 63
16.4.3 environment variable . . . 64
16.5 Typical usage . . . 64
16.5.1 SIEMENS Mosaic . . . 64
16.6 SEE ALSO . . . 65
16.7 COPYRIGHT . . . 65

17 Simple DICOM viewer. 67
17.1 SYNOPSIS . . . 67
17.2 DESCRIPTION . . . 67
17.3 PARAMETERS . . . 67
17.4 OPTIONS . . . 67
17.4.1 OPTIONS . . . 67
17.4.2 general options . . . 67
17.5 Typical usage . . . 68
17.6 Simple usage . . . 68
17.7 Wiki Link . . . 68
17.8 SEE ALSO . . . 68
17.9 COPYRIGHT . . . 68

18 provides a tool to convert a DICOM file into a XML infoset and vice-versa. 69
18.1 SYNOPSIS . . . 69
18.2 DESCRIPTION . . . 69
18.3 PARAMETERS . . . 69

18.4 OPTIONS . . . 69
18.4.1 PARAMETERS . . . 69
18.4.2 Options for DICOM to XML: . . . 69
18.4.3 Options for XML to DICOM: . . . 70
18.4.4 general options . . . 70
18.5 SEE ALSO . . . 70
18.6 COPYRIGHT . . . 70

19 Todo List **71**

20 Deprecated List **73**

21 Bug List **75**

22 Namespace Index **77**
22.1 Namespace List . . . 77

23 Hierarchical Index **79**
23.1 Class Hierarchy . . . 79

24 Class Index **87**
24.1 Class List . . . 87

25 File Index **103**
25.1 File List . . . 103

26 Namespace Documentation **109**
26.1 gdcm Namespace Reference . . . 109
26.1.1 Detailed Description . . . 123
26.1.2 Typedef Documentation . . . 124
26.1.2.1 AEComp . . . 124
26.1.2.2 ASComp . . . 124
26.1.2.3 BOOL_FUNCTION_PFILE_PFILE_POINTER . . . 124
26.1.2.4 CSComp . . . 124
26.1.2.5 DAComp . . . 124
26.1.2.6 DTComp . . . 124
26.1.2.7 FileList . . . 124
26.1.2.8 IconImage . . . 124
26.1.2.9 LOComp . . . 124
26.1.2.10 LTComp . . . 124

26.1.2.11 MacroEntry . . . 124
26.1.2.12 NestedMacroEntries . . . 124
26.1.2.13 PNComp . . . 124
26.1.2.14 SHComp . . . 124
26.1.2.15 STComp . . . 124
26.1.2.16 TMComp . . . 124
26.1.2.17 UIComp . . . 124
26.1.2.18 UTComp . . . 124
26.1.3 Enumeration Type Documentation . . . 124
26.1.3.1 CompOperators . . . 124
26.1.3.2 ECharSet . . . 125
26.1.3.3 EQueryLevel . . . 125
26.1.3.4 EQueryType . . . 125
26.1.3.5 ERootType . . . 126
26.1.3.6 LodModeType . . . 126
26.1.4 Function Documentation . . . 126
26.1.4.1 backslash . . . 126
26.1.4.2 GetVRFromTag . . . 126
26.1.4.3 operator"!= . . . 126
26.1.4.4 operator"!= . . . 126
26.1.4.5 operator<< . . . 126
26.1.4.6 operator<< . . . 126
26.1.4.7 operator<< . . . 126
26.1.4.8 operator<< . . . 126
26.1.4.9 operator<< . . . 126
26.1.4.10 operator<< . . . 127
26.1.4.11 operator<< . . . 127
26.1.4.12 operator<< . . . 127
26.1.4.13 operator<< . . . 127
26.1.4.14 operator<< . . . 127
26.1.4.15 operator<< . . . 127
26.1.4.16 operator<< . . . 127
26.1.4.17 operator<< . . . 127
26.1.4.18 operator<< . . . 127
26.1.4.19 operator<< . . . 127
26.1.4.20 operator<< . . . 127
26.1.4.21 operator<< . . . 127

26.1.4.22 operator<< . . . 127
26.1.4.23 operator<< . . . 127
26.1.4.24 operator<< . . . 128
26.1.4.25 operator<< . . . 128
26.1.4.26 operator<< . . . 128
26.1.4.27 operator<< . . . 128
26.1.4.28 operator<< . . . 128
26.1.4.29 operator<< . . . 128
26.1.4.30 operator<< . . . 128
26.1.4.31 operator<< . . . 128
26.1.4.32 operator<< . . . 128
26.1.4.33 operator<< . . . 128
26.1.4.34 operator<< . . . 128
26.1.4.35 operator<< . . . 128
26.1.4.36 operator<< . . . 128
26.1.4.37 operator<< . . . 128
26.1.4.38 operator<< . . . 128
26.1.4.39 operator<< . . . 128
26.1.4.40 operator<< . . . 128
26.1.4.41 operator<< . . . 129
26.1.4.42 operator<< . . . 129
26.1.4.43 operator<< . . . 129
26.1.4.44 operator<< . . . 129
26.1.4.45 operator<< . . . 129
26.1.4.46 operator<< . . . 129
26.1.4.47 operator<< . . . 129
26.1.4.48 operator<< . . . 129
26.1.4.49 operator<< . . . 129
26.1.4.50 operator<< . . . 129
26.1.4.51 operator<< . . . 129
26.1.4.52 operator<< . . . 130
26.1.4.53 operator<< . . . 130
26.1.4.54 operator<< . . . 130
26.1.4.55 operator<< . . . 130
26.1.4.56 operator<< . . . 130
26.1.4.57 operator<< . . . 130
26.1.4.58 operator<< . . . 130

26.1.4.59 operator== . . . 130
26.1.4.60 operator>> . . . 130
26.1.4.61 operator>> . . . 130
26.1.4.62 operator>> . . . 130
26.1.4.63 to_string . . . 130
26.1.4.64 TYPETOENCODING . . . 131
26.1.5 Variable Documentation . . . 131
26.1.5.1 GlobalInstance . . . 131
26.1.5.2 VRBINARY . . . 131
26.2 gdcm::network Namespace Reference . . . 131
26.2.1 Enumeration Type Documentation . . . 135
26.2.1.1 EEventID . . . 135
26.2.1.2 EStateID . . . 135
26.2.2 Function Documentation . . . 136
26.2.2.1 GetStateIndex . . . 136
26.2.3 Variable Documentation . . . 136
26.2.3.1 cMaxEventID . . . 136
26.2.3.2 cMaxStateID . . . 136
26.3 gdcm::SegmentHelper Namespace Reference . . . 136
26.4 gdcm::terminal Namespace Reference . . . 136
26.4.1 Detailed Description . . . 137
26.4.2 Enumeration Type Documentation . . . 137
26.4.2.1 Attribute . . . 137
26.4.2.2 Color . . . 137
26.4.2.3 Mode . . . 138
26.4.3 Function Documentation . . . 138
26.4.3.1 setattribute . . . 138
26.4.3.2 setbgcolor . . . 138
26.4.3.3 setfgcolor . . . 138
26.4.3.4 setmode . . . 138

27 Class Documentation **139**

27.1 gdcm::network::AAbortPDU Class Reference . . . 139
27.1.1 Detailed Description . . . 140
27.1.2 Constructor & Destructor Documentation . . . 140
27.1.2.1 AAbortPDU . . . 140
27.1.3 Member Function Documentation . . . 140

27.1.3.1 IsLastFragment . . . 140
27.1.3.2 Print . . . 140
27.1.3.3 Read . . . 140
27.1.3.4 SetReason . . . 141
27.1.3.5 SetSource . . . 141
27.1.3.6 Size . . . 141
27.1.3.7 Write . . . 141
27.2 gdcm::network::AAssociateACPDU Class Reference . . . 141
27.2.1 Detailed Description . . . 142
27.2.2 Member Typedef Documentation . . . 143
27.2.2.1 SizeType . . . 143
27.2.3 Constructor & Destructor Documentation . . . 143
27.2.3.1 AAssociateACPDU . . . 143
27.2.4 Member Function Documentation . . . 143
27.2.4.1 AddPresentationContextAC . . . 143
27.2.4.2 GetNumberOfPresentationContextAC . . . 143
27.2.4.3 GetPresentationContextAC . . . 143
27.2.4.4 GetUserInformation . . . 143
27.2.4.5 InitFromRQ . . . 143
27.2.4.6 IsLastFragment . . . 143
27.2.4.7 Print . . . 143
27.2.4.8 Read . . . 143
27.2.4.9 SetCalledAETitle . . . 143
27.2.4.10 SetCallingAETitle . . . 143
27.2.4.11 Size . . . 143
27.2.4.12 Write . . . 143
27.2.5 Friends And Related Function Documentation . . . 143
27.2.5.1 AAssociateRQPDU . . . 144
27.3 gdcm::network::AAssociateRJPDU Class Reference . . . 144
27.3.1 Detailed Description . . . 145
27.3.2 Constructor & Destructor Documentation . . . 145
27.3.2.1 AAssociateRJPDU . . . 145
27.3.3 Member Function Documentation . . . 145
27.3.3.1 IsLastFragment . . . 145
27.3.3.2 Print . . . 145
27.3.3.3 Read . . . 145
27.3.3.4 Size . . . 145

27.3.3.5 Write . . . 145
27.4 gdcm::network::AAssociateRQPDU Class Reference . . . 145
27.4.1 Detailed Description . . . 147
27.4.2 Member Typedef Documentation . . . 147
27.4.2.1 PresentationContextArrayType . . . 147
27.4.2.2 SizeType . . . 147
27.4.3 Constructor & Destructor Documentation . . . 147
27.4.3.1 AAssociateRQPDU . . . 147
27.4.3.2 AAssociateRQPDU . . . 147
27.4.4 Member Function Documentation . . . 148
27.4.4.1 AddPresentationContext . . . 148
27.4.4.2 GetCalledAETitle . . . 148
27.4.4.3 GetCallingAETitle . . . 148
27.4.4.4 GetNumberOfPresentationContext . . . 148
27.4.4.5 GetPresentationContext . . . 148
27.4.4.6 GetPresentationContextByAbstractSyntax . . . 148
27.4.4.7 GetPresentationContextByID . . . 148
27.4.4.8 GetPresentationContexts . . . 148
27.4.4.9 GetReserved43_74 . . . 148
27.4.4.10 GetUserInformation . . . 148
27.4.4.11 IsAETitleValid . . . 148
27.4.4.12 IsLastFragment . . . 148
27.4.4.13 Print . . . 148
27.4.4.14 Read . . . 148
27.4.4.15 SetCalledAETitle . . . 148
27.4.4.16 SetCallingAETitle . . . 149
27.4.4.17 SetUserInformation . . . 149
27.4.4.18 Size . . . 149
27.4.4.19 Write . . . 149
27.4.5 Friends And Related Function Documentation . . . 149
27.4.5.1 AAssociateACPDU . . . 149
27.5 gdcm::AbortEvent Class Reference . . . 149
27.6 gdcm::network::AbstractSyntax Class Reference . . . 150
27.6.1 Detailed Description . . . 150
27.6.2 Constructor & Destructor Documentation . . . 151
27.6.2.1 AbstractSyntax . . . 151
27.6.3 Member Function Documentation . . . 151

27.6.3.1 GetAsDataElement . . . 151
27.6.3.2 GetName . . . 151
27.6.3.3 operator== . . . 151
27.6.3.4 Print . . . 151
27.6.3.5 Read . . . 151
27.6.3.6 SetName . . . 151
27.6.3.7 SetNameFromUID . . . 151
27.6.3.8 Size . . . 151
27.6.3.9 Write . . . 151
27.7 gdcm::AnonymizeEvent Class Reference . . . 151
27.7.1 Detailed Description . . . 153
27.7.2 Member Typedef Documentation . . . 153
27.7.2.1 Self . . . 153
27.7.2.2 Superclass . . . 153
27.7.3 Constructor & Destructor Documentation . . . 153
27.7.3.1 AnonymizeEvent . . . 153
27.7.3.2 ∼AnonymizeEvent . . . 153
27.7.3.3 AnonymizeEvent . . . 153
27.7.4 Member Function Documentation . . . 153
27.7.4.1 CheckEvent . . . 153
27.7.4.2 GetEventName . . . 153
27.7.4.3 GetTag . . . 153
27.7.4.4 MakeObject . . . 153
27.7.4.5 SetTag . . . 153
27.8 gdcm::Anonymizer Class Reference . . . 154
27.8.1 Detailed Description . . . 155
27.8.2 Constructor & Destructor Documentation . . . 156
27.8.2.1 Anonymizer . . . 156
27.8.2.2 ∼Anonymizer . . . 156
27.8.3 Member Function Documentation . . . 156
27.8.3.1 BALCPProtect . . . 156
27.8.3.2 BasicApplicationLevelConfidentialityProfile . . . 156
27.8.3.3 CanEmptyTag . . . 157
27.8.3.4 ClearInternalUIDs . . . 157
27.8.3.5 Empty . . . 157
27.8.3.6 GetBasicApplicationLevelConfidentialityProfileAttributes . . . 157
27.8.3.7 GetCryptographicMessageSyntax . . . 157

27.8.3.8 GetFile . . . 157
27.8.3.9 New . . . 157
27.8.3.10 RecurseDataSet . . . 157
27.8.3.11 Remove . . . 157
27.8.3.12 RemoveGroupLength . . . 157
27.8.3.13 RemovePrivateTags . . . 158
27.8.3.14 RemoveRetired . . . 158
27.8.3.15 Replace . . . 158
27.8.3.16 Replace . . . 158
27.8.3.17 SetCryptographicMessageSyntax . . . 158
27.8.3.18 SetFile . . . 158
27.9 gdcm::AnyEvent Class Reference . . . 158
27.10 gdcm::network::ApplicationContext Class Reference . . . 160
27.10.1 Detailed Description . . . 160
27.10.2 Constructor & Destructor Documentation . . . 160
27.10.2.1 ApplicationContext . . . 160
27.10.3 Member Function Documentation . . . 161
27.10.3.1 GetName . . . 161
27.10.3.2 Print . . . 161
27.10.3.3 Read . . . 161
27.10.3.4 SetName . . . 161
27.10.3.5 Size . . . 161
27.10.3.6 Write . . . 161
27.11 gdcm::ApplicationEntity Class Reference . . . 161
27.11.1 Detailed Description . . . 162
27.11.2 Member Function Documentation . . . 162
27.11.2.1 IsValid . . . 162
27.11.2.2 Print . . . 162
27.11.2.3 SetBlob . . . 162
27.11.2.4 Squeeze . . . 162
27.11.3 Member Data Documentation . . . 162
27.11.3.1 Internal . . . 162
27.11.3.2 MaxLength . . . 162
27.11.3.3 MaxNumberOfComponents . . . 162
27.11.3.4 Padding . . . 162
27.11.3.5 Separator . . . 162
27.12 gdcm::network::AReleaseRPPDU Class Reference . . . 163

27.12.1 Detailed Description . . . 164
27.12.2 Constructor & Destructor Documentation . . . 164
27.12.2.1 AReleaseRPPDU . . . 164
27.12.3 Member Function Documentation . . . 164
27.12.3.1 IsLastFragment . . . 164
27.12.3.2 Print . . . 164
27.12.3.3 Read . . . 164
27.12.3.4 Size . . . 164
27.12.3.5 Write . . . 164
27.13gdcm::network::AReleaseRQPDU Class Reference . . . 164
27.13.1 Detailed Description . . . 165
27.13.2 Constructor & Destructor Documentation . . . 165
27.13.2.1 AReleaseRQPDU . . . 165
27.13.3 Member Function Documentation . . . 166
27.13.3.1 IsLastFragment . . . 166
27.13.3.2 Print . . . 166
27.13.3.3 Read . . . 166
27.13.3.4 Size . . . 166
27.13.3.5 Write . . . 166
27.14gdcm::network::ARTIMTimer Class Reference . . . 166
27.14.1 Detailed Description . . . 166
27.14.2 Constructor & Destructor Documentation . . . 167
27.14.2.1 ARTIMTimer . . . 167
27.14.3 Member Function Documentation . . . 167
27.14.3.1 GetElapsedTime . . . 167
27.14.3.2 GetHasExpired . . . 167
27.14.3.3 GetTimeout . . . 167
27.14.3.4 SetTimeout . . . 167
27.14.3.5 Start . . . 167
27.14.3.6 Stop . . . 167
27.15gdcm::ASN1 Class Reference . . . 167
27.15.1 Detailed Description . . . 168
27.15.2 Constructor & Destructor Documentation . . . 168
27.15.2.1 ASN1 . . . 168
27.15.2.2 ∼ASN1 . . . 168
27.15.3 Member Function Documentation . . . 168
27.15.3.1 ParseDump . . . 168

27.15.3.2 ParseDumpFile . . . 168
27.15.3.3 TestPBKDF2 . . . 168
27.16gdcm::network::AsynchronousOperationsWindowSub Class Reference . . . 168
27.16.1 Detailed Description . . . 168
27.16.2 Constructor & Destructor Documentation . . . 168
27.16.2.1 AsynchronousOperationsWindowSub . . . 168
27.16.3 Member Function Documentation . . . 168
27.16.3.1 Print . . . 169
27.16.3.2 Read . . . 169
27.16.3.3 Size . . . 169
27.16.3.4 Write . . . 169
27.17gdcm::Attribute< Group, Element, TVR, TVM > Class Template Reference . . . 169
27.17.1 Detailed Description . . . 170
27.17.2 Member Typedef Documentation . . . 171
27.17.2.1 ArrayType . . . 171
27.17.3 Member Enumeration Documentation . . . 171
27.17.3.1 anonymous enum . . . 171
27.17.4 Member Function Documentation . . . 171
27.17.4.1 GDCM_STATIC_ASSERT . . . 171
27.17.4.2 GDCM_STATIC_ASSERT . . . 171
27.17.4.3 GDCM_STATIC_ASSERT . . . 171
27.17.4.4 GetAsDataElement . . . 171
27.17.4.5 GetDictVM . . . 171
27.17.4.6 GetDictVR . . . 172
27.17.4.7 GetNumberOfValues . . . 172
27.17.4.8 GetTag . . . 172
27.17.4.9 GetValue . . . 172
27.17.4.10GetValue . . . 172
27.17.4.11GetValues . . . 173
27.17.4.12GetVM . . . 173
27.17.4.13GetVR . . . 173
27.17.4.14operator"!= . . . 173
27.17.4.15operator< . . . 173
27.17.4.16operator== . . . 173
27.17.4.17operator[] . . . 174
27.17.4.18operator[] . . . 174
27.17.4.19Print . . . 174

27.17.4.20 Set . . . 174
27.17.4.21 SetByteValue . . . 174
27.17.4.22 SetByteValueNoSwap . . . 174
27.17.4.23 SetFromDataElement . . . 175
27.17.4.24 SetFromDataSet . . . 175
27.17.4.25 SetValue . . . 175
27.17.4.26 SetValues . . . 175
27.17.5 Member Data Documentation . . . 175
27.17.5.1 Internal . . . 175
27.18 gdcm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference . . . 176
27.18.1 Member Typedef Documentation . . . 177
27.18.1.1 ArrayType . . . 177
27.18.2 Member Enumeration Documentation . . . 177
27.18.2.1 anonymous enum . . . 177
27.18.3 Member Function Documentation . . . 177
27.18.3.1 GDCM_STATIC_ASSERT . . . 177
27.18.3.2 GDCM_STATIC_ASSERT . . . 177
27.18.3.3 GDCM_STATIC_ASSERT . . . 178
27.18.3.4 GDCM_STATIC_ASSERT . . . 178
27.18.3.5 GetAsDataElement . . . 178
27.18.3.6 GetDictVM . . . 178
27.18.3.7 GetDictVR . . . 178
27.18.3.8 GetNumberOfValues . . . 178
27.18.3.9 GetTag . . . 178
27.18.3.10 GetValue . . . 178
27.18.3.11 GetValue . . . 178
27.18.3.12 GetValues . . . 178
27.18.3.13 GetVM . . . 178
27.18.3.14 GetVR . . . 179
27.18.3.15 operator"!= . . . 179
27.18.3.16 operator< . . . 179
27.18.3.17 operator== . . . 179
27.18.3.18 Print . . . 179
27.18.3.19 Set . . . 179
27.18.3.20 SetByteValue . . . 179
27.18.3.21 SetByteValueNoSwap . . . 179
27.18.3.22 SetFromDataElement . . . 180

27.18.3.23 SetFromDataSet . . . 180
27.18.3.24 SetValue . . . 180
27.18.4 Member Data Documentation . . . 180
27.18.4.1 Internal . . . 180
27.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference . . . 180
27.19.1 Member Function Documentation . . . 181
27.19.1.1 GetVM . . . 181
27.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference . . . 181
27.20.1 Member Function Documentation . . . 182
27.20.1.1 GetVM . . . 182
27.21 gdcm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference . . . 183
27.21.1 Member Typedef Documentation . . . 184
27.21.1.1 ArrayType . . . 184
27.21.2 Constructor & Destructor Documentation . . . 184
27.21.2.1 Attribute . . . 184
27.21.2.2 ~Attribute . . . 184
27.21.3 Member Function Documentation . . . 184
27.21.3.1 GDCM_STATIC_ASSERT . . . 184
27.21.3.2 GDCM_STATIC_ASSERT . . . 184
27.21.3.3 GDCM_STATIC_ASSERT . . . 184
27.21.3.4 GetAsDataElement . . . 184
27.21.3.5 GetDictVM . . . 185
27.21.3.6 GetDictVR . . . 185
27.21.3.7 GetNumberOfValues . . . 185
27.21.3.8 GetTag . . . 185
27.21.3.9 GetValue . . . 185
27.21.3.10 GetValue . . . 185
27.21.3.11 GetValues . . . 185
27.21.3.12 GetVM . . . 185
27.21.3.13 GetVR . . . 185
27.21.3.14 operator[] . . . 185
27.21.3.15 operator[] . . . 185
27.21.3.16 Print . . . 186
27.21.3.17 Set . . . 186
27.21.3.18 SetByteValue . . . 186
27.21.3.19 SetFromDataElement . . . 186
27.21.3.20 SetFromDataSet . . . 186

27.21.3.21SetNumberOfValues . . . 186
27.21.3.22SetValue . . . 186
27.21.3.23SetValue . . . 186
27.21.3.24SetValues . . . 187
27.22gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference . . . 187
27.22.1 Member Function Documentation . . . 188
27.22.1.1 GetVM . . . 188
27.23gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference . . . 188
27.23.1 Member Function Documentation . . . 189
27.23.1.1 GetVM . . . 190
27.24gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference . . . 190
27.24.1 Member Function Documentation . . . 191
27.24.1.1 GetVM . . . 191
27.25gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference . . . 191
27.25.1 Member Function Documentation . . . 192
27.25.1.1 GetVM . . . 193
27.26gdcm::AudioCodec Class Reference . . . 193
27.26.1 Detailed Description . . . 194
27.26.2 Constructor & Destructor Documentation . . . 194
27.26.2.1 AudioCodec . . . 194
27.26.2.2 ∼AudioCodec . . . 194
27.26.3 Member Function Documentation . . . 194
27.26.3.1 CanCode . . . 194
27.26.3.2 CanDecode . . . 195
27.26.3.3 Decode . . . 195
27.27gdcm::Base64 Class Reference . . . 195
27.27.1 Detailed Description . . . 195
27.27.2 Member Function Documentation . . . 195
27.27.2.1 Decode . . . 195
27.27.2.2 Encode . . . 196
27.27.2.3 GetDecodeLength . . . 196
27.27.2.4 GetEncodeLength . . . 196
27.28gdcm::network::BaseCompositeMessage Class Reference . . . 196
27.28.1 Detailed Description . . . 197
27.28.2 Constructor & Destructor Documentation . . . 198
27.28.2.1 ∼BaseCompositeMessage . . . 198
27.28.3 Member Function Documentation . . . 198

27.28.3.1 ConstructPDV . . . 198
27.29gdcm::network::BasePDU Class Reference . . . 198
27.29.1 Detailed Description . . . 199
27.29.2 Constructor & Destructor Documentation . . . 200
27.29.2.1 ∼BasePDU . . . 200
27.29.3 Member Function Documentation . . . 200
27.29.3.1 IsLastFragment . . . 200
27.29.3.2 Print . . . 200
27.29.3.3 Read . . . 200
27.29.3.4 Size . . . 200
27.29.3.5 Write . . . 200
27.30gdcm::BaseRootQuery Class Reference . . . 200
27.30.1 Detailed Description . . . 202
27.30.2 Constructor & Destructor Documentation . . . 202
27.30.2.1 BaseRootQuery . . . 202
27.30.2.2 ∼BaseRootQuery . . . 202
27.30.3 Member Function Documentation . . . 203
27.30.3.1 AddQueryDataSet . . . 203
27.30.3.2 Construct . . . 203
27.30.3.3 GetAbstractSyntaxUID . . . 203
27.30.3.4 GetQueryDataSet . . . 203
27.30.3.5 GetQueryDataSet . . . 203
27.30.3.6 GetQueryLevelFromQueryRoot . . . 203
27.30.3.7 GetQueryLevelFromString . . . 203
27.30.3.8 GetQueryLevelString . . . 203
27.30.3.9 GetTagListByLevel . . . 203
27.30.3.10 InitializeDataSet . . . 203
27.30.3.11 Print . . . 203
27.30.3.12 SetSearchParameter . . . 203
27.30.3.13 SetSearchParameter . . . 203
27.30.3.14 SetSearchParameter . . . 204
27.30.3.15 ValidateQuery . . . 204
27.30.3.16 WriteHelpFile . . . 204
27.30.3.17 WriteQuery . . . 204
27.30.4 Friends And Related Function Documentation . . . 204
27.30.4.1 QueryFactory . . . 204
27.30.5 Member Data Documentation . . . 204

27.30.5.1 mDataSet . . . 204
27.30.5.2 mHelpDescription . . . 204
27.30.5.3 mImage . . . 204
27.30.5.4 mPatient . . . 204
27.30.5.5 mRootType . . . 204
27.30.5.6 mSeries . . . 204
27.30.5.7 mStudy . . . 204
27.31 gdcm::SegmentHelper::BasicCodedEntry Struct Reference . . . 204
27.31.1 Detailed Description . . . 206
27.31.2 Constructor & Destructor Documentation . . . 206
27.31.2.1 BasicCodedEntry . . . 206
27.31.2.2 BasicCodedEntry . . . 206
27.31.2.3 BasicCodedEntry . . . 206
27.31.3 Member Function Documentation . . . 206
27.31.3.1 IsEmpty . . . 206
27.31.4 Member Data Documentation . . . 206
27.31.4.1 CM . . . 206
27.31.4.2 CSD . . . 206
27.31.4.3 CSV . . . 206
27.31.4.4 CV . . . 207
27.32 gdcm::BasicOffsetTable Class Reference . . . 207
27.32.1 Detailed Description . . . 208
27.32.2 Constructor & Destructor Documentation . . . 208
27.32.2.1 BasicOffsetTable . . . 208
27.32.3 Member Function Documentation . . . 208
27.32.3.1 Read . . . 209
27.32.4 Friends And Related Function Documentation . . . 209
27.32.4.1 operator<< . . . 209
27.33 gdcm::Bitmap Class Reference . . . 209
27.33.1 Detailed Description . . . 212
27.33.2 Member Typedef Documentation . . . 212
27.33.2.1 LUTPtr . . . 212
27.33.3 Constructor & Destructor Documentation . . . 212
27.33.3.1 Bitmap . . . 212
27.33.3.2 ~Bitmap . . . 212
27.33.4 Member Function Documentation . . . 212
27.33.4.1 AreOverlaysInPixelData . . . 212

27.33.4.2 Clear . . . 212
27.33.4.3 ComputeLossyFlag . . . 212
27.33.4.4 GetBuffer . . . 212
27.33.4.5 GetBuffer2 . . . 212
27.33.4.6 GetBufferLength . . . 212
27.33.4.7 GetColumns . . . 213
27.33.4.8 GetDataElement . . . 213
27.33.4.9 GetDataElement . . . 213
27.33.4.10 GetDimension . . . 213
27.33.4.11 GetDimensions . . . 213
27.33.4.12 GetLUT . . . 213
27.33.4.13 GetLUT . . . 213
27.33.4.14 GetNeedByteSwap . . . 213
27.33.4.15 GetNumberOfDimensions . . . 213
27.33.4.16 GetPhotometricInterpretation . . . 213
27.33.4.17 GetPixelFormat . . . 214
27.33.4.18 GetPixelFormat . . . 214
27.33.4.19 GetPlanarConfiguration . . . 214
27.33.4.20 GetRows . . . 214
27.33.4.21 GetTransferSyntax . . . 214
27.33.4.22 IsEmpty . . . 214
27.33.4.23 IsLossy . . . 214
27.33.4.24 IsTransferSyntaxCompatible . . . 214
27.33.4.25 Print . . . 214
27.33.4.26 SetColumns . . . 214
27.33.4.27 SetDataElement . . . 214
27.33.4.28 SetDimension . . . 215
27.33.4.29 SetDimensions . . . 215
27.33.4.30 SetLossyFlag . . . 215
27.33.4.31 SetLUT . . . 215
27.33.4.32 SetNeedByteSwap . . . 215
27.33.4.33 SetNumberOfDimensions . . . 215
27.33.4.34 SetPhotometricInterpretation . . . 215
27.33.4.35 SetPixelFormat . . . 215
27.33.4.36 SetPlanarConfiguration . . . 215
27.33.4.37 SetRows . . . 216
27.33.4.38 SetTransferSyntax . . . 216

27.33.4.39TryJPEG2000Codec . . . 216
27.33.4.40TryJPEG2000Codec2 . . . 216
27.33.4.41TryJPEGCodec . . . 216
27.33.4.42TryJPEGCodec2 . . . 216
27.33.4.43TryJPEGLSCodec . . . 216
27.33.4.44TryKAKADUCodec . . . 216
27.33.4.45TryPVRGCodec . . . 216
27.33.4.46TryRAWCodec . . . 216
27.33.4.47TryRLECodec . . . 216
27.33.5 Friends And Related Function Documentation . . . 216
27.33.5.1 ImageChangeTransferSyntax . . . 216
27.33.5.2 PixmapReader . . . 216
27.33.6 Member Data Documentation . . . 216
27.33.6.1 Dimensions . . . 216
27.33.6.2 LossyFlag . . . 216
27.33.6.3 LUT . . . 216
27.33.6.4 NeedByteSwap . . . 216
27.33.6.5 NumberOfDimensions . . . 216
27.33.6.6 PF . . . 216
27.33.6.7 PI . . . 216
27.33.6.8 PixelData . . . 216
27.33.6.9 PlanarConfiguration . . . 217
27.33.6.10TS . . . 217
27.34gdcm::BitmapToBitmapFilter Class Reference . . . 217
27.34.1 Detailed Description . . . 218
27.34.2 Constructor & Destructor Documentation . . . 218
27.34.2.1 BitmapToBitmapFilter . . . 218
27.34.2.2 ∼BitmapToBitmapFilter . . . 218
27.34.3 Member Function Documentation . . . 218
27.34.3.1 GetOutput . . . 218
27.34.3.2 GetOutputAsBitmap . . . 218
27.34.3.3 SetInput . . . 218
27.34.4 Member Data Documentation . . . 218
27.34.4.1 Input . . . 218
27.34.4.2 Output . . . 218
27.35gdcm::BoxRegion Class Reference . . . 219
27.35.1 Detailed Description . . . 220

27.35.2 Constructor & Destructor Documentation . . . 220
27.35.2.1 BoxRegion . . . 220
27.35.2.2 ∼BoxRegion . . . 220
27.35.2.3 BoxRegion . . . 220
27.35.3 Member Function Documentation . . . 220
27.35.3.1 Area . . . 220
27.35.3.2 BoundingBox . . . 221
27.35.3.3 Clone . . . 221
27.35.3.4 ComputeBoundingBox . . . 221
27.35.3.5 Empty . . . 221
27.35.3.6 GetXMax . . . 221
27.35.3.7 GetXMin . . . 221
27.35.3.8 GetYMax . . . 221
27.35.3.9 GetYMin . . . 221
27.35.3.10 GetZMax . . . 221
27.35.3.11 GetZMin . . . 221
27.35.3.12 IsValid . . . 221
27.35.3.13 operator= . . . 221
27.35.3.14 Print . . . 221
27.35.3.15 SetDomain . . . 222
27.36 gdcm::ByteBuffer Class Reference . . . 222
27.36.1 Detailed Description . . . 222
27.36.2 Constructor & Destructor Documentation . . . 222
27.36.2.1 ByteBuffer . . . 222
27.36.3 Member Function Documentation . . . 222
27.36.3.1 Get . . . 222
27.36.3.2 GetStart . . . 222
27.36.3.3 ShiftEnd . . . 222
27.36.3.4 UpdatePosition . . . 222
27.37 gdcm::ByteSwap< T > Class Template Reference . . . 223
27.37.1 Detailed Description . . . 223
27.37.2 Member Function Documentation . . . 223
27.37.2.1 Swap . . . 223
27.37.2.2 SwapFromSwapCodeIntoSystem . . . 223
27.37.2.3 SwapRange . . . 223
27.37.2.4 SwapRangeFromSwapCodeIntoSystem . . . 223
27.37.2.5 SystemIsBigEndian . . . 223

27.37.2.6 SystemIsLittleEndian . . . 224
27.38 gdcm::ByteSwapFilter Class Reference . . . 224
27.38.1 Detailed Description . . . 224
27.38.2 Constructor & Destructor Documentation . . . 224
27.38.2.1 ByteSwapFilter . . . 224
27.38.2.2 ∼ByteSwapFilter . . . 224
27.38.3 Member Function Documentation . . . 224
27.38.3.1 ByteSwap . . . 224
27.38.3.2 SetByteSwapTag . . . 224
27.39 gdcm::ByteValue Class Reference . . . 224
27.39.1 Detailed Description . . . 226
27.39.2 Constructor & Destructor Documentation . . . 226
27.39.2.1 ByteValue . . . 226
27.39.2.2 ByteValue . . . 227
27.39.2.3 ∼ByteValue . . . 227
27.39.3 Member Function Documentation . . . 227
27.39.3.1 Append . . . 227
27.39.3.2 Clear . . . 227
27.39.3.3 ComputeLength . . . 227
27.39.3.4 Fill . . . 227
27.39.3.5 GetBuffer . . . 227
27.39.3.6 GetLength . . . 227
27.39.3.7 GetPointer . . . 228
27.39.3.8 IsEmpty . . . 228
27.39.3.9 IsPrintable . . . 228
27.39.3.10 operator const std::vector< char > & . . . 228
27.39.3.11 operator= . . . 228
27.39.3.12 operator== . . . 228
27.39.3.13 operator== . . . 228
27.39.3.14 Print . . . 228
27.39.3.15 PrintASCII . . . 228
27.39.3.16 PrintASCIIXML . . . 228
27.39.3.17 PrintGroupLength . . . 228
27.39.3.18 PrintHex . . . 228
27.39.3.19 PrintHexXML . . . 228
27.39.3.20 PrintPNXML . . . 228
27.39.3.21 Read . . . 229

27.39.3.22 Read . . . 229
27.39.3.23 SetLength . . . 229
27.39.3.24 SetLengthOnly . . . 229
27.39.3.25 Write . . . 229
27.39.3.26 Write . . . 229
27.39.3.27 WriteBuffer . . . 229
27.40 gdcm::CAPICryptoFactory Class Reference . . . 229
27.40.1 Constructor & Destructor Documentation . . . 230
27.40.1.1 CAPICryptoFactory . . . 230
27.40.2 Member Function Documentation . . . 230
27.40.2.1 CreateCMSProvider . . . 230
27.41 gdcm::CAPICryptographicMessageSyntax Class Reference . . . 230
27.41.1 Constructor & Destructor Documentation . . . 232
27.41.1.1 CAPICryptographicMessageSyntax . . . 232
27.41.1.2 ∼CAPICryptographicMessageSyntax . . . 232
27.41.2 Member Function Documentation . . . 232
27.41.2.1 Decrypt . . . 232
27.41.2.2 Encrypt . . . 232
27.41.2.3 GetCipherType . . . 232
27.41.2.4 GetInitialized . . . 232
27.41.2.5 ParseCertificateFile . . . 232
27.41.2.6 ParseKeyFile . . . 232
27.41.2.7 SetCipherType . . . 232
27.41.2.8 SetPassword . . . 232
27.42 gdcm::network::CEchoRQ Class Reference . . . 233
27.42.1 Detailed Description . . . 234
27.42.2 Member Function Documentation . . . 234
27.42.2.1 ConstructPDV . . . 234
27.42.3 Member Data Documentation . . . 234
27.42.3.1 AffectedSOPClassUID . . . 234
27.42.3.2 MessageID . . . 234
27.43 gdcm::network::CEchoRSP Class Reference . . . 234
27.43.1 Detailed Description . . . 235
27.43.2 Member Function Documentation . . . 235
27.43.2.1 ConstructPDVByDataSet . . . 235
27.44 gdcm::network::CFind Class Reference . . . 235
27.44.1 Detailed Description . . . 235

27.45gdcm::network::CFindCancelRQ Class Reference . . . 236
27.45.1 Detailed Description . . . 236
27.45.2 Member Function Documentation . . . 236
27.45.2.1 ConstructPDVByDataSet . . . 237
27.46gdcm::network::CFindRQ Class Reference . . . 237
27.46.1 Detailed Description . . . 238
27.46.2 Member Function Documentation . . . 238
27.46.2.1 ConstructPDV . . . 238
27.47gdcm::network::CFindRSP Class Reference . . . 238
27.47.1 Detailed Description . . . 239
27.47.2 Member Function Documentation . . . 239
27.47.2.1 ConstructPDVByDataSet . . . 239
27.48gdcm::network::CMoveCancelRq Class Reference . . . 239
27.48.1 Member Function Documentation . . . 240
27.48.1.1 ConstructPDVByDataSet . . . 240
27.49gdcm::network::CMoveRQ Class Reference . . . 241
27.49.1 Detailed Description . . . 241
27.49.2 Member Function Documentation . . . 242
27.49.2.1 ConstructPDV . . . 242
27.50gdcm::network::CMoveRSP Class Reference . . . 242
27.50.1 Detailed Description . . . 243
27.50.2 Member Function Documentation . . . 243
27.50.2.1 ConstructPDVByDataSet . . . 243
27.51gdcm::Codec Class Reference . . . 243
27.51.1 Detailed Description . . . 244
27.52gdcm::Coder Class Reference . . . 244
27.52.1 Detailed Description . . . 245
27.52.2 Constructor & Destructor Documentation . . . 245
27.52.2.1 ∼Coder . . . 245
27.52.3 Member Function Documentation . . . 245
27.52.3.1 CanCode . . . 245
27.52.3.2 Code . . . 246
27.52.3.3 InternalCode . . . 246
27.53gdcm::CodeString Class Reference . . . 246
27.53.1 Detailed Description . . . 247
27.53.2 Member Typedef Documentation . . . 247
27.53.2.1 const_iterator . . . 247

27.53.2.2 const_reference . . . 247
27.53.2.3 const_reverse_iterator . . . 247
27.53.2.4 difference_type . . . 247
27.53.2.5 iterator . . . 247
27.53.2.6 pointer . . . 247
27.53.2.7 reference . . . 247
27.53.2.8 reverse_iterator . . . 247
27.53.2.9 size_type . . . 248
27.53.2.10 value_type . . . 248
27.53.3 Constructor & Destructor Documentation . . . 248
27.53.3.1 CodeString . . . 248
27.53.3.2 CodeString . . . 248
27.53.3.3 CodeString . . . 248
27.53.3.4 CodeString . . . 248
27.53.4 Member Function Documentation . . . 248
27.53.4.1 GetAsString . . . 248
27.53.4.2 IsValid . . . 248
27.53.4.3 Size . . . 248
27.53.4.4 TrimInternal . . . 248
27.53.5 Friends And Related Function Documentation . . . 248
27.53.5.1 operator"!= . . . 248
27.53.5.2 operator<< . . . 248
27.53.5.3 operator== . . . 248
27.54 gdcm::Command Class Reference . . . 248
27.54.1 Detailed Description . . . 250
27.54.2 Constructor & Destructor Documentation . . . 250
27.54.2.1 Command . . . 250
27.54.2.2 ~Command . . . 250
27.54.3 Member Function Documentation . . . 250
27.54.3.1 Execute . . . 250
27.54.3.2 Execute . . . 250
27.55 gdcm::CommandDataSet Class Reference . . . 250
27.55.1 Detailed Description . . . 252
27.55.2 Constructor & Destructor Documentation . . . 252
27.55.2.1 CommandDataSet . . . 252
27.55.2.2 ~CommandDataSet . . . 252
27.55.3 Member Function Documentation . . . 252

27.55.3.1 Insert . . . 252
27.55.3.2 Read . . . 252
27.55.3.3 Replace . . . 252
27.55.3.4 Write . . . 252
27.55.4 Friends And Related Function Documentation . . . 252
27.55.4.1 operator<< . . . 252
27.56gdcm::network::CompositeMessageFactory Class Reference . . . 252
27.56.1 Detailed Description . . . 253
27.56.2 Member Function Documentation . . . 253
27.56.2.1 ConstructCEchoRQ . . . 253
27.56.2.2 ConstructCFindRQ . . . 253
27.56.2.3 ConstructCMoveRQ . . . 253
27.56.2.4 ConstructCStoreRQ . . . 253
27.56.2.5 ConstructCStoreRSP . . . 253
27.57gdcm::CompositeNetworkFunctions Class Reference . . . 253
27.57.1 Detailed Description . . . 254
27.57.2 Member Typedef Documentation . . . 254
27.57.2.1 KeyValuePairArrayType . . . 254
27.57.2.2 KeyValuePairType . . . 255
27.57.3 Member Function Documentation . . . 255
27.57.3.1 CEcho . . . 255
27.57.3.2 CFind . . . 255
27.57.3.3 CMove . . . 255
27.57.3.4 ConstructQuery . . . 256
27.57.3.5 ConstructQuery . . . 256
27.57.3.6 CStore . . . 256
27.58gdcm::ConstCharWrapper Class Reference . . . 257
27.58.1 Detailed Description . . . 257
27.58.2 Constructor & Destructor Documentation . . . 257
27.58.2.1 ConstCharWrapper . . . 257
27.58.3 Member Function Documentation . . . 257
27.58.3.1 operator const char * . . . 257
27.59gdcm::CP246ExplicitDataElement Class Reference . . . 257
27.59.1 Detailed Description . . . 258
27.59.2 Member Function Documentation . . . 258
27.59.2.1 GetLength . . . 258
27.59.2.2 Read . . . 259

27.59.2.3 ReadPreValue . . . 259
27.59.2.4 ReadValue . . . 259
27.59.2.5 ReadWithLength . . . 259
27.60 gdcm::CryptoFactory Class Reference . . . 259
27.60.1 Detailed Description . . . 260
27.60.2 Member Enumeration Documentation . . . 260
27.60.2.1 CryptoLib . . . 260
27.60.3 Constructor & Destructor Documentation . . . 260
27.60.3.1 CryptoFactory . . . 260
27.60.3.2 CryptoFactory . . . 260
27.60.3.3 ∼CryptoFactory . . . 260
27.60.4 Member Function Documentation . . . 260
27.60.4.1 CreateCMSProvider . . . 260
27.60.4.2 GetFactoryInstance . . . 260
27.61 gdcm::CryptographicMessageSyntax Class Reference . . . 261
27.61.1 Member Enumeration Documentation . . . 261
27.61.1.1 CipherTypes . . . 261
27.61.2 Constructor & Destructor Documentation . . . 262
27.61.2.1 CryptographicMessageSyntax . . . 262
27.61.2.2 ∼CryptographicMessageSyntax . . . 262
27.61.3 Member Function Documentation . . . 262
27.61.3.1 Decrypt . . . 262
27.61.3.2 Encrypt . . . 262
27.61.3.3 GetCipherType . . . 262
27.61.3.4 ParseCertificateFile . . . 262
27.61.3.5 ParseKeyFile . . . 262
27.61.3.6 SetCipherType . . . 262
27.61.3.7 SetPassword . . . 263
27.62 gdcm::CSAElement Class Reference . . . 263
27.62.1 Detailed Description . . . 264
27.62.2 Member Typedef Documentation . . . 265
27.62.2.1 DataPtr . . . 265
27.62.3 Constructor & Destructor Documentation . . . 265
27.62.3.1 CSAElement . . . 265
27.62.3.2 CSAElement . . . 265
27.62.4 Member Function Documentation . . . 265
27.62.4.1 GetByteValue . . . 265

27.62.4.2 GetKey . . . 265
27.62.4.3 GetName . . . 265
27.62.4.4 GetNoOfItems . . . 265
27.62.4.5 GetSyngoDT . . . 265
27.62.4.6 GetValue . . . 265
27.62.4.7 GetValue . . . 266
27.62.4.8 GetVM . . . 266
27.62.4.9 GetVR . . . 266
27.62.4.10 IsEmpty . . . 266
27.62.4.11 operator< . . . 266
27.62.4.12 operator= . . . 266
27.62.4.13 operator== . . . 266
27.62.4.14 SetByteValue . . . 266
27.62.4.15 SetKey . . . 266
27.62.4.16 SetName . . . 266
27.62.4.17 SetNoOfItems . . . 266
27.62.4.18 SetSyngoDT . . . 266
27.62.4.19 SetValue . . . 266
27.62.4.20 SetVM . . . 266
27.62.4.21 SetVR . . . 266
27.62.5 Friends And Related Function Documentation . . . 267
27.62.5.1 operator<< . . . 267
27.62.6 Member Data Documentation . . . 267
27.62.6.1 DataField . . . 267
27.62.6.2 KeyField . . . 267
27.62.6.3 NameField . . . 267
27.62.6.4 NoOfItemsField . . . 267
27.62.6.5 SyngoDTField . . . 267
27.62.6.6 ValueMultiplicityField . . . 267
27.62.6.7 VRField . . . 267
27.63 gdcm::CSAHeader Class Reference . . . 267
27.63.1 Detailed Description . . . 269
27.63.2 Member Enumeration Documentation . . . 269
27.63.2.1 CSAHeaderType . . . 269
27.63.3 Constructor & Destructor Documentation . . . 269
27.63.3.1 CSAHeader . . . 269
27.63.3.2 ~CSAHeader . . . 269

27.63.4 Member Function Documentation . . . 269
27.63.4.1 FindCSAElementByName . . . 270
27.63.4.2 GetCSADataInfo . . . 270
27.63.4.3 GetCSAEEnd . . . 270
27.63.4.4 GetCSAElementByName . . . 270
27.63.4.5 GetCSAImageHeaderInfoTag . . . 270
27.63.4.6 GetCSASeriesHeaderInfoTag . . . 270
27.63.4.7 GetDataSet . . . 271
27.63.4.8 GetFormat . . . 271
27.63.4.9 GetInterfile . . . 271
27.63.4.10 LoadFromDataElement . . . 271
27.63.4.11 Print . . . 271
27.63.4.12 Read . . . 271
27.63.4.13 Write . . . 271
27.63.5 Friends And Related Function Documentation . . . 271
27.63.5.1 operator<< . . . 271
27.64 gdcm::CSAHeaderDict Class Reference . . . 271
27.64.1 Detailed Description . . . 272
27.64.2 Member Typedef Documentation . . . 272
27.64.2.1 ConstIterator . . . 272
27.64.2.2 Iterator . . . 272
27.64.2.3 MapCSAHeaderDictEntry . . . 272
27.64.3 Constructor & Destructor Documentation . . . 272
27.64.3.1 CSAHeaderDict . . . 272
27.64.4 Member Function Documentation . . . 272
27.64.4.1 AddCSAHeaderDictEntry . . . 273
27.64.4.2 Begin . . . 273
27.64.4.3 End . . . 273
27.64.4.4 GetCSAHeaderDictEntry . . . 273
27.64.4.5 IsEmpty . . . 273
27.64.4.6 LoadDefault . . . 273
27.64.5 Friends And Related Function Documentation . . . 273
27.64.5.1 Dicts . . . 273
27.64.5.2 operator<< . . . 273
27.65 gdcm::CSAHeaderDictEntry Class Reference . . . 273
27.65.1 Detailed Description . . . 274
27.65.2 Constructor & Destructor Documentation . . . 274

27.65.2.1 CSAHeaderDictEntry . . . 274
27.65.3 Member Function Documentation . . . 274
27.65.3.1 GetDescription . . . 274
27.65.3.2 GetName . . . 274
27.65.3.3 GetVM . . . 274
27.65.3.4 GetVR . . . 274
27.65.3.5 operator< . . . 274
27.65.3.6 SetDescription . . . 275
27.65.3.7 SetName . . . 275
27.65.3.8 SetVM . . . 275
27.65.3.9 SetVR . . . 275
27.65.4 Friends And Related Function Documentation . . . 275
27.65.4.1 operator<< . . . 275
27.66gdcm::CSAHeaderDictException Class Reference . . . 275
27.67gdcm::network::CStoreRQ Class Reference . . . 276
27.67.1 Detailed Description . . . 277
27.67.2 Member Function Documentation . . . 277
27.67.2.1 ConstructPDV . . . 277
27.68gdcm::network::CStoreRSP Class Reference . . . 277
27.68.1 Detailed Description . . . 278
27.68.2 Member Function Documentation . . . 278
27.68.2.1 ConstructPDV . . . 278
27.69gdcm::Curve Class Reference . . . 279
27.69.1 Detailed Description . . . 280
27.69.2 Constructor & Destructor Documentation . . . 280
27.69.2.1 Curve . . . 280
27.69.2.2 ∼Curve . . . 280
27.69.2.3 Curve . . . 280
27.69.3 Member Function Documentation . . . 280
27.69.3.1 Decode . . . 281
27.69.3.2 GetAsPoints . . . 281
27.69.3.3 GetCurveDataDescriptor . . . 281
27.69.3.4 GetDataValueRepresentation . . . 281
27.69.3.5 GetDimensions . . . 281
27.69.3.6 GetGroup . . . 281
27.69.3.7 GetNumberOfCurves . . . 281
27.69.3.8 GetNumberOfPoints . . . 281

27.69.3.9 GetTypeOfData . . . 281
27.69.3.10 GetTypeOfDataDescription . . . 281
27.69.3.11 IsEmpty . . . 281
27.69.3.12 Print . . . 281
27.69.3.13 SetCoordinateStartValue . . . 281
27.69.3.14 SetCoordinateStepValue . . . 281
27.69.3.15 SetCurve . . . 281
27.69.3.16 SetCurveDataDescriptor . . . 281
27.69.3.17 SetCurveDescription . . . 281
27.69.3.18 SetDataValueRepresentation . . . 281
27.69.3.19 SetDimensions . . . 281
27.69.3.20 SetGroup . . . 281
27.69.3.21 SetNumberOfPoints . . . 281
27.69.3.22 SetTypeOfData . . . 281
27.69.3.23 Update . . . 281
27.70 gdcm::DataElement Class Reference . . . 282
27.70.1 Detailed Description . . . 284
27.70.2 Member Typedef Documentation . . . 285
27.70.2.1 ValuePtr . . . 285
27.70.3 Constructor & Destructor Documentation . . . 285
27.70.3.1 DataElement . . . 285
27.70.3.2 DataElement . . . 285
27.70.4 Member Function Documentation . . . 285
27.70.4.1 Clear . . . 285
27.70.4.2 Empty . . . 285
27.70.4.3 GetByteValue . . . 285
27.70.4.4 GetLength . . . 285
27.70.4.5 GetSequenceOfFragments . . . 285
27.70.4.6 GetSequenceOfFragments . . . 286
27.70.4.7 GetTag . . . 286
27.70.4.8 GetTag . . . 286
27.70.4.9 GetValue . . . 286
27.70.4.10 GetValue . . . 286
27.70.4.11 GetValueAsSQ . . . 286
27.70.4.12 GetVL . . . 287
27.70.4.13 GetVL . . . 287
27.70.4.14 GetVR . . . 287

27.70.4.15 IsEmpty . . . 287
27.70.4.16 IsUndefinedLength . . . 287
27.70.4.17 operator< . . . 287
27.70.4.18 operator= . . . 287
27.70.4.19 operator== . . . 288
27.70.4.20 Read . . . 288
27.70.4.21 ReadOrSkip . . . 288
27.70.4.22 ReadPreValue . . . 288
27.70.4.23 ReadValue . . . 288
27.70.4.24 ReadValueWithLength . . . 288
27.70.4.25 ReadWithLength . . . 288
27.70.4.26 SetByteValue . . . 288
27.70.4.27 SetTag . . . 288
27.70.4.28 SetValue . . . 289
27.70.4.29 SetValueFieldLength . . . 289
27.70.4.30 SetVL . . . 289
27.70.4.31 SetVLToUndefined . . . 289
27.70.4.32 SetVR . . . 289
27.70.4.33 Write . . . 290
27.70.5 Friends And Related Function Documentation . . . 290
27.70.5.1 operator<< . . . 290
27.70.6 Member Data Documentation . . . 290
27.70.6.1 TagField . . . 290
27.70.6.2 ValueField . . . 290
27.70.6.3 ValueLengthField . . . 290
27.70.6.4 VRField . . . 290
27.71 gdcm::DataElementException Class Reference . . . 290
27.72 gdcm::DataEvent Class Reference . . . 291
27.72.1 Detailed Description . . . 292
27.72.2 Member Typedef Documentation . . . 292
27.72.2.1 Self . . . 292
27.72.2.2 Superclass . . . 292
27.72.3 Constructor & Destructor Documentation . . . 292
27.72.3.1 DataEvent . . . 293
27.72.3.2 ~DataEvent . . . 293
27.72.3.3 DataEvent . . . 293
27.72.4 Member Function Documentation . . . 293

27.72.4.1 CheckEvent . . . 293
27.72.4.2 GetData . . . 293
27.72.4.3 GetDataLength . . . 293
27.72.4.4 GetEventName . . . 293
27.72.4.5 MakeObject . . . 293
27.72.4.6 SetData . . . 293
27.73gdcm::DataSet Class Reference . . . 293
27.73.1 Detailed Description . . . 296
27.73.2 Member Typedef Documentation . . . 296
27.73.2.1 ConstIterator . . . 296
27.73.2.2 DataElementSet . . . 296
27.73.2.3 Iterator . . . 296
27.73.2.4 SizeType . . . 296
27.73.3 Member Function Documentation . . . 296
27.73.3.1 Begin . . . 296
27.73.3.2 Begin . . . 296
27.73.3.3 Clear . . . 296
27.73.3.4 ComputeDataElement . . . 297
27.73.3.5 ComputeGroupLength . . . 297
27.73.3.6 End . . . 297
27.73.3.7 End . . . 297
27.73.3.8 FindDataElement . . . 297
27.73.3.9 FindDataElement . . . 297
27.73.3.10 FindNextDataElement . . . 297
27.73.3.11 GetDataElement . . . 297
27.73.3.12 GetDataElement . . . 298
27.73.3.13 GetDEEnd . . . 298
27.73.3.14 GetDES . . . 298
27.73.3.15 GetDES . . . 298
27.73.3.16 GetLength . . . 298
27.73.3.17 GetMediaStorage . . . 298
27.73.3.18 GetPrivateCreator . . . 298
27.73.3.19 Insert . . . 298
27.73.3.20 InsertDataElement . . . 298
27.73.3.21 IsEmpty . . . 299
27.73.3.22 operator() . . . 299
27.73.3.23 operator= . . . 299

27.73.3.24operator[] . . . 299
27.73.3.25Print . . . 299
27.73.3.26Read . . . 299
27.73.3.27ReadNested . . . 299
27.73.3.28ReadSelectedPrivateTags . . . 299
27.73.3.29ReadSelectedPrivateTagsWithLength . . . 299
27.73.3.30ReadSelectedTags . . . 299
27.73.3.31ReadSelectedTagsWithLength . . . 299
27.73.3.32ReadUpToTag . . . 299
27.73.3.33ReadUpToTagWithLength . . . 299
27.73.3.34ReadWithLength . . . 299
27.73.3.35Remove . . . 299
27.73.3.36Replace . . . 300
27.73.3.37ReplaceEmpty . . . 300
27.73.3.38Size . . . 300
27.73.3.39Write . . . 300
27.73.4 Friends And Related Function Documentation . . . 300
27.73.4.1 CSAHeader . . . 300
27.73.4.2 operator<< . . . 300
27.74gdcm::DataSetEvent Class Reference . . . 300
27.74.1 Detailed Description . . . 302
27.74.2 Member Typedef Documentation . . . 302
27.74.2.1 Self . . . 302
27.74.2.2 Superclass . . . 302
27.74.3 Constructor & Destructor Documentation . . . 302
27.74.3.1 DataSetEvent . . . 302
27.74.3.2 ~DataSetEvent . . . 302
27.74.3.3 DataSetEvent . . . 302
27.74.4 Member Function Documentation . . . 302
27.74.4.1 CheckEvent . . . 302
27.74.4.2 GetDataSet . . . 302
27.74.4.3 GetEventName . . . 302
27.74.4.4 MakeObject . . . 302
27.75gdcm::DataSetHelper Class Reference . . . 303
27.75.1 Detailed Description . . . 303
27.75.2 Member Function Documentation . . . 303
27.75.2.1 ComputeVR . . . 303

27.76gdcm::Decoder Class Reference . . . 303
27.76.1 Detailed Description . . . 304
27.76.2 Constructor & Destructor Documentation . . . 304
27.76.2.1 ∼Decoder . . . 304
27.76.3 Member Function Documentation . . . 304
27.76.3.1 CanDecode . . . 304
27.76.3.2 Decode . . . 304
27.76.3.3 DecodeByStreams . . . 304
27.77gdcm::DefinedTerms Class Reference . . . 305
27.77.1 Detailed Description . . . 305
27.77.2 Constructor & Destructor Documentation . . . 305
27.77.2.1 DefinedTerms . . . 305
27.78gdcm::Defs Class Reference . . . 305
27.78.1 Detailed Description . . . 306
27.78.2 Constructor & Destructor Documentation . . . 306
27.78.2.1 Defs . . . 306
27.78.2.2 ∼Defs . . . 306
27.78.3 Member Function Documentation . . . 306
27.78.3.1 GetIODFromFile . . . 306
27.78.3.2 GetIODNameFromMediaStorage . . . 306
27.78.3.3 GetIODs . . . 307
27.78.3.4 GetIODs . . . 307
27.78.3.5 GetMacros . . . 307
27.78.3.6 GetMacros . . . 307
27.78.3.7 GetModules . . . 307
27.78.3.8 GetModules . . . 307
27.78.3.9 GetTypeFromTag . . . 307
27.78.3.10 IsEmpty . . . 307
27.78.3.11 LoadDefaults . . . 307
27.78.3.12 LoadFromFile . . . 307
27.78.3.13 Verify . . . 307
27.78.3.14 Verify . . . 307
27.78.4 Friends And Related Function Documentation . . . 307
27.78.4.1 Global . . . 307
27.79gdcm::DeltaEncodingCodec Class Reference . . . 308
27.79.1 Detailed Description . . . 309
27.79.2 Constructor & Destructor Documentation . . . 309

27.79.2.1 DeltaEncodingCodec . . . 309
27.79.2.2 ~DeltaEncodingCodec . . . 309
27.79.3 Member Function Documentation . . . 309
27.79.3.1 CanDecode . . . 309
27.79.3.2 Decode . . . 309
27.79.3.3 Decode . . . 309
27.80 gdcm::DICOMDIR Class Reference . . . 309
27.80.1 Detailed Description . . . 309
27.80.2 Constructor & Destructor Documentation . . . 310
27.80.2.1 DICOMDIR . . . 310
27.80.2.2 DICOMDIR . . . 310
27.81 gdcm::DICOMDIRGenerator Class Reference . . . 310
27.81.1 Detailed Description . . . 311
27.81.2 Member Typedef Documentation . . . 311
27.81.2.1 FilenamesType . . . 311
27.81.2.2 FilenameType . . . 311
27.81.3 Constructor & Destructor Documentation . . . 311
27.81.3.1 DICOMDIRGenerator . . . 311
27.81.3.2 ~DICOMDIRGenerator . . . 311
27.81.4 Member Function Documentation . . . 311
27.81.4.1 AddImageDirectoryRecord . . . 311
27.81.4.2 AddPatientDirectoryRecord . . . 311
27.81.4.3 AddSeriesDirectoryRecord . . . 311
27.81.4.4 AddStudyDirectoryRecord . . . 311
27.81.4.5 Generate . . . 311
27.81.4.6 GetFile . . . 312
27.81.4.7 GetScanner . . . 312
27.81.4.8 SetDescriptor . . . 312
27.81.4.9 SetFile . . . 312
27.81.4.10 SetFilenames . . . 312
27.81.4.11 SetRootDirectory . . . 312
27.82 gdcm::Dict Class Reference . . . 312
27.82.1 Detailed Description . . . 313
27.82.2 Member Typedef Documentation . . . 313
27.82.2.1 ConstIterator . . . 313
27.82.2.2 Iterator . . . 313
27.82.2.3 MapDictEntry . . . 313

27.82.3 Constructor & Destructor Documentation . . . 313
27.82.3.1 Dict . . . 313
27.82.4 Member Function Documentation . . . 313
27.82.4.1 AddDictEntry . . . 313
27.82.4.2 Begin . . . 313
27.82.4.3 End . . . 314
27.82.4.4 GetDictEntry . . . 314
27.82.4.5 GetDictEntryByKeyword . . . 314
27.82.4.6 GetDictEntryByName . . . 314
27.82.4.7 GetKeywordFromTag . . . 314
27.82.4.8 IsEmpty . . . 314
27.82.4.9 LoadDefault . . . 314
27.82.5 Friends And Related Function Documentation . . . 314
27.82.5.1 Dicts . . . 314
27.82.5.2 operator<< . . . 314
27.83gdcm::DictConverter Class Reference . . . 315
27.83.1 Detailed Description . . . 315
27.83.2 Member Enumeration Documentation . . . 316
27.83.2.1 OutputTypes . . . 316
27.83.3 Constructor & Destructor Documentation . . . 316
27.83.3.1 DictConverter . . . 316
27.83.3.2 ~DictConverter . . . 316
27.83.4 Member Function Documentation . . . 316
27.83.4.1 AddGroupLength . . . 316
27.83.4.2 Convert . . . 316
27.83.4.3 ConvertToCXX . . . 316
27.83.4.4 ConvertToXML . . . 316
27.83.4.5 GetDictName . . . 316
27.83.4.6 GetInputFilename . . . 316
27.83.4.7 GetOutputFilename . . . 316
27.83.4.8 GetOutputType . . . 316
27.83.4.9 Readuint16 . . . 316
27.83.4.10 ReadVM . . . 316
27.83.4.11 ReadVR . . . 316
27.83.4.12 SetDictName . . . 316
27.83.4.13 SetInputFileName . . . 316
27.83.4.14 SetOutputFileName . . . 316

27.83.4.15 SetOutputType . . . 316
27.83.4.16 WriteFooter . . . 317
27.83.4.17 WriteHeader . . . 317
27.84 gdcm::DictEntry Class Reference . . . 317
27.84.1 Detailed Description . . . 318
27.84.2 Constructor & Destructor Documentation . . . 318
27.84.2.1 DictEntry . . . 318
27.84.3 Member Function Documentation . . . 318
27.84.3.1 GetKeyword . . . 318
27.84.3.2 GetName . . . 318
27.84.3.3 GetRetired . . . 318
27.84.3.4 GetVM . . . 318
27.84.3.5 GetVR . . . 319
27.84.3.6 IsUnique . . . 319
27.84.3.7 SetElementXX . . . 319
27.84.3.8 SetGroupXX . . . 319
27.84.3.9 SetKeyword . . . 319
27.84.3.10 SetName . . . 319
27.84.3.11 SetRetired . . . 319
27.84.3.12 SetVM . . . 319
27.84.3.13 SetVR . . . 319
27.84.4 Friends And Related Function Documentation . . . 319
27.84.4.1 Dict . . . 319
27.84.4.2 operator<< . . . 319
27.85 gdcm::DictPrinter Class Reference . . . 319
27.85.1 Detailed Description . . . 321
27.85.2 Constructor & Destructor Documentation . . . 321
27.85.2.1 DictPrinter . . . 321
27.85.2.2 ~DictPrinter . . . 321
27.85.3 Member Function Documentation . . . 321
27.85.3.1 Print . . . 321
27.85.3.2 PrintDataElement2 . . . 321
27.85.3.3 PrintDataSet2 . . . 321
27.86 gdcm::Dicts Class Reference . . . 321
27.86.1 Detailed Description . . . 322
27.86.2 Member Enumeration Documentation . . . 322
27.86.2.1 ConstructorType . . . 322

27.86.3 Constructor & Destructor Documentation . . . 322
27.86.3.1 Dicts . . . 322
27.86.3.2 ∼Dicts . . . 322
27.86.4 Member Function Documentation . . . 322
27.86.4.1 GetConstructorString . . . 323
27.86.4.2 GetCSAHeaderDict . . . 323
27.86.4.3 GetDictEntry . . . 323
27.86.4.4 GetDictEntry . . . 323
27.86.4.5 GetPrivateDict . . . 323
27.86.4.6 GetPrivateDict . . . 323
27.86.4.7 GetPublicDict . . . 323
27.86.4.8 IsEmpty . . . 323
27.86.4.9 LoadDefaults . . . 323
27.86.5 Friends And Related Function Documentation . . . 323
27.86.5.1 Global . . . 323
27.86.5.2 operator<< . . . 323
27.87gdcm::network::DIMSE Class Reference . . . 324
27.87.1 Detailed Description . . . 324
27.87.2 Member Enumeration Documentation . . . 324
27.87.2.1 CommandTypes . . . 324
27.88gdcm::DirectionCosines Class Reference . . . 325
27.88.1 Detailed Description . . . 326
27.88.2 Constructor & Destructor Documentation . . . 326
27.88.2.1 DirectionCosines . . . 326
27.88.2.2 DirectionCosines . . . 326
27.88.2.3 ∼DirectionCosines . . . 326
27.88.3 Member Function Documentation . . . 326
27.88.3.1 ComputeDistAlongNormal . . . 326
27.88.3.2 Cross . . . 326
27.88.3.3 CrossDot . . . 326
27.88.3.4 Dot . . . 326
27.88.3.5 IsValid . . . 326
27.88.3.6 Normalize . . . 327
27.88.3.7 operator const double * . . . 327
27.88.3.8 Print . . . 327
27.88.3.9 SetFromString . . . 327
27.89gdcm::Directory Class Reference . . . 327

27.89.1 Detailed Description . . . 328
27.89.2 Member Typedef Documentation . . . 328
27.89.2.1 FilenamesType . . . 328
27.89.2.2 FilenameType . . . 328
27.89.3 Constructor & Destructor Documentation . . . 328
27.89.3.1 Directory . . . 328
27.89.3.2 ∼Directory . . . 328
27.89.4 Member Function Documentation . . . 328
27.89.4.1 Explore . . . 328
27.89.4.2 GetDirectories . . . 329
27.89.4.3 GetFilenames . . . 329
27.89.4.4 GetToplevel . . . 329
27.89.4.5 Load . . . 329
27.89.4.6 Print . . . 329
27.89.5 Friends And Related Function Documentation . . . 329
27.89.5.1 operator<< . . . 329
27.90 gdcm::DirectoryHelper Class Reference . . . 330
27.90.1 Detailed Description . . . 330
27.90.2 Member Function Documentation . . . 330
27.90.2.1 GetCTImageSeriesUIDs . . . 330
27.90.2.2 GetFilenamesFromSeriesUIDs . . . 330
27.90.2.3 GetFrameOfReference . . . 330
27.90.2.4 GetMRImageSeriesUIDs . . . 330
27.90.2.5 GetRTStructSeriesUIDs . . . 331
27.90.2.6 GetSeriesUIDsBySOPClassUID . . . 331
27.90.2.7 GetSOPClassUID . . . 331
27.90.2.8 GetStringValueFromTag . . . 331
27.90.2.9 LoadImageFromFiles . . . 331
27.90.2.10 RetrieveSOPInstanceUIDFromIndex . . . 331
27.90.2.11 RetrieveSOPInstanceUIDFromZPosition . . . 331
27.91 gdcm::DummyValueGenerator Class Reference . . . 331
27.91.1 Detailed Description . . . 331
27.91.2 Member Function Documentation . . . 331
27.91.2.1 Generate . . . 332
27.92 gdcm::Dumper Class Reference . . . 332
27.92.1 Detailed Description . . . 333
27.92.2 Constructor & Destructor Documentation . . . 333

27.92.2.1 Dumper . . . 333
27.92.2.2 ∼Dumper . . . 333
27.93gdcm::Element< TVR, TVM > Class Template Reference . . . 334
27.93.1 Detailed Description . . . 335
27.93.2 Member Typedef Documentation . . . 336
27.93.2.1 Type . . . 336
27.93.3 Member Function Documentation . . . 336
27.93.3.1 GetAsDataElement . . . 336
27.93.3.2 GetLength . . . 336
27.93.3.3 GetValue . . . 336
27.93.3.4 GetValue . . . 336
27.93.3.5 GetValues . . . 336
27.93.3.6 GetVM . . . 336
27.93.3.7 GetVR . . . 336
27.93.3.8 operator[] . . . 336
27.93.3.9 Print . . . 336
27.93.3.10 Read . . . 336
27.93.3.11 Set . . . 336
27.93.3.12 SetFromDataElement . . . 336
27.93.3.13 SetNoSwap . . . 336
27.93.3.14 SetValue . . . 336
27.93.3.15 Write . . . 336
27.93.4 Member Data Documentation . . . 336
27.93.4.1 Internal . . . 336
27.94gdcm::Element< TVR, VM::VM1_2 > Class Template Reference . . . 337
27.94.1 Member Typedef Documentation . . . 338
27.94.1.1 Parent . . . 338
27.94.2 Member Function Documentation . . . 338
27.94.2.1 SetLength . . . 338
27.95gdcm::Element< TVR, VM::VM1_n > Class Template Reference . . . 338
27.95.1 Member Typedef Documentation . . . 339
27.95.1.1 Type . . . 339
27.95.2 Constructor & Destructor Documentation . . . 339
27.95.2.1 Element . . . 339
27.95.2.2 ∼Element . . . 339
27.95.2.3 Element . . . 339
27.95.3 Member Function Documentation . . . 339

27.95.3.1 GetAsDataElement . . . 339
27.95.3.2 GetLength . . . 339
27.95.3.3 GetValue . . . 339
27.95.3.4 GetValue . . . 339
27.95.3.5 GetVM . . . 339
27.95.3.6 GetVR . . . 340
27.95.3.7 operator= . . . 340
27.95.3.8 operator[] . . . 340
27.95.3.9 Print . . . 340
27.95.3.10 Read . . . 340
27.95.3.11 Set . . . 340
27.95.3.12 SetArray . . . 340
27.95.3.13 SetFromDataElement . . . 340
27.95.3.14 SetLength . . . 340
27.95.3.15 SetNoSwap . . . 340
27.95.3.16 SetValue . . . 340
27.95.3.17 Write . . . 340
27.95.3.18 WriteASCII . . . 340
27.96 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference . . . 340
27.96.1 Member Typedef Documentation . . . 342
27.96.1.1 Parent . . . 342
27.96.2 Member Function Documentation . . . 342
27.96.2.1 SetLength . . . 342
27.97 gdcm::Element< TVR, VM::VM2_n > Class Template Reference . . . 342
27.97.1 Member Typedef Documentation . . . 343
27.97.1.1 Parent . . . 343
27.97.2 Member Function Documentation . . . 343
27.97.2.1 SetLength . . . 343
27.98 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference . . . 343
27.98.1 Member Typedef Documentation . . . 345
27.98.1.1 Parent . . . 345
27.98.2 Member Function Documentation . . . 345
27.98.2.1 SetLength . . . 345
27.99 gdcm::Element< TVR, VM::VM3_n > Class Template Reference . . . 345
27.99.1 Member Typedef Documentation . . . 346
27.99.1.1 Parent . . . 346
27.99.2 Member Function Documentation . . . 346

27.99.2.1 SetLength . . . 346
27.100 gdcm::Element< VR::AS, VM::VM5 > Class Template Reference . . . 346
27.100.1 Member Function Documentation . . . 347
27.100.1.1 GetLength . . . 347
27.100.1.2 Print . . . 347
27.100.2 Member Data Documentation . . . 347
27.100.2.1 Internal . . . 347
27.101 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference . . . 347
27.102 gdcm::Element< VR::OW, VM::VM1 > Class Template Reference . . . 348
27.103 gdcm::ElementDisableCombinations< TVR, TVM > Class Template Reference . . . 350
27.103.1 Detailed Description . . . 350
27.104 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference . . . 351
27.105 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference . . . 351
27.106 gdcm::EncapsulatedDocument Class Reference . . . 351
27.106.1 Detailed Description . . . 351
27.106.2 Constructor & Destructor Documentation . . . 351
27.106.2.1 EncapsulatedDocument . . . 351
27.107 gdcm::EncodingImplementation< T > Singleton Reference . . . 352
27.107.1 Detailed Description . . . 352
27.108 gdcm::EncodingImplementation< VR::VRASCII > Class Template Reference . . . 352
27.108.1 Member Function Documentation . . . 352
27.108.1.1 Read . . . 352
27.108.1.2 ReadComputeLength . . . 353
27.108.1.3 ReadNoSwap . . . 353
27.108.1.4 Write . . . 353
27.108.1.5 Write . . . 353
27.108.1.6 Write . . . 353
27.109 gdcm::EncodingImplementation< VR::VRBINARY > Class Template Reference . . . 353
27.109.1 Member Function Documentation . . . 353
27.109.1.1 Read . . . 353
27.109.1.2 ReadComputeLength . . . 354
27.109.1.3 ReadNoSwap . . . 354
27.109.1.4 Write . . . 354
27.110 gdcm::EndEvent Class Reference . . . 354
27.111 gdcm::EnumeratedValues Class Reference . . . 355
27.111.1 Detailed Description . . . 355
27.111.2 Constructor & Destructor Documentation . . . 356

27.111.2.1 EnumeratedValues . . . 356
27.112 gdcm::Event Class Reference . . . 356
27.112.1 Detailed Description . . . 358
27.112.2 Constructor & Destructor Documentation . . . 358
27.112.2.1 Event . . . 358
27.112.2.2 Event . . . 358
27.112.2.3 ~Event . . . 358
27.112.3 Member Function Documentation . . . 358
27.112.3.1 CheckEvent . . . 358
27.112.3.2 GetEventName . . . 358
27.112.3.3 MakeObject . . . 358
27.112.3.4 Print . . . 358
27.113 gdcm::Exception Class Reference . . . 359
27.113.1 Detailed Description . . . 360
27.113.2 Constructor & Destructor Documentation . . . 360
27.113.2.1 Exception . . . 360
27.113.2.2 ~Exception . . . 360
27.113.3 Member Function Documentation . . . 360
27.113.3.1 GetDescription . . . 360
27.113.3.2 what . . . 360
27.114 gdcm::ExitEvent Class Reference . . . 360
27.115 gdcm::ExplicitDataElement Class Reference . . . 362
27.115.1 Detailed Description . . . 363
27.115.2 Member Function Documentation . . . 363
27.115.2.1 GetLength . . . 363
27.115.2.2 Read . . . 363
27.115.2.3 ReadPreValue . . . 363
27.115.2.4 ReadValue . . . 363
27.115.2.5 ReadWithLength . . . 363
27.115.2.6 Write . . . 363
27.116 gdcm::ExplicitImplicitDataElement Class Reference . . . 363
27.116.1 Detailed Description . . . 365
27.116.2 Member Function Documentation . . . 365
27.116.2.1 GetLength . . . 365
27.116.2.2 Read . . . 365
27.116.2.3 ReadPreValue . . . 365
27.116.2.4 ReadValue . . . 365

27.116.2.5 ReadWithLength . . . 365
27.117 gdcm::Fiducials Class Reference . . . 365
27.117.1 Detailed Description . . . 365
27.117.2 Constructor & Destructor Documentation . . . 366
27.117.2.1 Fiducials . . . 366
27.118 gdcm::File Class Reference . . . 366
27.118.1 Detailed Description . . . 367
27.118.2 Constructor & Destructor Documentation . . . 368
27.118.2.1 File . . . 368
27.118.3 Member Function Documentation . . . 368
27.118.3.1 GetDataSet . . . 368
27.118.3.2 GetDataSet . . . 368
27.118.3.3 GetHeader . . . 368
27.118.3.4 GetHeader . . . 368
27.118.3.5 Read . . . 368
27.118.3.6 SetDataSet . . . 368
27.118.3.7 SetHeader . . . 369
27.118.3.8 Write . . . 369
27.118.4 Friends And Related Function Documentation . . . 369
27.118.4.1 operator<< . . . 369
27.119 gdcm::FileAnonymizer Class Reference . . . 369
27.119.1 Detailed Description . . . 370
27.119.2 Constructor & Destructor Documentation . . . 371
27.119.2.1 FileAnonymizer . . . 371
27.119.2.2 ~FileAnonymizer . . . 371
27.119.3 Member Function Documentation . . . 371
27.119.3.1 Empty . . . 371
27.119.3.2 Remove . . . 371
27.119.3.3 Replace . . . 371
27.119.3.4 Replace . . . 371
27.119.3.5 SetInputFileName . . . 371
27.119.3.6 SetOutputFileName . . . 372
27.119.3.7 Write . . . 372
27.120 gdcm::FileChangeTransferSyntax Class Reference . . . 372
27.120.1 Detailed Description . . . 373
27.120.2 Constructor & Destructor Documentation . . . 374
27.120.2.1 FileChangeTransferSyntax . . . 374

27.120.2.2 ~FileChangeTransferSyntax . . . 374
27.120.3 Member Function Documentation . . . 374
27.120.3.1 Change . . . 374
27.120.3.2 GetCodec . . . 374
27.120.3.3 New . . . 374
27.120.3.4 SetInputFileName . . . 374
27.120.3.5 SetOutputFileName . . . 374
27.120.3.6 SetTransferSyntax . . . 374
27.121 gdcm::FileDerivation Class Reference . . . 374
27.121.1 Detailed Description . . . 375
27.121.2 Constructor & Destructor Documentation . . . 375
27.121.2.1 FileDerivation . . . 375
27.121.2.2 ~FileDerivation . . . 375
27.121.3 Member Function Documentation . . . 375
27.121.3.1 AddDerivationDescription . . . 375
27.121.3.2 AddPurposeOfReferenceCodeSequence . . . 375
27.121.3.3 AddReference . . . 376
27.121.3.4 AddSourceImageSequence . . . 376
27.121.3.5 Derive . . . 376
27.121.3.6 GetFile . . . 376
27.121.3.7 GetFile . . . 376
27.121.3.8 SetDerivationCodeSequenceCodeValue . . . 376
27.121.3.9 SetDerivationDescription . . . 376
27.121.3.10 SetFile . . . 376
27.121.3.11 SetPurposeOfReferenceCodeSequenceCodeValue . . . 377
27.122 gdcm::FileExplicitFilter Class Reference . . . 377
27.122.1 Detailed Description . . . 377
27.122.2 Constructor & Destructor Documentation . . . 378
27.122.2.1 FileExplicitFilter . . . 378
27.122.2.2 ~FileExplicitFilter . . . 378
27.122.3 Member Function Documentation . . . 378
27.122.3.1 Change . . . 378
27.122.3.2 ChangeFMI . . . 378
27.122.3.3 GetFile . . . 378
27.122.3.4 ProcessDataSet . . . 378
27.122.3.5 SetChangePrivateTags . . . 378
27.122.3.6 SetFile . . . 378

27.122.3.7 SetRecomputeItemLength . . . 378
27.122.3.8 SetRecomputeSequenceLength . . . 379
27.122.3.9 SetUseVRUN . . . 379
27.123 gdcm::FileMetaInformation Class Reference . . . 379
27.123.1 Detailed Description . . . 381
27.123.2 Constructor & Destructor Documentation . . . 381
27.123.2.1 FileMetaInformation . . . 381
27.123.2.2 FileMetaInformation . . . 381
27.123.3 Member Function Documentation . . . 381
27.123.3.1 AppendImplementationClassUID . . . 381
27.123.3.2 ComputeDataSetMediaStorageSOPClass . . . 381
27.123.3.3 ComputeDataSetTransferSyntax . . . 381
27.123.3.4 Default . . . 381
27.123.3.5 FillFromDataSet . . . 381
27.123.3.6 GetDataSetTransferSyntax . . . 382
27.123.3.7 GetFileMetaInformationVersion . . . 382
27.123.3.8 GetFullLength . . . 382
27.123.3.9 GetGDCMImplementationClassUID . . . 382
27.123.3.10 GetGDCMImplementationVersionName . . . 382
27.123.3.11 GetGDCMSourceApplicationEntityTitle . . . 382
27.123.3.12 GetImplementationClassUID . . . 382
27.123.3.13 GetImplementationVersionName . . . 382
27.123.3.14 GetMediaStorage . . . 382
27.123.3.15 GetMediaStorageAsString . . . 382
27.123.3.16 GetMetaInformationTS . . . 382
27.123.3.17 GetPreamble . . . 382
27.123.3.18 SetPreamble . . . 382
27.123.3.19 GetSourceApplicationEntityTitle . . . 382
27.123.3.20 Insert . . . 382
27.123.3.21 IsValid . . . 382
27.123.3.22 Read . . . 382
27.123.3.23 ReadCompat . . . 383
27.123.3.24 ReadCompatInternal . . . 383
27.123.3.25 Replace . . . 383
27.123.3.26 SetDataSetTransferSyntax . . . 383
27.123.3.27 SetImplementationClassUID . . . 383
27.123.3.28 SetImplementationVersionName . . . 383

27.123.3.2 SetPreamble . . . 383
27.123.3.3 SetSourceApplicationEntityTitle . . . 383
27.123.3.4 Write . . . 383
27.123.4 Friends And Related Function Documentation . . . 383
27.123.4.1 operator<< . . . 383
27.123.5 Member Data Documentation . . . 383
27.123.5.1 DataSetMS . . . 383
27.123.5.2 DataSetTS . . . 384
27.123.5.3 MetaInformationTS . . . 384
27.124 gdcm::Filename Class Reference . . . 384
27.124.1 Detailed Description . . . 385
27.124.2 Constructor & Destructor Documentation . . . 385
27.124.2.1 Filename . . . 385
27.124.3 Member Function Documentation . . . 385
27.124.3.1 EndWith . . . 385
27.124.3.2 GetExtension . . . 385
27.124.3.3 GetFileName . . . 385
27.124.3.4 GetName . . . 385
27.124.3.5 GetPath . . . 385
27.124.3.6 IsEmpty . . . 385
27.124.3.7 IsIdentical . . . 385
27.124.3.8 Join . . . 385
27.124.3.9 operator const char * . . . 386
27.124.3.10 ToUnixSlashes . . . 386
27.124.3.11 ToWindowsSlashes . . . 386
27.125 gdcm::FileNameEvent Class Reference . . . 386
27.125.1 Detailed Description . . . 387
27.125.2 Member Typedef Documentation . . . 388
27.125.2.1 Self . . . 388
27.125.2.2 Superclass . . . 388
27.125.3 Constructor & Destructor Documentation . . . 388
27.125.3.1 FileNameEvent . . . 388
27.125.3.2 ~FileNameEvent . . . 388
27.125.3.3 FileNameEvent . . . 388
27.125.4 Member Function Documentation . . . 388
27.125.4.1 CheckEvent . . . 388
27.125.4.2 GetEventName . . . 388

27.125.4.3 GetFileName . . . 388
27.125.4.4 MakeObject . . . 388
27.125.4.5 SetFileName . . . 388
27.126 gdcm::FilenameGenerator Class Reference . . . 388
27.126.1 Detailed Description . . . 389
27.126.2 Member Typedef Documentation . . . 389
27.126.2.1 FilenamesType . . . 389
27.126.2.2 FilenameType . . . 389
27.126.2.3 SizeType . . . 389
27.126.3 Constructor & Destructor Documentation . . . 389
27.126.3.1 FilenameGenerator . . . 390
27.126.3.2 ~FilenameGenerator . . . 390
27.126.4 Member Function Documentation . . . 390
27.126.4.1 Generate . . . 390
27.126.4.2 GetFilename . . . 390
27.126.4.3 GetFilenames . . . 390
27.126.4.4 GetNumberOfFilenames . . . 390
27.126.4.5 GetPattern . . . 390
27.126.4.6 GetPrefix . . . 390
27.126.4.7 SetNumberOfFilenames . . . 390
27.126.4.8 SetPattern . . . 390
27.126.4.9 SetPrefix . . . 391
27.127 gdcm::FileSet Class Reference . . . 391
27.127.1 Detailed Description . . . 391
27.127.2 Member Typedef Documentation . . . 391
27.127.2.1 FilesType . . . 391
27.127.2.2 FileType . . . 391
27.127.3 Constructor & Destructor Documentation . . . 391
27.127.3.1 FileSet . . . 391
27.127.4 Member Function Documentation . . . 391
27.127.4.1 AddFile . . . 392
27.127.4.2 AddFile . . . 392
27.127.4.3 GetFiles . . . 392
27.127.4.4 SetFiles . . . 392
27.127.5 Friends And Related Function Documentation . . . 392
27.127.5.1 operator<< . . . 392
27.128 gdcm::FileStreamer Class Reference . . . 392

27.128.1 Detailed Description . . . 394
27.128.2 Constructor & Destructor Documentation . . . 394
27.128.2.1 FileStreamer . . . 394
27.128.2.2 ~FileStreamer . . . 394
27.128.3 Member Function Documentation . . . 394
27.128.3.1 AppendToDataElement . . . 394
27.128.3.2 AppendToGroupDataElement . . . 394
27.128.3.3 CheckDataElement . . . 394
27.128.3.4 CheckTemplateFileName . . . 394
27.128.3.5 New . . . 394
27.128.3.6 ReserveDataElement . . . 395
27.128.3.7 ReserveGroupDataElement . . . 395
27.128.3.8 SetOutputFileName . . . 395
27.128.3.9 SetTemplateFileName . . . 395
27.128.3.10 StartDataElement . . . 395
27.128.3.11 StartGroupDataElement . . . 395
27.128.3.12 StopDataElement . . . 395
27.128.3.13 StopGroupDataElement . . . 395
27.129 gdcm::FileWithName Class Reference . . . 396
27.129.1 Detailed Description . . . 397
27.129.2 Constructor & Destructor Documentation . . . 397
27.129.2.1 FileWithName . . . 397
27.129.3 Member Data Documentation . . . 397
27.129.3.1 filename . . . 397
27.130 gdcm::FindPatientRootQuery Class Reference . . . 397
27.130.1 Detailed Description . . . 398
27.130.2 Constructor & Destructor Documentation . . . 398
27.130.2.1 FindPatientRootQuery . . . 398
27.130.3 Member Function Documentation . . . 398
27.130.3.1 GetAbstractSyntaxUID . . . 398
27.130.3.2 GetTagListByLevel . . . 399
27.130.3.3 InitializeDataSet . . . 399
27.130.3.4 ValidateQuery . . . 399
27.130.4 Friends And Related Function Documentation . . . 399
27.130.4.1 QueryFactory . . . 399
27.131 gdcm::FindStudyRootQuery Class Reference . . . 399
27.131.1 Detailed Description . . . 401

27.131.2 Constructor & Destructor Documentation . . . 401
27.131.2.1 FindStudyRootQuery . . . 401
27.131.3 Member Function Documentation . . . 401
27.131.3.1 GetAbstractSyntaxUID . . . 401
27.131.3.2 GetTagListByLevel . . . 401
27.131.3.3 InitializeDataSet . . . 401
27.131.3.4 ValidateQuery . . . 401
27.131.4 Friends And Related Function Documentation . . . 401
27.131.4.1 QueryFactory . . . 401
27.132 gdcm::Fragment Class Reference . . . 401
27.132.1 Detailed Description . . . 403
27.132.2 Constructor & Destructor Documentation . . . 403
27.132.2.1 Fragment . . . 403
27.132.3 Member Function Documentation . . . 403
27.132.3.1 ComputeLength . . . 403
27.132.3.2 GetLength . . . 403
27.132.3.3 Read . . . 403
27.132.3.4 ReadBacktrack . . . 403
27.132.3.5 ReadPreValue . . . 403
27.132.3.6 ReadValue . . . 403
27.132.3.7 Write . . . 404
27.132.4 Friends And Related Function Documentation . . . 404
27.132.4.1 operator<< . . . 404
27.133 gdcm::Global Class Reference . . . 404
27.133.1 Detailed Description . . . 405
27.133.2 Constructor & Destructor Documentation . . . 405
27.133.2.1 Global . . . 405
27.133.2.2 ~Global . . . 405
27.133.3 Member Function Documentation . . . 405
27.133.3.1 Append . . . 405
27.133.3.2 GetDefs . . . 405
27.133.3.3 GetDicts . . . 405
27.133.3.4 GetDicts . . . 405
27.133.3.5 GetInstance . . . 405
27.133.3.6 LoadResourcesFiles . . . 406
27.133.3.7 Locate . . . 406
27.133.3.8 Prepend . . . 406

27.133.4 Friends And Related Function Documentation . . . 406
27.133.4.1 operator<< . . . 406
27.134 gdcm::GroupDict Class Reference . . . 406
27.134.1 Detailed Description . . . 407
27.134.2 Member Typedef Documentation . . . 407
27.134.2.1 GroupStringVector . . . 407
27.134.3 Constructor & Destructor Documentation . . . 407
27.134.3.1 GroupDict . . . 407
27.134.3.2 ~GroupDict . . . 407
27.134.4 Member Function Documentation . . . 407
27.134.4.1 Add . . . 407
27.134.4.2 GetAbbreviation . . . 407
27.134.4.3 GetName . . . 407
27.134.4.4 Insert . . . 408
27.134.4.5 Size . . . 408
27.134.5 Friends And Related Function Documentation . . . 408
27.134.5.1 operator<< . . . 408
27.135 gdcm::IconImageFilter Class Reference . . . 408
27.135.1 Detailed Description . . . 408
27.135.2 Constructor & Destructor Documentation . . . 409
27.135.2.1 IconImageFilter . . . 409
27.135.2.2 ~IconImageFilter . . . 409
27.135.3 Member Function Documentation . . . 409
27.135.3.1 Extract . . . 409
27.135.3.2 ExtractIconImages . . . 409
27.135.3.3 ExtractVeproIconImages . . . 409
27.135.3.4 GetFile . . . 409
27.135.3.5 GetFile . . . 409
27.135.3.6 GetIconImage . . . 410
27.135.3.7 GetNumberOfIconImages . . . 410
27.135.3.8 SetFile . . . 410
27.136 gdcm::IconImageGenerator Class Reference . . . 410
27.136.1 Detailed Description . . . 411
27.136.2 Constructor & Destructor Documentation . . . 411
27.136.2.1 IconImageGenerator . . . 411
27.136.2.2 ~IconImageGenerator . . . 411
27.136.3 Member Function Documentation . . . 411

27.136.3.1 AutoPixelMinMax . . . 411
27.136.3.2 ConvertRGBToPaletteColor . . . 411
27.136.3.3 Generate . . . 411
27.136.3.4 GetIconImage . . . 412
27.136.3.5 GetPixmap . . . 412
27.136.3.6 GetPixmap . . . 412
27.136.3.7 SetOutputDimensions . . . 412
27.136.3.8 SetOutsideValuePixel . . . 412
27.136.3.9 SetPixelMinMax . . . 412
27.136.3.10 SetPixmap . . . 412
27.137 gdcm::ignore_char Struct Reference . . . 412
27.137.1 Constructor & Destructor Documentation . . . 413
27.137.1.1 ignore_char . . . 413
27.137.2 Member Data Documentation . . . 413
27.137.2.1 m_char . . . 413
27.138 gdcm::Image Class Reference . . . 413
27.138.1 Detailed Description . . . 414
27.138.2 Constructor & Destructor Documentation . . . 415
27.138.2.1 Image . . . 415
27.138.2.2 ~Image . . . 415
27.138.3 Member Function Documentation . . . 415
27.138.3.1 GetDirectionCosines . . . 415
27.138.3.2 GetDirectionCosines . . . 415
27.138.3.3 GetIntercept . . . 415
27.138.3.4 GetOrigin . . . 415
27.138.3.5 GetOrigin . . . 416
27.138.3.6 GetSlope . . . 416
27.138.3.7 GetSpacing . . . 416
27.138.3.8 GetSpacing . . . 416
27.138.3.9 Print . . . 416
27.138.3.10 SetDirectionCosines . . . 416
27.138.3.11 SetDirectionCosines . . . 416
27.138.3.12 SetDirectionCosines . . . 416
27.138.3.13 SetIntercept . . . 416
27.138.3.14 SetOrigin . . . 416
27.138.3.15 SetOrigin . . . 416
27.138.3.16 SetOrigin . . . 416

27.138.3.17SetSlope . . . 416
27.138.3.18SetSpacing . . . 416
27.138.3.19SetSpacing . . . 417
27.139gdcm::ImageApplyLookupTable Class Reference . . . 417
27.139.1Detailed Description . . . 419
27.139.2Constructor & Destructor Documentation . . . 419
27.139.2.1ImageApplyLookupTable . . . 419
27.139.2.2~ImageApplyLookupTable . . . 419
27.139.3Member Function Documentation . . . 419
27.139.3.1Apply . . . 419
27.140gdcm::ImageChangePhotometricInterpretation Class Reference . . . 419
27.140.1Detailed Description . . . 421
27.140.2Constructor & Destructor Documentation . . . 421
27.140.2.1ImageChangePhotometricInterpretation . . . 421
27.140.2.2~ImageChangePhotometricInterpretation . . . 421
27.140.3Member Function Documentation . . . 421
27.140.3.1Change . . . 421
27.140.3.2ChangeMonochrome . . . 421
27.140.3.3GetPhotometricInterpretation . . . 421
27.140.3.4RGB2YBR . . . 421
27.140.3.5SetPhotometricInterpretation . . . 421
27.140.3.6YBR2RGB . . . 422
27.141gdcm::ImageChangePlanarConfiguration Class Reference . . . 422
27.141.1Detailed Description . . . 424
27.141.2Constructor & Destructor Documentation . . . 424
27.141.2.1ImageChangePlanarConfiguration . . . 424
27.141.2.2~ImageChangePlanarConfiguration . . . 424
27.141.3Member Function Documentation . . . 424
27.141.3.1Change . . . 424
27.141.3.2GetPlanarConfiguration . . . 424
27.141.3.3RGBPixelsToRGBPlanes . . . 424
27.141.3.4RGBPlanesToRGBPixels . . . 424
27.141.3.5SetPlanarConfiguration . . . 424
27.142gdcm::ImageChangeTransferSyntax Class Reference . . . 425
27.142.1Detailed Description . . . 427
27.142.2Constructor & Destructor Documentation . . . 427
27.142.2.1ImageChangeTransferSyntax . . . 427

27.142.2.2 ~ImageChangeTransferSyntax . . . 427
27.142.3 Member Function Documentation . . . 427
27.142.3.1 Change . . . 427
27.142.3.2 GetTransferSyntax . . . 427
27.142.3.3 SetCompressIconImage . . . 428
27.142.3.4 SetForce . . . 428
27.142.3.5 SetTransferSyntax . . . 428
27.142.3.6 SetUserCodec . . . 428
27.142.3.7 TryJPEG2000Codec . . . 428
27.142.3.8 TryJPEGCodec . . . 428
27.142.3.9 TryJPEGLSCodec . . . 428
27.142.3.10 TryRAWCodec . . . 428
27.142.3.11 TryRLECodec . . . 428
27.143 gdcm::ImageCodec Class Reference . . . 429
27.143.1 Detailed Description . . . 431
27.143.2 Member Typedef Documentation . . . 431
27.143.2.1 LUTPtr . . . 431
27.143.3 Constructor & Destructor Documentation . . . 431
27.143.3.1 ImageCodec . . . 431
27.143.3.2 ~ImageCodec . . . 431
27.143.4 Member Function Documentation . . . 431
27.143.4.1 AppendFrameEncode . . . 431
27.143.4.2 AppendRowEncode . . . 431
27.143.4.3 CanCode . . . 432
27.143.4.4 CanDecode . . . 432
27.143.4.5 Clone . . . 432
27.143.4.6 Decode . . . 432
27.143.4.7 DecodeByStreams . . . 432
27.143.4.8 DoByteSwap . . . 432
27.143.4.9 DoInvertMonochrome . . . 432
27.143.4.10 DoOverlayCleanup . . . 432
27.143.4.11 DoPaddedCompositePixelCode . . . 432
27.143.4.12 DoPlanarConfiguration . . . 432
27.143.4.13 DoSimpleCopy . . . 432
27.143.4.14 DoYBR . . . 432
27.143.4.15 GetDimensions . . . 433
27.143.4.16 GetHeaderInfo . . . 433

27.143.4.17 GetLossyFlag . . . 433
27.143.4.18 GetLUT . . . 433
27.143.4.19 GetNeedByteSwap . . . 433
27.143.4.20 GetNumberOfDimensions . . . 433
27.143.4.21 GetPhotometricInterpretation . . . 433
27.143.4.22 GetPixelFormat . . . 433
27.143.4.23 GetPixelFormat . . . 433
27.143.4.24 GetPlanarConfiguration . . . 433
27.143.4.25 IsFrameEncoder . . . 433
27.143.4.26 IsLossy . . . 433
27.143.4.27 IsRowEncoder . . . 433
27.143.4.28 IsValid . . . 433
27.143.4.29 SetDimensions . . . 433
27.143.4.30 SetDimensions . . . 434
27.143.4.31 SetLossyFlag . . . 434
27.143.4.32 SetLUT . . . 434
27.143.4.33 SetNeedByteSwap . . . 434
27.143.4.34 SetNeedOverlayCleanup . . . 434
27.143.4.35 SetNumberOfDimensions . . . 434
27.143.4.36 SetPhotometricInterpretation . . . 434
27.143.4.37 SetPixelFormat . . . 434
27.143.4.38 SetPlanarConfiguration . . . 434
27.143.4.39 StartEncode . . . 434
27.143.4.40 StopEncode . . . 434
27.143.5 Friends And Related Function Documentation . . . 434
27.143.5.1 FileChangeTransferSyntax . . . 434
27.143.5.2 ImageChangePhotometricInterpretation . . . 435
27.143.6 Member Data Documentation . . . 435
27.143.6.1 Dimensions . . . 435
27.143.6.2 LossyFlag . . . 435
27.143.6.3 LUT . . . 435
27.143.6.4 NeedByteSwap . . . 435
27.143.6.5 NeedOverlayCleanup . . . 435
27.143.6.6 NumberOfDimensions . . . 435
27.143.6.7 PF . . . 435
27.143.6.8 PI . . . 435
27.143.6.9 PlanarConfiguration . . . 435

27.143.6.18 RequestPaddedCompositePixelCode . . . 435
27.143.6.19 RequestPlanarConfiguration . . . 435
27.144 gdcm::ImageConverter Class Reference . . . 435
27.144.1 Detailed Description . . . 435
27.144.2 Constructor & Destructor Documentation . . . 436
27.144.2.1 ImageConverter . . . 436
27.144.2.2 ~ImageConverter . . . 436
27.144.3 Member Function Documentation . . . 436
27.144.3.1 Convert . . . 436
27.144.3.2 GetOuput . . . 436
27.144.3.3 SetInput . . . 436
27.145 gdcm::ImageFragmentSplitter Class Reference . . . 436
27.145.1 Detailed Description . . . 439
27.145.2 Constructor & Destructor Documentation . . . 439
27.145.2.1 ImageFragmentSplitter . . . 439
27.145.2.2 ~ImageFragmentSplitter . . . 439
27.145.3 Member Function Documentation . . . 439
27.145.3.1 GetFragmentSizeMax . . . 439
27.145.3.2 SetForce . . . 439
27.145.3.3 SetFragmentSizeMax . . . 439
27.145.3.4 Split . . . 439
27.146 gdcm::ImageHelper Class Reference . . . 439
27.146.1 Detailed Description . . . 440
27.146.2 Member Function Documentation . . . 440
27.146.2.1 ComputeMediaStorageFromModality . . . 440
27.146.2.2 ComputeSpacingFromImagePositionPatient . . . 441
27.146.2.3 GetDimensionsValue . . . 441
27.146.2.4 GetDirectionCosinesFromDataSet . . . 441
27.146.2.5 GetDirectionCosinesValue . . . 441
27.146.2.6 GetForcePixelSpacing . . . 441
27.146.2.7 GetForceRescaleInterceptSlope . . . 441
27.146.2.8 GetLUT . . . 441
27.146.2.9 GetOriginValue . . . 441
27.146.2.10 GetPhotometricInterpretationValue . . . 441
27.146.2.11 GetPixelFormatValue . . . 441
27.146.2.12 GetPlanarConfigurationValue . . . 441
27.146.2.13 GetPointerFromElement . . . 441

27.146.2.14GetRescaleInterceptSlopeValue . . . 441
27.146.2.15GetSpacingTagFromMediaStorage . . . 442
27.146.2.16GetSpacingValue . . . 442
27.146.2.17GetZSpacingTagFromMediaStorage . . . 442
27.146.2.18SetDimensionsValue . . . 442
27.146.2.19SetDirectionCosinesValue . . . 442
27.146.2.20SetForcePixelSpacing . . . 442
27.146.2.21SetForceRescaleInterceptSlope . . . 442
27.146.2.22SetOriginValue . . . 442
27.146.2.23SetRescaleInterceptSlopeValue . . . 442
27.146.2.24SetSpacingValue . . . 442
27.147gdcm::ImageReader Class Reference . . . 443
27.147.1Detailed Description . . . 445
27.147.2Constructor & Destructor Documentation . . . 445
27.147.2.1ImageReader . . . 445
27.147.2.2~ImageReader . . . 445
27.147.3Member Function Documentation . . . 445
27.147.3.1GetImage . . . 445
27.147.3.2GetImage . . . 445
27.147.3.3Read . . . 445
27.147.3.4ReadACRNEMAImage . . . 446
27.147.3.5ReadImage . . . 446
27.148gdcm::ImageRegionReader Class Reference . . . 446
27.148.1Detailed Description . . . 448
27.148.2Constructor & Destructor Documentation . . . 448
27.148.2.1ImageRegionReader . . . 448
27.148.2.2~ImageRegionReader . . . 448
27.148.3Member Function Documentation . . . 448
27.148.3.1ComputeBufferLength . . . 448
27.148.3.2GetRegion . . . 448
27.148.3.3Read . . . 448
27.148.3.4ReadInformation . . . 448
27.148.3.5ReadIntoBuffer . . . 449
27.148.3.6SetRegion . . . 449
27.149gdcm::ImageToImageFilter Class Reference . . . 449
27.149.1Detailed Description . . . 450
27.149.2Constructor & Destructor Documentation . . . 450

27.149.2.1 ImageToImageFilter . . . 451
27.149.2.2 ~ImageToImageFilter . . . 451
27.149.3 Member Function Documentation . . . 451
27.149.3.1 GetInput . . . 451
27.149.3.2 GetOutput . . . 451
27.150 gdcm::ImageWriter Class Reference . . . 451
27.150.1 Detailed Description . . . 453
27.150.2 Constructor & Destructor Documentation . . . 453
27.150.2.1 ImageWriter . . . 453
27.150.2.2 ~ImageWriter . . . 453
27.150.3 Member Function Documentation . . . 453
27.150.3.1 GetImage . . . 453
27.150.3.2 GetImage . . . 453
27.150.3.3 Write . . . 453
27.151 gdcm::network::ImplementationClassUIDSub Class Reference . . . 454
27.151.1 Detailed Description . . . 454
27.151.2 Constructor & Destructor Documentation . . . 454
27.151.2.1 ImplementationClassUIDSub . . . 454
27.151.3 Member Function Documentation . . . 454
27.151.3.1 Print . . . 454
27.151.3.2 Read . . . 454
27.151.3.3 Size . . . 454
27.151.3.4 Write . . . 454
27.152 gdcm::network::ImplementationUIDSub Class Reference . . . 454
27.152.1 Detailed Description . . . 455
27.152.2 Constructor & Destructor Documentation . . . 455
27.152.2.1 ImplementationUIDSub . . . 455
27.152.3 Member Function Documentation . . . 455
27.152.3.1 Write . . . 455
27.153 gdcm::network::ImplementationVersionNameSub Class Reference . . . 455
27.153.1 Detailed Description . . . 455
27.153.2 Constructor & Destructor Documentation . . . 455
27.153.2.1 ImplementationVersionNameSub . . . 455
27.153.3 Member Function Documentation . . . 455
27.153.3.1 Print . . . 455
27.153.3.2 Read . . . 455
27.153.3.3 Size . . . 455

27.153.3.4 Write . . . 456
27.154 gdcm::ImplicitDataElement Class Reference . . . 456
27.154.1 Detailed Description . . . 457
27.154.2 Member Function Documentation . . . 457
27.154.2.1 GetLength . . . 457
27.154.2.2 Read . . . 457
27.154.2.3 ReadPreValue . . . 457
27.154.2.4 ReadValue . . . 457
27.154.2.5 ReadValueWithLength . . . 457
27.154.2.6 ReadWithLength . . . 457
27.154.2.7 Write . . . 457
27.155 gdcm::InitializeEvent Class Reference . . . 458
27.156 gdcm::IOD Class Reference . . . 459
27.156.1 Detailed Description . . . 459
27.156.2 Member Typedef Documentation . . . 459
27.156.2.1 MapIODEntry . . . 459
27.156.2.2 SizeType . . . 459
27.156.3 Constructor & Destructor Documentation . . . 459
27.156.3.1 IOD . . . 459
27.156.4 Member Function Documentation . . . 460
27.156.4.1 AddIODEntry . . . 460
27.156.4.2 Clear . . . 460
27.156.4.3 GetIODEntry . . . 460
27.156.4.4 GetNumberOfIODs . . . 460
27.156.4.5 GetTypeFromTag . . . 460
27.156.5 Friends And Related Function Documentation . . . 460
27.156.5.1 operator<< . . . 460
27.157 gdcm::IODEntry Class Reference . . . 460
27.157.1 Detailed Description . . . 461
27.157.2 Constructor & Destructor Documentation . . . 461
27.157.2.1 IODEntry . . . 461
27.157.3 Member Function Documentation . . . 461
27.157.3.1 GetIE . . . 461
27.157.3.2 GetName . . . 461
27.157.3.3 GetRef . . . 461
27.157.3.4 GetUsage . . . 462
27.157.3.5 GetUsageType . . . 462

27.157.3.6 SetIE . . . 462
27.157.3.7 SetName . . . 462
27.157.3.8 SetRef . . . 462
27.157.3.9 SetUsage . . . 462
27.157.4 Friends And Related Function Documentation . . . 462
27.157.4.1 operator<< . . . 462
27.158 gdcm::IODs Class Reference . . . 462
27.158.1 Detailed Description . . . 463
27.158.2 Member Typedef Documentation . . . 463
27.158.2.1 IODMapType . . . 463
27.158.2.2 IODMapTypeConstIterator . . . 463
27.158.2.3 IODName . . . 463
27.158.3 Constructor & Destructor Documentation . . . 463
27.158.3.1 IODs . . . 463
27.158.4 Member Function Documentation . . . 463
27.158.4.1 AddIOD . . . 463
27.158.4.2 Begin . . . 463
27.158.4.3 Clear . . . 463
27.158.4.4 End . . . 463
27.158.4.5 GetIOD . . . 463
27.158.5 Friends And Related Function Documentation . . . 463
27.158.5.1 operator<< . . . 464
27.159 gdcm::IPPSorter Class Reference . . . 464
27.159.1 Detailed Description . . . 465
27.159.2 Constructor & Destructor Documentation . . . 465
27.159.2.1 IPPSorter . . . 465
27.159.3 Member Function Documentation . . . 465
27.159.3.1 GetDirectionCosinesTolerance . . . 465
27.159.3.2 GetZSpacing . . . 466
27.159.3.3 GetZSpacingTolerance . . . 466
27.159.3.4 SetComputeZSpacing . . . 466
27.159.3.5 SetDirectionCosinesTolerance . . . 466
27.159.3.6 SetDropDuplicatePositions . . . 466
27.159.3.7 SetZSpacingTolerance . . . 466
27.159.3.8 Sort . . . 467
27.159.4 Member Data Documentation . . . 467
27.159.4.1 ComputeZSpacing . . . 467

27.159.4.2 DirCosTolerance . . . 467
27.159.4.3 DropDuplicatePositions . . . 467
27.159.4.4 ZSpacing . . . 467
27.159.4.5 ZTolerance . . . 467
27.160 gdcm::Item Class Reference . . . 467
27.160.1 Detailed Description . . . 469
27.160.2 Constructor & Destructor Documentation . . . 469
27.160.2.1 Item . . . 469
27.160.2.2 Item . . . 469
27.160.3 Member Function Documentation . . . 469
27.160.3.1 Clear . . . 469
27.160.3.2 FindDataElement . . . 469
27.160.3.3 GetDataElement . . . 469
27.160.3.4 GetLength . . . 469
27.160.3.5 GetNestedDataSet . . . 469
27.160.3.6 GetNestedDataSet . . . 470
27.160.3.7 InsertDataElement . . . 470
27.160.3.8 Read . . . 470
27.160.3.9 SetNestedDataSet . . . 470
27.160.3.10 Write . . . 470
27.160.4 Friends And Related Function Documentation . . . 470
27.160.4.1 operator<< . . . 470
27.161 gdcm::IterationEvent Class Reference . . . 470
27.162 gdcm::JPEG12Codec Class Reference . . . 472
27.162.1 Detailed Description . . . 473
27.162.2 Constructor & Destructor Documentation . . . 473
27.162.2.1 JPEG12Codec . . . 473
27.162.2.2 ~JPEG12Codec . . . 473
27.162.3 Member Function Documentation . . . 473
27.162.3.1 DecodeByStreams . . . 473
27.162.3.2 EncodeBuffer . . . 473
27.162.3.3 GetHeaderInfo . . . 473
27.162.3.4 InternalCode . . . 473
27.162.3.5 IsStateSuspension . . . 473
27.163 gdcm::JPEG16Codec Class Reference . . . 474
27.163.1 Detailed Description . . . 475
27.163.2 Constructor & Destructor Documentation . . . 475

27.163.2.1 JPEG16Codec . . . 475
27.163.2.2 ~JPEG16Codec . . . 475
27.163.3 Member Function Documentation . . . 475
27.163.3.1 DecodeByStreams . . . 475
27.163.3.2 EncodeBuffer . . . 475
27.163.3.3 GetHeaderInfo . . . 475
27.163.3.4 InternalCode . . . 475
27.163.3.5 IsStateSuspension . . . 476
27.164 gdcm::JPEG2000Codec Class Reference . . . 476
27.164.1 Detailed Description . . . 477
27.164.2 Constructor & Destructor Documentation . . . 478
27.164.2.1 JPEG2000Codec . . . 478
27.164.2.2 ~JPEG2000Codec . . . 478
27.164.3 Member Function Documentation . . . 478
27.164.3.1 AppendFrameEncode . . . 478
27.164.3.2 AppendRowEncode . . . 478
27.164.3.3 CanCode . . . 478
27.164.3.4 CanDecode . . . 478
27.164.3.5 Clone . . . 478
27.164.3.6 Code . . . 478
27.164.3.7 Decode . . . 478
27.164.3.8 DecodeByStreams . . . 478
27.164.3.9 DecodeExtent . . . 479
27.164.3.10 GetHeaderInfo . . . 479
27.164.3.11 GetQuality . . . 479
27.164.3.12 GetRate . . . 479
27.164.3.13 IsFrameEncoder . . . 479
27.164.3.14 IsRowEncoder . . . 479
27.164.3.15 SetNumberOfResolutions . . . 479
27.164.3.16 SetQuality . . . 479
27.164.3.17 SetRate . . . 479
27.164.3.18 SetReversible . . . 479
27.164.3.19 SetTileSize . . . 479
27.164.3.20 StartEncode . . . 479
27.164.3.21 StopEncode . . . 479
27.164.4 Friends And Related Function Documentation . . . 479
27.164.4.1 Bitmap . . . 479

27.164.4.2 ImageRegionReader . . . 479
27.165 gdcm::JPEG8Codec Class Reference . . . 480
27.165.1 Detailed Description . . . 481
27.165.2 Constructor & Destructor Documentation . . . 481
27.165.2.1 JPEG8Codec . . . 481
27.165.2.2 ~JPEG8Codec . . . 481
27.165.3 Member Function Documentation . . . 481
27.165.3.1 DecodeByStreams . . . 481
27.165.3.2 EncodeBuffer . . . 481
27.165.3.3 GetHeaderInfo . . . 481
27.165.3.4 InternalCode . . . 481
27.165.3.5 IsStateSuspension . . . 481
27.166 gdcm::JPEGCodec Class Reference . . . 482
27.166.1 Detailed Description . . . 484
27.166.2 Constructor & Destructor Documentation . . . 484
27.166.2.1 JPEGCodec . . . 484
27.166.2.2 ~JPEGCodec . . . 484
27.166.3 Member Function Documentation . . . 484
27.166.3.1 AppendFrameEncode . . . 484
27.166.3.2 AppendRowEncode . . . 484
27.166.3.3 CanCode . . . 484
27.166.3.4 CanDecode . . . 484
27.166.3.5 Clone . . . 485
27.166.3.6 Code . . . 485
27.166.3.7 ComputeOffsetTable . . . 485
27.166.3.8 Decode . . . 485
27.166.3.9 DecodeByStreams . . . 485
27.166.3.10 DecodeExtent . . . 485
27.166.3.11 EncodeBuffer . . . 485
27.166.3.12 GetHeaderInfo . . . 485
27.166.3.13 GetLossless . . . 485
27.166.3.14 GetQuality . . . 485
27.166.3.15 IsFrameEncoder . . . 485
27.166.3.16 IsRowEncoder . . . 486
27.166.3.17 IsStateSuspension . . . 486
27.166.3.18 IsValid . . . 486
27.166.3.19 SetBitSample . . . 486

27.166.3.25 SetLossless . . . 486
27.166.3.26 SetPixelFormat . . . 486
27.166.3.27 SetQuality . . . 486
27.166.3.28 StartEncode . . . 486
27.166.3.29 StopEncode . . . 486
27.166.4 Friends And Related Function Documentation . . . 486
27.166.4.1 ImageRegionReader . . . 486
27.166.5 Member Data Documentation . . . 486
27.166.5.1 BitSample . . . 486
27.166.5.2 Quality . . . 486
27.167 gdcm::JPEGLSCodec Class Reference . . . 487
27.167.1 Detailed Description . . . 488
27.167.2 Constructor & Destructor Documentation . . . 489
27.167.2.1 JPEGLSCodec . . . 489
27.167.2.2 ~JPEGLSCodec . . . 489
27.167.3 Member Function Documentation . . . 489
27.167.3.1 AppendFrameEncode . . . 489
27.167.3.2 AppendRowEncode . . . 489
27.167.3.3 CanCode . . . 489
27.167.3.4 CanDecode . . . 489
27.167.3.5 Clone . . . 489
27.167.3.6 Code . . . 489
27.167.3.7 Decode . . . 489
27.167.3.8 Decode . . . 489
27.167.3.9 DecodeExtent . . . 490
27.167.3.10 GetBufferLength . . . 490
27.167.3.11 GetHeaderInfo . . . 490
27.167.3.12 GetLossless . . . 490
27.167.3.13 IsFrameEncoder . . . 490
27.167.3.14 IsRowEncoder . . . 490
27.167.3.15 SetBufferLength . . . 490
27.167.3.16 SetLossless . . . 490
27.167.3.17 SetLossyError . . . 490
27.167.3.18 StartEncode . . . 490
27.167.3.19 StopEncode . . . 490
27.167.4 Friends And Related Function Documentation . . . 490
27.167.4.1 ImageRegionReader . . . 490

27.168 gdcm::JSON Class Reference . . . 490
27.168.1 Detailed Description . . . 491
27.168.2 Constructor & Destructor Documentation . . . 491
27.168.2.1 JSON . . . 491
27.168.2.2 ~JSON . . . 491
27.168.3 Member Function Documentation . . . 491
27.168.3.1 Code . . . 491
27.168.3.2 Decode . . . 491
27.168.3.3 GetPrettyPrint . . . 491
27.168.3.4 PrettyPrintOff . . . 491
27.168.3.5 PrettyPrintOn . . . 491
27.168.3.6 SetPrettyPrint . . . 492
27.169 gdcm::KAKADUCodec Class Reference . . . 492
27.169.1 Detailed Description . . . 493
27.169.2 Constructor & Destructor Documentation . . . 493
27.169.2.1 KAKADUCodec . . . 493
27.169.2.2 ~KAKADUCodec . . . 493
27.169.3 Member Function Documentation . . . 493
27.169.3.1 CanCode . . . 493
27.169.3.2 CanDecode . . . 493
27.169.3.3 Clone . . . 493
27.169.3.4 Code . . . 493
27.169.3.5 Decode . . . 494
27.170 gdcm::LO Class Reference . . . 494
27.170.1 Detailed Description . . . 495
27.170.2 Member Typedef Documentation . . . 495
27.170.2.1 const_iterator . . . 495
27.170.2.2 const_reference . . . 495
27.170.2.3 const_reverse_iterator . . . 495
27.170.2.4 difference_type . . . 495
27.170.2.5 iterator . . . 495
27.170.2.6 pointer . . . 495
27.170.2.7 reference . . . 495
27.170.2.8 reverse_iterator . . . 495
27.170.2.9 size_type . . . 495
27.170.2.10 Superclass . . . 495
27.170.2.11 value_type . . . 496

27.170.3 Constructor & Destructor Documentation . . . 496
27.170.3.1 LO . . . 496
27.170.3.2 LO . . . 496
27.170.3.3 LO . . . 496
27.170.3.4 LO . . . 496
27.170.4 Member Function Documentation . . . 496
27.170.4.1 IsValid . . . 496
27.171 gdcm::LookupTable Class Reference . . . 496
27.171.1 Detailed Description . . . 498
27.171.2 Member Enumeration Documentation . . . 498
27.171.2.1 LookupTableType . . . 498
27.171.3 Constructor & Destructor Documentation . . . 498
27.171.3.1 LookupTable . . . 498
27.171.3.2 ~LookupTable . . . 498
27.171.3.3 LookupTable . . . 498
27.171.4 Member Function Documentation . . . 498
27.171.4.1 Allocate . . . 499
27.171.4.2 Clear . . . 499
27.171.4.3 Decode . . . 499
27.171.4.4 Decode . . . 499
27.171.4.5 GetBitSample . . . 499
27.171.4.6 GetBufferAsRGBA . . . 499
27.171.4.7 GetLUT . . . 499
27.171.4.8 GetLUTDescriptor . . . 499
27.171.4.9 GetLUTLength . . . 499
27.171.4.10 GetPointer . . . 499
27.171.4.11 InitializeBlueLUT . . . 499
27.171.4.12 Initialized . . . 499
27.171.4.13 InitializeGreenLUT . . . 500
27.171.4.14 InitializeLUT . . . 500
27.171.4.15 InitializeRedLUT . . . 500
27.171.4.16 Print . . . 500
27.171.4.17 SetBlueLUT . . . 500
27.171.4.18 SetGreenLUT . . . 500
27.171.4.19 SetLUT . . . 500
27.171.4.20 SetRedLUT . . . 500
27.171.4.21 WriteBufferAsRGBA . . . 500

27.171.5 Member Data Documentation . . . 500
27.171.5.1 BitSample . . . 500
27.171.5.2 IncompleteLUT . . . 500
27.171.5.3 Internal . . . 500
27.172 gdcm::Scanner::ltstr Struct Reference . . . 500
27.172.1 Member Function Documentation . . . 501
27.172.1.1 operator() . . . 501
27.173 gdcm::Macro Class Reference . . . 501
27.173.1 Detailed Description . . . 501
27.173.2 Member Typedef Documentation . . . 502
27.173.2.1 ArrayIncludeMacrosType . . . 502
27.173.2.2 MapModuleEntry . . . 502
27.173.3 Constructor & Destructor Documentation . . . 502
27.173.3.1 Macro . . . 502
27.173.4 Member Function Documentation . . . 502
27.173.4.1 AddMacroEntry . . . 502
27.173.4.2 Clear . . . 502
27.173.4.3 FindMacroEntry . . . 502
27.173.4.4 GetMacroEntry . . . 502
27.173.4.5 GetName . . . 502
27.173.4.6 SetName . . . 502
27.173.4.7 Verify . . . 502
27.173.5 Friends And Related Function Documentation . . . 502
27.173.5.1 operator<< . . . 502
27.174 gdcm::Macros Class Reference . . . 502
27.174.1 Detailed Description . . . 503
27.174.2 Member Typedef Documentation . . . 503
27.174.2.1 ModuleMapType . . . 503
27.174.3 Constructor & Destructor Documentation . . . 503
27.174.3.1 Macros . . . 503
27.174.4 Member Function Documentation . . . 503
27.174.4.1 AddMacro . . . 503
27.174.4.2 Clear . . . 503
27.174.4.3 GetMacro . . . 503
27.174.4.4 IsEmpty . . . 503
27.174.5 Friends And Related Function Documentation . . . 504
27.174.5.1 operator<< . . . 504

27.175 gdcm::network::MaximumLengthSub Class Reference . . . 504
27.175.1 Detailed Description . . . 504
27.175.2 Constructor & Destructor Documentation . . . 504
27.175.2.1 MaximumLengthSub . . . 504
27.175.3 Member Function Documentation . . . 504
27.175.3.1 GetMaximumLength . . . 504
27.175.3.2 Print . . . 504
27.175.3.3 Read . . . 504
27.175.3.4 SetMaximumLength . . . 504
27.175.3.5 Size . . . 504
27.175.3.6 Write . . . 504
27.176 gdcm::MD5 Class Reference . . . 505
27.176.1 Detailed Description . . . 505
27.176.2 Constructor & Destructor Documentation . . . 505
27.176.2.1 MD5 . . . 505
27.176.2.2 ~MD5 . . . 505
27.176.3 Member Function Documentation . . . 505
27.176.3.1 Compute . . . 505
27.176.3.2 ComputeFile . . . 505
27.177 gdcm::MediaStorage Class Reference . . . 506
27.177.1 Detailed Description . . . 509
27.177.2 Member Enumeration Documentation . . . 509
27.177.2.1 MSType . . . 509
27.177.2.2 ObjectType . . . 511
27.177.3 Constructor & Destructor Documentation . . . 512
27.177.3.1 MediaStorage . . . 512
27.177.4 Member Function Documentation . . . 512
27.177.4.1 GetModality . . . 512
27.177.4.2 GetModalityDimension . . . 512
27.177.4.3 GetMSString . . . 512
27.177.4.4 GetMSType . . . 512
27.177.4.5 GetNumberOfModality . . . 512
27.177.4.6 GetNumberOfMSString . . . 512
27.177.4.7 GetNumberOfMSType . . . 512
27.177.4.8 GetString . . . 512
27.177.4.9 GuessFromModality . . . 512
27.177.4.10 Image . . . 512

27.177.4.11 IsUndefined . . . 513
27.177.4.12 operator MSType . . . 513
27.177.4.13 SetFromDataSet . . . 513
27.177.4.14 SetFromFile . . . 513
27.177.4.15 SetFromHeader . . . 513
27.177.4.16 SetFromModality . . . 513
27.177.4.17 SetFromSourceImageSequence . . . 513
27.177.5 Friends And Related Function Documentation . . . 513
27.177.5.1 operator<< . . . 513
27.178 gdcm::MemberCommand< T > Class Template Reference . . . 513
27.178.1 Detailed Description . . . 515
27.178.2 Member Typedef Documentation . . . 515
27.178.2.1 Self . . . 515
27.178.2.2 TConstMemberFunctionPointer . . . 516
27.178.2.3 TMemberFunctionPointer . . . 516
27.178.3 Constructor & Destructor Documentation . . . 516
27.178.3.1 MemberCommand . . . 516
27.178.3.2 ~MemberCommand . . . 516
27.178.4 Member Function Documentation . . . 516
27.178.4.1 Execute . . . 516
27.178.4.2 Execute . . . 516
27.178.4.3 New . . . 516
27.178.4.4 SetCallbackFunction . . . 516
27.178.4.5 SetCallbackFunction . . . 517
27.178.5 Member Data Documentation . . . 517
27.178.5.1 m_ConstMemberFunction . . . 517
27.178.5.2 m_MemberFunction . . . 517
27.178.5.3 m_This . . . 517
27.179 gdcm::MeshPrimitive Class Reference . . . 517
27.179.1 Detailed Description . . . 519
27.179.2 Member Typedef Documentation . . . 519
27.179.2.1 PrimitivesData . . . 519
27.179.3 Member Enumeration Documentation . . . 519
27.179.3.1 MPType . . . 519
27.179.4 Constructor & Destructor Documentation . . . 520
27.179.4.1 MeshPrimitive . . . 520
27.179.4.2 ~MeshPrimitive . . . 520

27.179.5 Member Function Documentation . . . 520
27.179.5.1 AddPrimitiveData . . . 520
27.179.5.2 GetMPType . . . 520
27.179.5.3 GetMPTypeString . . . 520
27.179.5.4 GetNumberOfPrimitivesData . . . 520
27.179.5.5 GetPrimitiveData . . . 520
27.179.5.6 GetPrimitiveData . . . 520
27.179.5.7 GetPrimitiveData . . . 520
27.179.5.8 GetPrimitiveData . . . 520
27.179.5.9 GetPrimitivesData . . . 520
27.179.5.10 GetPrimitivesData . . . 520
27.179.5.11 GetPrimitiveType . . . 520
27.179.5.12 SetPrimitiveData . . . 520
27.179.5.13 SetPrimitiveData . . . 520
27.179.5.14 SetPrimitivesData . . . 520
27.179.5.15 SetPrimitiveType . . . 520
27.179.6 Member Data Documentation . . . 520
27.179.6.1 PrimitiveData . . . 520
27.179.6.2 PrimitiveType . . . 520
27.180 gdcm::ModifiedEvent Class Reference . . . 520
27.181 gdcm::Module Class Reference . . . 522
27.181.1 Detailed Description . . . 522
27.181.2 Member Typedef Documentation . . . 523
27.181.2.1 ArrayIncludeMacrosType . . . 523
27.181.2.2 MapModuleEntry . . . 523
27.181.3 Constructor & Destructor Documentation . . . 523
27.181.3.1 Module . . . 523
27.181.4 Member Function Documentation . . . 523
27.181.4.1 AddMacro . . . 523
27.181.4.2 AddModuleEntry . . . 523
27.181.4.3 Clear . . . 523
27.181.4.4 FindModuleEntryInMacros . . . 523
27.181.4.5 GetModuleEntryInMacros . . . 523
27.181.4.6 GetName . . . 523
27.181.4.7 SetName . . . 523
27.181.4.8 Verify . . . 523
27.181.5 Friends And Related Function Documentation . . . 523

27.181.5.1operator<< . . . 523
27.182gdcm::ModuleEntry Class Reference . . . 524
27.182.1Detailed Description . . . 525
27.182.2Member Typedef Documentation . . . 525
27.182.2.1Description . . . 525
27.182.3Constructor & Destructor Documentation . . . 525
27.182.3.1ModuleEntry . . . 525
27.182.3.2~ModuleEntry . . . 526
27.182.4Member Function Documentation . . . 526
27.182.4.1GetDescription . . . 526
27.182.4.2GetName . . . 526
27.182.4.3GetType . . . 526
27.182.4.4SetDescription . . . 526
27.182.4.5SetName . . . 526
27.182.4.6SetType . . . 526
27.182.5Friends And Related Function Documentation . . . 526
27.182.5.1operator<< . . . 526
27.182.6Member Data Documentation . . . 526
27.182.6.1DataElementType . . . 526
27.182.6.2DescriptionField . . . 526
27.182.6.3Name . . . 526
27.183gdcm::Modules Class Reference . . . 526
27.183.1Detailed Description . . . 527
27.183.2Member Typedef Documentation . . . 527
27.183.2.1ModuleMapType . . . 527
27.183.3Constructor & Destructor Documentation . . . 527
27.183.3.1Modules . . . 527
27.183.4Member Function Documentation . . . 527
27.183.4.1AddModule . . . 527
27.183.4.2Clear . . . 527
27.183.4.3GetModule . . . 527
27.183.4.4IsEmpty . . . 528
27.183.5Friends And Related Function Documentation . . . 528
27.183.5.1operator<< . . . 528
27.184gdcm::MovePatientRootQuery Class Reference . . . 528
27.184.1Detailed Description . . . 529
27.184.2Constructor & Destructor Documentation . . . 529

27.184.2.1MovePatientRootQuery . . . 529
27.184.3Member Function Documentation . . . 529
27.184.3.1GetAbstractSyntaxUID . . . 529
27.184.3.2GetTagListByLevel . . . 530
27.184.3.3InitializeDataSet . . . 530
27.184.3.4ValidateQuery . . . 530
27.184.4Friends And Related Function Documentation . . . 530
27.184.4.1QueryFactory . . . 530
27.185gdcm::MoveStudyRootQuery Class Reference . . . 530
27.185.1Detailed Description . . . 532
27.185.2Constructor & Destructor Documentation . . . 532
27.185.2.1MoveStudyRootQuery . . . 532
27.185.3Member Function Documentation . . . 532
27.185.3.1GetAbstractSyntaxUID . . . 532
27.185.3.2GetTagListByLevel . . . 532
27.185.3.3InitializeDataSet . . . 532
27.185.3.4ValidateQuery . . . 532
27.185.4Friends And Related Function Documentation . . . 532
27.185.4.1QueryFactory . . . 532
27.186gdcm::NestedModuleEntries Class Reference . . . 533
27.186.1Detailed Description . . . 534
27.186.2Member Typedef Documentation . . . 534
27.186.2.1SizeType . . . 534
27.186.3Constructor & Destructor Documentation . . . 534
27.186.3.1NestedModuleEntries . . . 534
27.186.4Member Function Documentation . . . 534
27.186.4.1AddModuleEntry . . . 534
27.186.4.2GetModuleEntry . . . 534
27.186.4.3GetModuleEntry . . . 534
27.186.4.4GetNumberOfModuleEntries . . . 534
27.186.5Friends And Related Function Documentation . . . 535
27.186.5.1operator<< . . . 535
27.187gdcm::NoEvent Class Reference . . . 535
27.187.1Detailed Description . . . 535
27.188gdcm::Object Class Reference . . . 536
27.188.1Detailed Description . . . 537
27.188.2Constructor & Destructor Documentation . . . 537

27.188.2.1Object . . . 537
27.188.2.2∼Object . . . 537
27.188.2.3Object . . . 537
27.188.3Member Function Documentation . . . 537
27.188.3.1operator= . . . 537
27.188.3.2Print . . . 537
27.188.3.3Register . . . 538
27.188.3.4UnRegister . . . 538
27.188.4Friends And Related Function Documentation . . . 538
27.188.4.1operator<< . . . 538
27.188.4.2SmartPointer . . . 538
27.189gdcm::OpenSSLCryptoFactory Class Reference . . . 538
27.189.1Constructor & Destructor Documentation . . . 539
27.189.1.1OpenSSLCryptoFactory . . . 539
27.189.2Member Function Documentation . . . 539
27.189.2.1CreateCMSProvider . . . 539
27.189.2.2InitOpenSSL . . . 539
27.190gdcm::OpenSSLCryptographicMessageSyntax Class Reference . . . 539
27.190.1Constructor & Destructor Documentation . . . 541
27.190.1.1OpenSSLCryptographicMessageSyntax . . . 541
27.190.1.2∼OpenSSLCryptographicMessageSyntax . . . 541
27.190.2Member Function Documentation . . . 541
27.190.2.1Decrypt . . . 541
27.190.2.2Encrypt . . . 541
27.190.2.3GetCipherType . . . 541
27.190.2.4ParseCertificateFile . . . 541
27.190.2.5ParseKeyFile . . . 541
27.190.2.6SetCipherType . . . 541
27.190.2.7SetPassword . . . 541
27.191gdcm::OpenSSLP7CryptoFactory Class Reference . . . 542
27.191.1Constructor & Destructor Documentation . . . 542
27.191.1.1OpenSSLP7CryptoFactory . . . 542
27.191.2Member Function Documentation . . . 543
27.191.2.1CreateCMSProvider . . . 543
27.192gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference . . . 543
27.192.1Detailed Description . . . 544
27.192.2Constructor & Destructor Documentation . . . 544

27.192.2.1 OpenSSLP7CryptographicMessageSyntax . . . 544
27.192.2.2 ~OpenSSLP7CryptographicMessageSyntax . . . 544
27.192.3 Member Function Documentation . . . 544
27.192.3.1 Decrypt . . . 545
27.192.3.2 Encrypt . . . 545
27.192.3.3 GetCipherType . . . 545
27.192.3.4 ParseCertificateFile . . . 545
27.192.3.5 ParseKeyFile . . . 545
27.192.3.6 SetCipherType . . . 545
27.192.3.7 SetPassword . . . 545
27.193 gdcm::Orientation Class Reference . . . 545
27.193.1 Detailed Description . . . 546
27.193.2 Member Enumeration Documentation . . . 546
27.193.2.1 OrientationType . . . 546
27.193.3 Constructor & Destructor Documentation . . . 547
27.193.3.1 Orientation . . . 547
27.193.3.2 ~Orientation . . . 547
27.193.4 Member Function Documentation . . . 547
27.193.4.1 GetLabel . . . 547
27.193.4.2 GetMajorAxisFromPatientRelativeDirectionCosine . . . 547
27.193.4.3 GetObliquityThresholdCosineValue . . . 547
27.193.4.4 GetType . . . 547
27.193.4.5 Print . . . 547
27.193.4.6 SetObliquityThresholdCosineValue . . . 547
27.193.5 Friends And Related Function Documentation . . . 547
27.193.5.1 operator<< . . . 547
27.194 gdcm::Overlay Class Reference . . . 547
27.194.1 Detailed Description . . . 550
27.194.2 Member Enumeration Documentation . . . 550
27.194.2.1 OverlayType . . . 550
27.194.3 Constructor & Destructor Documentation . . . 550
27.194.3.1 Overlay . . . 550
27.194.3.2 ~Overlay . . . 550
27.194.3.3 Overlay . . . 550
27.194.4 Member Function Documentation . . . 550
27.194.4.1 Decompress . . . 550
27.194.4.2 GetBitPosition . . . 551

27.194.4.3 GetBitsAllocated . . . 551
27.194.4.4 GetColumns . . . 551
27.194.4.5 GetDescription . . . 551
27.194.4.6 GetGroup . . . 551
27.194.4.7 GetOrigin . . . 551
27.194.4.8 GetOverlayData . . . 551
27.194.4.9 GetOverlayTypeAsString . . . 551
27.194.4.10 GetOverlayTypeFromString . . . 551
27.194.4.11 GetRows . . . 551
27.194.4.12 GetType . . . 551
27.194.4.13 GetTypeAsEnum . . . 551
27.194.4.14 GetUnpackBuffer . . . 551
27.194.4.15 GetUnpackBufferLength . . . 552
27.194.4.16 GrabOverlayFromPixelData . . . 552
27.194.4.17 IsEmpty . . . 552
27.194.4.18 IsInPixelData . . . 552
27.194.4.19 IsInPixelData . . . 552
27.194.4.20 IsZero . . . 552
27.194.4.21 Print . . . 552
27.194.4.22 SetBitPosition . . . 552
27.194.4.23 SetBitsAllocated . . . 552
27.194.4.24 SetColumns . . . 552
27.194.4.25 SetDescription . . . 552
27.194.4.26 SetFrameOrigin . . . 552
27.194.4.27 SetGroup . . . 553
27.194.4.28 SetNumberOfFrames . . . 553
27.194.4.29 SetOrigin . . . 553
27.194.4.30 SetOverlay . . . 553
27.194.4.31 SetRows . . . 553
27.194.4.32 SetType . . . 553
27.194.4.33 Update . . . 553
27.195 gdcm::ParseException Class Reference . . . 553
27.195.1 Detailed Description . . . 555
27.195.2 Constructor & Destructor Documentation . . . 555
27.195.2.1 ParseException . . . 555
27.195.2.2 ~ParseException . . . 555
27.195.3 Member Function Documentation . . . 555

27.195.3.1 GetLastElement . . . 555
27.195.3.2 operator= . . . 555
27.195.3.3 SetLastElement . . . 555
27.196 gdcm::Parser Class Reference . . . 555
27.196.1 Detailed Description . . . 556
27.196.2 Member Typedef Documentation . . . 556
27.196.2.1 EndElementHandler . . . 556
27.196.2.2 StartElementHandler . . . 556
27.196.3 Member Enumeration Documentation . . . 556
27.196.3.1 ErrorType . . . 556
27.196.4 Constructor & Destructor Documentation . . . 557
27.196.4.1 Parser . . . 557
27.196.4.2 ~Parser . . . 557
27.196.5 Member Function Documentation . . . 557
27.196.5.1 GetBuffer . . . 557
27.196.5.2 GetCurrentByteIndex . . . 557
27.196.5.3 GetErrorCode . . . 557
27.196.5.4 GetErrorString . . . 557
27.196.5.5 GetUserData . . . 557
27.196.5.6 Parse . . . 557
27.196.5.7 ParseBuffer . . . 557
27.196.5.8 Process . . . 557
27.196.5.9 SetElementHandler . . . 557
27.196.5.10 SetUserData . . . 557
27.197 gdcm::Patient Class Reference . . . 557
27.197.1 Detailed Description . . . 557
27.197.2 Constructor & Destructor Documentation . . . 558
27.197.2.1 Patient . . . 558
27.198 gdcm::network::PDataTFPDU Class Reference . . . 558
27.198.1 Detailed Description . . . 559
27.198.2 Member Typedef Documentation . . . 559
27.198.2.1 SizeType . . . 559
27.198.3 Constructor & Destructor Documentation . . . 559
27.198.3.1 PDataTFPDU . . . 559
27.198.4 Member Function Documentation . . . 559
27.198.4.1 AddPresentationDataValue . . . 559
27.198.4.2 GetNumberOfPresentationDataValues . . . 559

27.198.4.3 GetPresentationDataValue . . . 559
27.198.4.4 IsLastFragment . . . 559
27.198.4.5 Print . . . 559
27.198.4.6 Read . . . 559
27.198.4.7 ReadInto . . . 560
27.198.4.8 Size . . . 560
27.198.4.9 Write . . . 560
27.199 gdcm::PDBElement Class Reference . . . 560
27.199.1 Detailed Description . . . 561
27.199.2 Constructor & Destructor Documentation . . . 561
27.199.2.1 PDBElement . . . 561
27.199.3 Member Function Documentation . . . 561
27.199.3.1 GetName . . . 561
27.199.3.2 GetValue . . . 561
27.199.3.3 operator== . . . 561
27.199.3.4 SetName . . . 561
27.199.3.5 SetValue . . . 561
27.199.4 Friends And Related Function Documentation . . . 561
27.199.4.1 operator<< . . . 561
27.199.5 Member Data Documentation . . . 562
27.199.5.1 NameField . . . 562
27.199.5.2 ValueField . . . 562
27.200 gdcm::PDBHeader Class Reference . . . 562
27.200.1 Detailed Description . . . 563
27.200.2 Constructor & Destructor Documentation . . . 563
27.200.2.1 PDBHeader . . . 563
27.200.2.2 ~PDBHeader . . . 563
27.200.3 Member Function Documentation . . . 563
27.200.3.1 FindPDBElementByName . . . 563
27.200.3.2 GetPDBEEnd . . . 563
27.200.3.3 GetPDBElementByName . . . 563
27.200.3.4 GetPDBInfoTag . . . 563
27.200.3.5 LoadFromDataElement . . . 563
27.200.3.6 Print . . . 564
27.200.4 Friends And Related Function Documentation . . . 564
27.200.4.1 operator<< . . . 564
27.201 gdcm::PDFCodec Class Reference . . . 564

27.201.1 Detailed Description . . . 565
27.201.2 Constructor & Destructor Documentation . . . 565
27.201.2.1 PDFCodec . . . 565
27.201.2.2 ~PDFCodec . . . 565
27.201.3 Member Function Documentation . . . 565
27.201.3.1 CanCode . . . 565
27.201.3.2 CanDecode . . . 566
27.201.3.3 Decode . . . 566
27.202 gdcm::network::PDUFactory Class Reference . . . 566
27.202.1 Detailed Description . . . 566
27.202.2 Member Function Documentation . . . 567
27.202.2.1 ConstructAbortPDU . . . 567
27.202.2.2 ConstructPDU . . . 567
27.202.2.3 ConstructReleasePDU . . . 567
27.202.2.4 CreateCEchoPDU . . . 567
27.202.2.5 CreateCFindPDU . . . 567
27.202.2.6 CreateCMovePDU . . . 567
27.202.2.7 CreateCStoreRQPDU . . . 567
27.202.2.8 CreateCStoreRSPPDU . . . 567
27.202.2.9 DetermineEventByPDU . . . 567
27.202.2.10 GetPDVs . . . 567
27.203 gdcm::PersonName Class Reference . . . 567
27.203.1 Detailed Description . . . 568
27.203.2 Member Function Documentation . . . 568
27.203.2.1 GetMaxLength . . . 568
27.203.2.2 GetNumberOfComponents . . . 568
27.203.2.3 Print . . . 568
27.203.2.4 SetBlob . . . 568
27.203.2.5 SetComponents . . . 568
27.203.2.6 SetComponents . . . 568
27.203.3 Member Data Documentation . . . 568
27.203.3.1 Component . . . 568
27.203.3.2 MaxLength . . . 568
27.203.3.3 MaxNumberOfComponents . . . 568
27.203.3.4 Padding . . . 568
27.203.3.5 Separator . . . 568
27.204 gdcm::PGXCodec Class Reference . . . 568

27.204.1 Detailed Description . . . 570
27.204.2 Constructor & Destructor Documentation . . . 570
27.204.2.1 PGXCodec . . . 570
27.204.2.2 ~PGXCodec . . . 570
27.204.3 Member Function Documentation . . . 570
27.204.3.1 CanCode . . . 570
27.204.3.2 CanDecode . . . 570
27.204.3.3 Clone . . . 570
27.204.3.4 GetHeaderInfo . . . 570
27.204.3.5 Read . . . 570
27.204.3.6 Write . . . 570
27.205 gdcm::PhotometricInterpretation Class Reference . . . 570
27.205.1 Detailed Description . . . 571
27.205.2 Member Enumeration Documentation . . . 572
27.205.2.1 PIType . . . 572
27.205.3 Constructor & Destructor Documentation . . . 572
27.205.3.1 PhotometricInterpretation . . . 572
27.205.4 Member Function Documentation . . . 572
27.205.4.1 GetPIString . . . 572
27.205.4.2 GetPIType . . . 572
27.205.4.3 GetSamplesPerPixel . . . 572
27.205.4.4 GetString . . . 572
27.205.4.5 GetType . . . 572
27.205.4.6 IsLossless . . . 572
27.205.4.7 IsLossy . . . 572
27.205.4.8 IsRetired . . . 572
27.205.4.9 IsSameColorSpace . . . 573
27.205.4.10 operator PIType . . . 573
27.205.5 Friends And Related Function Documentation . . . 573
27.205.5.1 operator<< . . . 573
27.206 gdcm::PixelFormat Class Reference . . . 573
27.206.1 Detailed Description . . . 574
27.206.2 Member Enumeration Documentation . . . 575
27.206.2.1 ScalarType . . . 575
27.206.3 Constructor & Destructor Documentation . . . 575
27.206.3.1 PixelFormat . . . 575
27.206.3.2 PixelFormat . . . 575

27.206.4 Member Function Documentation . . . 575
27.206.4.1 GetBitsAllocated . . . 575
27.206.4.2 GetBitsStored . . . 576
27.206.4.3 GetHighBit . . . 576
27.206.4.4 GetMax . . . 576
27.206.4.5 GetMin . . . 576
27.206.4.6 GetPixelRepresentation . . . 576
27.206.4.7 GetPixelSize . . . 576
27.206.4.8 GetSamplesPerPixel . . . 576
27.206.4.9 GetScalarType . . . 577
27.206.4.10 GetScalarTypeAsString . . . 577
27.206.4.11 IsCompatible . . . 577
27.206.4.12 IsValid . . . 577
27.206.4.13 operator ScalarType . . . 577
27.206.4.14 operator"!= . . . 577
27.206.4.15 operator"!= . . . 577
27.206.4.16 operator== . . . 577
27.206.4.17 operator== . . . 577
27.206.4.18 Print . . . 577
27.206.4.19 SetBitsAllocated . . . 577
27.206.4.20 SetBitsStored . . . 577
27.206.4.21 SetHighBit . . . 577
27.206.4.22 SetPixelRepresentation . . . 577
27.206.4.23 SetSamplesPerPixel . . . 577
27.206.4.24 SetScalarType . . . 577
27.206.4.25 Validate . . . 578
27.206.5 Friends And Related Function Documentation . . . 578
27.206.5.1 Bitmap . . . 578
27.206.5.2 operator<< . . . 578
27.207 gdcm::Pixmap Class Reference . . . 578
27.207.1 Detailed Description . . . 580
27.207.2 Constructor & Destructor Documentation . . . 580
27.207.2.1 Pixmap . . . 580
27.207.2.2 ~Pixmap . . . 580
27.207.3 Member Function Documentation . . . 580
27.207.3.1 AreOverlaysInPixelData . . . 580
27.207.3.2 GetCurve . . . 580

27.207.3.3 GetCurve . . . 580
27.207.3.4 GetIconImage . . . 580
27.207.3.5 GetIconImage . . . 580
27.207.3.6 GetNumberOfCurves . . . 580
27.207.3.7 GetNumberOfOverlays . . . 580
27.207.3.8 GetOverlay . . . 580
27.207.3.9 GetOverlay . . . 580
27.207.3.10 Print . . . 580
27.207.3.11 RemoveOverlay . . . 581
27.207.3.12 SetIconImage . . . 581
27.207.3.13 SetNumberOfCurves . . . 581
27.207.3.14 SetNumberOfOverlays . . . 581
27.207.4 Member Data Documentation . . . 581
27.207.4.1 Curves . . . 581
27.207.4.2 Icon . . . 581
27.207.4.3 Overlays . . . 581
27.208 gdcm::PixmapReader Class Reference . . . 581
27.208.1 Detailed Description . . . 583
27.208.2 Constructor & Destructor Documentation . . . 583
27.208.2.1 PixmapReader . . . 583
27.208.2.2 ~PixmapReader . . . 583
27.208.3 Member Function Documentation . . . 583
27.208.3.1 GetPixmap . . . 583
27.208.3.2 GetPixmap . . . 583
27.208.3.3 Read . . . 583
27.208.3.4 ReadACRNEMAImage . . . 583
27.208.3.5 ReadImage . . . 583
27.208.3.6 ReadImageInternal . . . 584
27.208.4 Member Data Documentation . . . 584
27.208.4.1 PixelData . . . 584
27.209 gdcm::PixmapToPixmapFilter Class Reference . . . 584
27.209.1 Detailed Description . . . 585
27.209.2 Constructor & Destructor Documentation . . . 585
27.209.2.1 PixmapToPixmapFilter . . . 585
27.209.2.2 ~PixmapToPixmapFilter . . . 586
27.209.3 Member Function Documentation . . . 586
27.209.3.1 GetInput . . . 586

27.209.3.2 GetOutput . . . 586
27.209.3.3 GetOutputAsPixmap . . . 586
27.210 gdcm::PixmapWriter Class Reference . . . 586
27.210.1 Detailed Description . . . 588
27.210.2 Constructor & Destructor Documentation . . . 588
27.210.2.1 PixmapWriter . . . 588
27.210.2.2 ~PixmapWriter . . . 588
27.210.3 Member Function Documentation . . . 588
27.210.3.1 DoIconImage . . . 588
27.210.3.2 GetImage . . . 588
27.210.3.3 GetImage . . . 588
27.210.3.4 GetPixmap . . . 588
27.210.3.5 GetPixmap . . . 588
27.210.3.6 PrepareWrite . . . 588
27.210.3.7 SetImage . . . 589
27.210.3.8 SetPixmap . . . 589
27.210.3.9 Write . . . 589
27.210.4 Member Data Documentation . . . 589
27.210.4.1 PixelData . . . 589
27.211 gdcm::PNMCodec Class Reference . . . 589
27.211.1 Detailed Description . . . 591
27.211.2 Constructor & Destructor Documentation . . . 591
27.211.2.1 PNMCodec . . . 591
27.211.2.2 ~PNMCodec . . . 591
27.211.3 Member Function Documentation . . . 591
27.211.3.1 CanCode . . . 591
27.211.3.2 CanDecode . . . 591
27.211.3.3 Clone . . . 591
27.211.3.4 GetBufferLength . . . 591
27.211.3.5 GetHeaderInfo . . . 591
27.211.3.6 Read . . . 592
27.211.3.7 SetBufferLength . . . 592
27.211.3.8 Write . . . 592
27.212 gdcm::Preamble Class Reference . . . 592
27.212.1 Detailed Description . . . 592
27.212.2 Constructor & Destructor Documentation . . . 593
27.212.2.1 Preamble . . . 593

27.212.2.2 ~Preamble . . . 593
27.212.2.3 Preamble . . . 593
27.212.3 Member Function Documentation . . . 593
27.212.3.1 Clear . . . 593
27.212.3.2 Create . . . 593
27.212.3.3 GetInternal . . . 593
27.212.3.4 GetLength . . . 593
27.212.3.5 IsEmpty . . . 593
27.212.3.6 IsValid . . . 593
27.212.3.7 operator= . . . 593
27.212.3.8 Print . . . 593
27.212.3.9 Read . . . 593
27.212.3.10 Remove . . . 593
27.212.3.11 Valid . . . 593
27.212.3.12 Write . . . 593
27.212.4 Friends And Related Function Documentation . . . 593
27.212.4.1 operator<< . . . 593
27.213 gdcm::PresentationContext Class Reference . . . 593
27.213.1 Detailed Description . . . 594
27.213.2 Member Typedef Documentation . . . 594
27.213.2.1 SizeType . . . 594
27.213.2.2 TransferSyntaxArrayType . . . 594
27.213.3 Constructor & Destructor Documentation . . . 594
27.213.3.1 PresentationContext . . . 594
27.213.3.2 PresentationContext . . . 594
27.213.4 Member Function Documentation . . . 594
27.213.4.1 AddTransferSyntax . . . 594
27.213.4.2 GetAbstractSyntax . . . 595
27.213.4.3 GetNumberOfTransferSyntaxes . . . 595
27.213.4.4 GetPresentationContextID . . . 595
27.213.4.5 GetTransferSyntax . . . 595
27.213.4.6 operator== . . . 595
27.213.4.7 Print . . . 595
27.213.4.8 SetAbstractSyntax . . . 595
27.213.4.9 SetPresentationContextID . . . 595
27.214 gdcm::network::PresentationContextAC Class Reference . . . 595
27.214.1 Detailed Description . . . 595

27.214.2 Constructor & Destructor Documentation . . . 596
27.214.2.1 PresentationContextAC . . . 596
27.214.3 Member Function Documentation . . . 596
27.214.3.1 GetPresentationContextID . . . 596
27.214.3.2 GetReason . . . 596
27.214.3.3 GetTransferSyntax . . . 596
27.214.3.4 Print . . . 596
27.214.3.5 Read . . . 596
27.214.3.6 SetPresentationContextID . . . 596
27.214.3.7 SetReason . . . 596
27.214.3.8 SetTransferSyntax . . . 596
27.214.3.9 Size . . . 596
27.214.3.10 Write . . . 596
27.215 gdcm::PresentationContextGenerator Class Reference . . . 596
27.215.1 Detailed Description . . . 597
27.215.2 Member Typedef Documentation . . . 597
27.215.2.1 PresentationContextArrayType . . . 597
27.215.2.2 SizeType . . . 597
27.215.3 Constructor & Destructor Documentation . . . 597
27.215.3.1 PresentationContextGenerator . . . 597
27.215.4 Member Function Documentation . . . 597
27.215.4.1 AddPresentationContext . . . 598
27.215.4.2 GenerateFromFilenames . . . 598
27.215.4.3 GenerateFromUID . . . 598
27.215.4.4 GetDefaultTransferSyntax . . . 598
27.215.4.5 GetPresentationContexts . . . 598
27.215.4.6 SetDefaultTransferSyntax . . . 598
27.215.4.7 SetMergeModeToAbstractSyntax . . . 598
27.215.4.8 SetMergeModeToTransferSyntax . . . 598
27.216 gdcm::network::PresentationContextRQ Class Reference . . . 598
27.216.1 Detailed Description . . . 599
27.216.2 Member Typedef Documentation . . . 599
27.216.2.1 SizeType . . . 599
27.216.3 Constructor & Destructor Documentation . . . 599
27.216.3.1 PresentationContextRQ . . . 599
27.216.3.2 PresentationContextRQ . . . 599
27.216.3.3 PresentationContextRQ . . . 599

27.216.4 Member Function Documentation . . . 599
27.216.4.1 AddTransferSyntax . . . 599
27.216.4.2 GetAbstractSyntax . . . 600
27.216.4.3 GetAbstractSyntax . . . 600
27.216.4.4 GetNumberOfTransferSyntaxes . . . 600
27.216.4.5 GetPresentationContextID . . . 600
27.216.4.6 GetTransferSyntax . . . 600
27.216.4.7 GetTransferSyntax . . . 600
27.216.4.8 GetTransferSyntaxes . . . 600
27.216.4.9 operator== . . . 600
27.216.4.10 Print . . . 600
27.216.4.11 Read . . . 600
27.216.4.12 SetAbstractSyntax . . . 600
27.216.4.13 SetPresentationContextID . . . 600
27.216.4.14 Size . . . 600
27.216.4.15 Write . . . 600
27.217 gdcm::network::PresentationDataValue Class Reference . . . 600
27.217.1 Detailed Description . . . 601
27.217.2 Constructor & Destructor Documentation . . . 601
27.217.2.1 PresentationDataValue . . . 601
27.217.3 Member Function Documentation . . . 601
27.217.3.1 ConcatenatePDVBlobs . . . 601
27.217.3.2 ConcatenatePDVBlobsAsExplicit . . . 601
27.217.3.3 GetBlob . . . 601
27.217.3.4 GetIsCommand . . . 601
27.217.3.5 GetIsLastFragment . . . 601
27.217.3.6 GetMessageHeader . . . 601
27.217.3.7 GetPresentationContextID . . . 601
27.217.3.8 Print . . . 601
27.217.3.9 Read . . . 601
27.217.3.10 ReadInto . . . 602
27.217.3.11 SetBlob . . . 602
27.217.3.12 SetCommand . . . 602
27.217.3.13 SetDataSet . . . 602
27.217.3.14 SetLastFragment . . . 602
27.217.3.15 SetMessageHeader . . . 602
27.217.3.16 SetPresentationContextID . . . 602

27.217.3.15 Size . . . 602
27.217.3.16 Write . . . 602
27.218 gdcm::Printer Class Reference . . . 602
27.218.1 Detailed Description . . . 604
27.218.2 Member Enumeration Documentation . . . 604
27.218.2.1 PrintStyles . . . 604
27.218.3 Constructor & Destructor Documentation . . . 604
27.218.3.1 Printer . . . 604
27.218.3.2 ~Printer . . . 604
27.218.4 Member Function Documentation . . . 604
27.218.4.1 GetPrintStyle . . . 604
27.218.4.2 Print . . . 604
27.218.4.3 PrintDataElement . . . 604
27.218.4.4 PrintDataSet . . . 604
27.218.4.5 PrintSQ . . . 605
27.218.4.6 SetColor . . . 605
27.218.4.7 SetFile . . . 605
27.218.4.8 SetStyle . . . 605
27.218.5 Member Data Documentation . . . 605
27.218.5.1 F . . . 605
27.218.5.2 MaxPrintLength . . . 605
27.218.5.3 PrintStyle . . . 605
27.219 gdcm::PrivateDict Class Reference . . . 605
27.219.1 Detailed Description . . . 606
27.219.2 Constructor & Destructor Documentation . . . 606
27.219.2.1 PrivateDict . . . 606
27.219.2.2 ~PrivateDict . . . 606
27.219.3 Member Function Documentation . . . 606
27.219.3.1 AddDictEntry . . . 606
27.219.3.2 FindDictEntry . . . 606
27.219.3.3 GetDictEntry . . . 606
27.219.3.4 IsEmpty . . . 606
27.219.3.5 LoadDefault . . . 606
27.219.3.6 PrintXML . . . 606
27.219.3.7 RemoveDictEntry . . . 606
27.219.4 Friends And Related Function Documentation . . . 606
27.219.4.1 Dicts . . . 606

27.219.4.2 operator<< . . . 606
27.220 gdcm::PrivateTag Class Reference . . . 607
27.220.1 Detailed Description . . . 608
27.220.2 Constructor & Destructor Documentation . . . 608
27.220.2.1 PrivateTag . . . 608
27.220.2.2 PrivateTag . . . 608
27.220.3 Member Function Documentation . . . 608
27.220.3.1 GetAsDataElement . . . 608
27.220.3.2 GetOwner . . . 608
27.220.3.3 operator< . . . 608
27.220.3.4 ReadFromCommaSeparatedString . . . 608
27.220.3.5 SetOwner . . . 608
27.220.4 Friends And Related Function Documentation . . . 608
27.220.4.1 operator<< . . . 608
27.221 gdcm::ProgressEvent Class Reference . . . 609
27.221.1 Detailed Description . . . 610
27.221.2 Member Typedef Documentation . . . 610
27.221.2.1 Self . . . 610
27.221.2.2 Superclass . . . 610
27.221.3 Constructor & Destructor Documentation . . . 610
27.221.3.1 ProgressEvent . . . 610
27.221.3.2 ~ProgressEvent . . . 610
27.221.3.3 ProgressEvent . . . 610
27.221.4 Member Function Documentation . . . 610
27.221.4.1 CheckEvent . . . 610
27.221.4.2 GetEventName . . . 610
27.221.4.3 GetProgress . . . 610
27.221.4.4 MakeObject . . . 610
27.221.4.5 SetProgress . . . 611
27.222 gdcm::PVRGCodec Class Reference . . . 611
27.222.1 Detailed Description . . . 612
27.222.2 Constructor & Destructor Documentation . . . 612
27.222.2.1 PVRGCodec . . . 612
27.222.2.2 ~PVRGCodec . . . 612
27.222.3 Member Function Documentation . . . 612
27.222.3.1 CanCode . . . 612
27.222.3.2 CanDecode . . . 612

27.222.3.3 Clone . . . 613
27.222.3.4 Code . . . 613
27.222.3.5 Decode . . . 613
27.222.3.6 SetLossyFlag . . . 613
27.223 gdcm::PythonFilter Class Reference . . . 613
27.223.1 Detailed Description . . . 613
27.223.2 Constructor & Destructor Documentation . . . 614
27.223.2.1 PythonFilter . . . 614
27.223.2.2 ~PythonFilter . . . 614
27.223.3 Member Function Documentation . . . 614
27.223.3.1 GetFile . . . 614
27.223.3.2 GetFile . . . 614
27.223.3.3 SetDicts . . . 614
27.223.3.4 SetFile . . . 614
27.223.3.5 ToPyObject . . . 614
27.223.3.6 UseDictAlways . . . 614
27.224 gdcm::QueryBase Class Reference . . . 614
27.224.1 Detailed Description . . . 615
27.224.2 Constructor & Destructor Documentation . . . 615
27.224.2.1 ~QueryBase . . . 615
27.224.3 Member Function Documentation . . . 615
27.224.3.1 GetAllRequiredTags . . . 615
27.224.3.2 GetAllTags . . . 615
27.224.3.3 GetHierachicalSearchTags . . . 615
27.224.3.4 GetName . . . 616
27.224.3.5 GetOptionalTags . . . 616
27.224.3.6 GetQueryLevel . . . 616
27.224.3.7 GetRequiredTags . . . 616
27.224.3.8 GetUniqueTags . . . 616
27.225 gdcm::QueryFactory Class Reference . . . 616
27.225.1 Detailed Description . . . 616
27.225.2 Member Function Documentation . . . 617
27.225.2.1 GetCharacterFromCurrentLocale . . . 617
27.225.2.2 ListCharSets . . . 617
27.225.2.3 ProduceCharacterSetDataElement . . . 617
27.225.2.4 ProduceQuery . . . 617
27.226 gdcm::QueryImage Class Reference . . . 617

27.226.1 Detailed Description . . . 618
27.226.2 Member Function Documentation . . . 618
27.226.2.1 GetHierachicalSearchTags . . . 619
27.226.2.2 GetName . . . 619
27.226.2.3 GetOptionalTags . . . 619
27.226.2.4 GetQueryLevel . . . 619
27.226.2.5 GetRequiredTags . . . 619
27.226.2.6 GetUniqueTags . . . 619
27.227 gdcm::QueryPatient Class Reference . . . 619
27.227.1 Detailed Description . . . 620
27.227.2 Member Function Documentation . . . 620
27.227.2.1 GetHierachicalSearchTags . . . 621
27.227.2.2 GetName . . . 621
27.227.2.3 GetOptionalTags . . . 621
27.227.2.4 GetQueryLevel . . . 621
27.227.2.5 GetRequiredTags . . . 621
27.227.2.6 GetUniqueTags . . . 621
27.228 gdcm::QuerySeries Class Reference . . . 621
27.228.1 Detailed Description . . . 622
27.228.2 Member Function Documentation . . . 622
27.228.2.1 GetHierachicalSearchTags . . . 623
27.228.2.2 GetName . . . 623
27.228.2.3 GetOptionalTags . . . 623
27.228.2.4 GetQueryLevel . . . 623
27.228.2.5 GetRequiredTags . . . 623
27.228.2.6 GetUniqueTags . . . 623
27.229 gdcm::QueryStudy Class Reference . . . 623
27.229.1 Detailed Description . . . 624
27.229.2 Member Function Documentation . . . 624
27.229.2.1 GetHierachicalSearchTags . . . 625
27.229.2.2 GetName . . . 625
27.229.2.3 GetOptionalTags . . . 625
27.229.2.4 GetQueryLevel . . . 625
27.229.2.5 GetRequiredTags . . . 625
27.229.2.6 GetUniqueTags . . . 625
27.230 gdcm::RAWCodec Class Reference . . . 625
27.230.1 Detailed Description . . . 627

27.230.2 Constructor & Destructor Documentation . . . 627
27.230.2.1 RAWCodec . . . 627
27.230.2.2 ~RAWCodec . . . 627
27.230.3 Member Function Documentation . . . 627
27.230.3.1 CanCode . . . 627
27.230.3.2 CanDecode . . . 627
27.230.3.3 Clone . . . 627
27.230.3.4 Code . . . 627
27.230.3.5 Decode . . . 627
27.230.3.6 DecodeByStreams . . . 628
27.230.3.7 DecodeBytes . . . 628
27.230.3.8 GetHeaderInfo . . . 628
27.231 gdcm::Reader Class Reference . . . 628
27.231.1 Detailed Description . . . 630
27.231.2 Constructor & Destructor Documentation . . . 631
27.231.2.1 Reader . . . 631
27.231.2.2 ~Reader . . . 631
27.231.3 Member Function Documentation . . . 631
27.231.3.1 CanRead . . . 631
27.231.3.2 GetFile . . . 631
27.231.3.3 GetFile . . . 631
27.231.3.4 GetStreamCurrentPosition . . . 631
27.231.3.5 GetStreamPtr . . . 631
27.231.3.6 Read . . . 631
27.231.3.7 ReadDataSet . . . 632
27.231.3.8 ReadMetaInformation . . . 632
27.231.3.9 ReadPreamble . . . 632
27.231.3.10 ReadSelectedPrivateTags . . . 632
27.231.3.11 ReadSelectedTags . . . 632
27.231.3.12 ReadUpToTag . . . 632
27.231.3.13 SetFile . . . 632
27.231.3.14 SetFileName . . . 632
27.231.3.15 SetStream . . . 633
27.231.4 Friends And Related Function Documentation . . . 633
27.231.4.1 StreamImageReader . . . 633
27.231.5 Member Data Documentation . . . 633
27.231.5.1 F . . . 633

27.232 gdcm::Region Class Reference . . . 633
27.232.1 Detailed Description . . . 634
27.232.2 Constructor & Destructor Documentation . . . 634
27.232.2.1 Region . . . 634
27.232.2.2 ~Region . . . 634
27.232.3 Member Function Documentation . . . 634
27.232.3.1 Area . . . 634
27.232.3.2 Clone . . . 634
27.232.3.3 ComputeBoundingBox . . . 634
27.232.3.4 Empty . . . 634
27.232.3.5 IsValid . . . 635
27.232.3.6 Print . . . 635
27.233 gdcm::Rescaler Class Reference . . . 635
27.233.1 Detailed Description . . . 636
27.233.2 Constructor & Destructor Documentation . . . 636
27.233.2.1 Rescaler . . . 636
27.233.2.2 ~Rescaler . . . 636
27.233.3 Member Function Documentation . . . 636
27.233.3.1 ComputeInterceptSlopePixelType . . . 636
27.233.3.2 ComputePixelTypeFromMinMax . . . 637
27.233.3.3 GetIntercept . . . 637
27.233.3.4 GetSlope . . . 637
27.233.3.5 InverseRescale . . . 637
27.233.3.6 InverseRescaleFunctionIntoBestFit . . . 637
27.233.3.7 Rescale . . . 637
27.233.3.8 RescaleFunctionIntoBestFit . . . 637
27.233.3.9 SetIntercept . . . 637
27.233.3.10 SetMinMaxForPixelType . . . 637
27.233.3.11 SetPixelFormat . . . 637
27.233.3.12 SetSlope . . . 637
27.233.3.13 SetTargetPixelType . . . 637
27.233.3.14 SetUseTargetPixelType . . . 637
27.234 gdcm::RLECodec Class Reference . . . 638
27.234.1 Detailed Description . . . 639
27.234.2 Constructor & Destructor Documentation . . . 639
27.234.2.1 RLECodec . . . 639
27.234.2.2 ~RLECodec . . . 639

27.234.3 Member Function Documentation . . . 640
27.234.3.1 AppendFrameEncode . . . 640
27.234.3.2 AppendRowEncode . . . 640
27.234.3.3 CanCode . . . 640
27.234.3.4 CanDecode . . . 640
27.234.3.5 Clone . . . 640
27.234.3.6 Code . . . 640
27.234.3.7 Decode . . . 640
27.234.3.8 DecodeByStreams . . . 640
27.234.3.9 DecodeExtent . . . 640
27.234.3.10 GetBufferLength . . . 640
27.234.3.11 GetHeaderInfo . . . 641
27.234.3.12 GetFrameEncoder . . . 641
27.234.3.13 GetRowEncoder . . . 641
27.234.3.14 GetBufferLength . . . 641
27.234.3.15 SetLength . . . 641
27.234.3.16 StartEncode . . . 641
27.234.3.17 StopEncode . . . 641
27.234.4 Friends And Related Function Documentation . . . 641
27.234.4.1 ImageRegionReader . . . 641
27.235 gdcm::network::RoleSelectionSub Class Reference . . . 641
27.235.1 Detailed Description . . . 642
27.235.2 Constructor & Destructor Documentation . . . 642
27.235.2.1 RoleSelectionSub . . . 642
27.235.3 Member Function Documentation . . . 642
27.235.3.1 Print . . . 642
27.235.3.2 Read . . . 642
27.235.3.3 SetTuple . . . 642
27.235.3.4 Size . . . 642
27.235.3.5 Write . . . 642
27.236 gdcm::SerieHelper::Rule Struct Reference . . . 642
27.236.1 Member Data Documentation . . . 643
27.236.1.1 elem . . . 643
27.236.1.2 group . . . 643
27.236.1.3 op . . . 643
27.236.1.4 value . . . 643
27.237 gdcm::Scanner Class Reference . . . 643

27.237.1 Detailed Description . . . 645
27.237.2 Member Typedef Documentation . . . 646
27.237.2.1 ConstIterator . . . 646
27.237.2.2 MappingType . . . 646
27.237.2.3 TagToValue . . . 646
27.237.2.4 TagToValueValueType . . . 646
27.237.2.5 ValuesType . . . 646
27.237.3 Constructor & Destructor Documentation . . . 646
27.237.3.1 Scanner . . . 646
27.237.3.2 ~Scanner . . . 646
27.237.4 Member Function Documentation . . . 646
27.237.4.1 AddPrivateTag . . . 646
27.237.4.2 AddSkipTag . . . 646
27.237.4.3 AddTag . . . 646
27.237.4.4 Begin . . . 647
27.237.4.5 ClearSkipTags . . . 647
27.237.4.6 ClearTags . . . 647
27.237.4.7 End . . . 647
27.237.4.8 GetAllFilenamesFromTagToValue . . . 647
27.237.4.9 GetFilenameFromTagToValue . . . 647
27.237.4.10 GetFilenames . . . 647
27.237.4.11 GetKeys . . . 647
27.237.4.12 GetMapping . . . 647
27.237.4.13 GetMappingFromTagToValue . . . 647
27.237.4.14 GetMappings . . . 647
27.237.4.15 GetOrderedValues . . . 647
27.237.4.16 GetValue . . . 648
27.237.4.17 GetValues . . . 648
27.237.4.18 GetValues . . . 648
27.237.4.19 IsKey . . . 648
27.237.4.20 New . . . 648
27.237.4.21 Print . . . 648
27.237.4.22 ProcessPublicTag . . . 648
27.237.4.23 Scan . . . 648
27.237.5 Friends And Related Function Documentation . . . 649
27.237.5.1 operator<< . . . 649
27.238 gdcm::Segment Class Reference . . . 649

27.238.1 Detailed Description . . . 651
27.238.2 Member Typedef Documentation . . . 651
27.238.2.1 SurfaceVector . . . 651
27.238.3 Member Enumeration Documentation . . . 651
27.238.3.1 ALGOType . . . 651
27.238.4 Constructor & Destructor Documentation . . . 651
27.238.4.1 Segment . . . 651
27.238.4.2 ~Segment . . . 651
27.238.5 Member Function Documentation . . . 651
27.238.5.1 AddSurface . . . 651
27.238.5.2 GetALGOType . . . 651
27.238.5.3 GetALGOTypeString . . . 651
27.238.5.4 GetAnatomicRegion . . . 651
27.238.5.5 GetAnatomicRegion . . . 651
27.238.5.6 GetPropertyCategory . . . 651
27.238.5.7 GetPropertyCategory . . . 652
27.238.5.8 GetPropertyType . . . 652
27.238.5.9 GetPropertyType . . . 652
27.238.5.10 GetSegmentAlgorithmName . . . 652
27.238.5.11 GetSegmentAlgorithmType . . . 652
27.238.5.12 GetSegmentDescription . . . 652
27.238.5.13 GetSegmentLabel . . . 652
27.238.5.14 GetSegmentNumber . . . 652
27.238.5.15 GetSurface . . . 652
27.238.5.16 GetSurfaceCount . . . 652
27.238.5.17 GetSurfaces . . . 652
27.238.5.18 GetSurfaces . . . 652
27.238.5.19 SetAnatomicRegion . . . 652
27.238.5.20 SetPropertyCategory . . . 652
27.238.5.21 SetPropertyType . . . 652
27.238.5.22 SetSegmentAlgorithmName . . . 652
27.238.5.23 SetSegmentAlgorithmType . . . 652
27.238.5.24 SetSegmentAlgorithmType . . . 652
27.238.5.25 SetSegmentDescription . . . 652
27.238.5.26 SetSegmentLabel . . . 652
27.238.5.27 SetSegmentNumber . . . 652
27.238.5.28 SetSurfaceCount . . . 652

27.238.6 Member Data Documentation . . . 652
27.238.6.1 AnatomicRegion . . . 652
27.238.6.2 PropertyCategory . . . 652
27.238.6.3 PropertyType . . . 652
27.238.6.4 SegmentAlgorithmName . . . 653
27.238.6.5 SegmentAlgorithmType . . . 653
27.238.6.6 SegmentDescription . . . 653
27.238.6.7 SegmentLabel . . . 653
27.238.6.8 SegmentNumber . . . 653
27.238.6.9 SurfaceCount . . . 653
27.238.6.10 Surfaces . . . 653
27.239 gdcm::SegmentedPaletteColorLookupTable Class Reference . . . 653
27.239.1 Detailed Description . . . 654
27.239.2 Constructor & Destructor Documentation . . . 654
27.239.2.1 SegmentedPaletteColorLookupTable . . . 654
27.239.2.2 ~SegmentedPaletteColorLookupTable . . . 654
27.239.3 Member Function Documentation . . . 654
27.239.3.1 Print . . . 654
27.239.3.2 SetLUT . . . 655
27.240 gdcm::SegmentReader Class Reference . . . 655
27.240.1 Detailed Description . . . 657
27.240.2 Member Typedef Documentation . . . 657
27.240.2.1 SegmentMap . . . 657
27.240.2.2 SegmentVector . . . 657
27.240.3 Constructor & Destructor Documentation . . . 657
27.240.3.1 SegmentReader . . . 657
27.240.3.2 ~SegmentReader . . . 657
27.240.4 Member Function Documentation . . . 657
27.240.4.1 GetSegments . . . 657
27.240.4.2 GetSegments . . . 657
27.240.4.3 Read . . . 657
27.240.4.4 ReadSegment . . . 657
27.240.4.5 ReadSegments . . . 657
27.240.5 Member Data Documentation . . . 657
27.240.5.1 Segments . . . 657
27.241 gdcm::SegmentWriter Class Reference . . . 658
27.241.1 Detailed Description . . . 659

27.241.2 Member Typedef Documentation . . . 659
27.241.2.1 SegmentVector . . . 659
27.241.3 Constructor & Destructor Documentation . . . 659
27.241.3.1 SegmentWriter . . . 659
27.241.3.2 ~SegmentWriter . . . 659
27.241.4 Member Function Documentation . . . 659
27.241.4.1 AddSegment . . . 659
27.241.4.2 GetNumberOfSegments . . . 659
27.241.4.3 GetSegment . . . 659
27.241.4.4 GetSegments . . . 659
27.241.4.5 GetSegments . . . 659
27.241.4.6 PrepareWrite . . . 659
27.241.4.7 SetNumberOfSegments . . . 659
27.241.4.8 SetSegments . . . 659
27.241.4.9 Write . . . 660
27.241.5 Member Data Documentation . . . 660
27.241.5.1 Segments . . . 660
27.242 gdcm::SequenceOfFragments Class Reference . . . 660
27.242.1 Detailed Description . . . 662
27.242.2 Member Typedef Documentation . . . 662
27.242.2.1 ConstIterator . . . 662
27.242.2.2 FragmentVector . . . 662
27.242.2.3 Iterator . . . 662
27.242.2.4 SizeType . . . 662
27.242.3 Constructor & Destructor Documentation . . . 662
27.242.3.1 SequenceOfFragments . . . 662
27.242.4 Member Function Documentation . . . 663
27.242.4.1 AddFragment . . . 663
27.242.4.2 Begin . . . 663
27.242.4.3 Begin . . . 663
27.242.4.4 Clear . . . 663
27.242.4.5 ComputeByteLength . . . 663
27.242.4.6 ComputeLength . . . 663
27.242.4.7 End . . . 663
27.242.4.8 End . . . 663
27.242.4.9 GetBuffer . . . 663
27.242.4.10 GetFragBuffer . . . 663

27.242.4.10GetFragment . . . 663
27.242.4.11GetLength . . . 663
27.242.4.12GetNumberOfFragments . . . 663
27.242.4.13GetTable . . . 664
27.242.4.14GetTable . . . 664
27.242.4.15New . . . 664
27.242.4.16operator== . . . 664
27.242.4.17Print . . . 664
27.242.4.18Read . . . 664
27.242.4.19ReadPreValue . . . 664
27.242.4.20ReadValue . . . 664
27.242.4.21SetLength . . . 664
27.242.4.22Write . . . 664
27.242.4.23WriteBuffer . . . 664
27.243gdcm::SequenceOfItems Class Reference . . . 665
27.243.1Detailed Description . . . 667
27.243.2Member Typedef Documentation . . . 668
27.243.2.1ConstIterator . . . 668
27.243.2.2ItemVector . . . 668
27.243.2.3Iterator . . . 668
27.243.2.4SizeType . . . 668
27.243.3Constructor & Destructor Documentation . . . 668
27.243.3.1SequenceOfItems . . . 668
27.243.4Member Function Documentation . . . 668
27.243.4.1AddItem . . . 668
27.243.4.2Begin . . . 668
27.243.4.3Begin . . . 668
27.243.4.4Clear . . . 668
27.243.4.5ComputeLength . . . 668
27.243.4.6End . . . 668
27.243.4.7End . . . 669
27.243.4.8FindDataElement . . . 669
27.243.4.9GetItem . . . 669
27.243.4.10GetItem . . . 669
27.243.4.11GetLength . . . 669
27.243.4.12GetNumberOfItems . . . 669
27.243.4.13IsUndefinedLength . . . 669

27.243.4.14New . . . 669
27.243.4.15operator= . . . 669
27.243.4.16operator== . . . 669
27.243.4.17Print . . . 670
27.243.4.18Read . . . 670
27.243.4.19RemoveItemByIndex . . . 670
27.243.4.20SetLength . . . 670
27.243.4.21SetLengthToUndefined . . . 670
27.243.4.22SetNumberOfItems . . . 670
27.243.4.23Write . . . 670
27.243.5Member Data Documentation . . . 670
27.243.5.1Items . . . 670
27.243.5.2SequenceLengthField . . . 671
27.244gdcm::SerieHelper Class Reference . . . 671
27.244.1Detailed Description . . . 672
27.244.2Member Typedef Documentation . . . 672
27.244.2.1SerieRestrictions . . . 672
27.244.2.2SingleSerieUIDFileSetmap . . . 672
27.244.3Constructor & Destructor Documentation . . . 672
27.244.3.1SerieHelper . . . 673
27.244.3.2~SerieHelper . . . 673
27.244.4Member Function Documentation . . . 673
27.244.4.1AddFile . . . 673
27.244.4.2AddFileName . . . 673
27.244.4.3AddRestriction . . . 673
27.244.4.4AddRestriction . . . 673
27.244.4.5AddRestriction . . . 673
27.244.4.6Clear . . . 673
27.244.4.7CreateDefaultUniqueSeriesIdentifier . . . 673
27.244.4.8CreateUniqueSeriesIdentifier . . . 673
27.244.4.9FileNameOrdering . . . 673
27.244.4.10GetFirstSingleSerieUIDFileSet . . . 673
27.244.4.11GetNextSingleSerieUIDFileSet . . . 673
27.244.4.12ImagePositionPatientOrdering . . . 673
27.244.4.13OrderFileList . . . 673
27.244.4.14SetDirectory . . . 673
27.244.4.15SetLoadMode . . . 673

27.244.4.15 SetUseSeriesDetails . . . 673
27.244.4.17 UserOrdering . . . 673
27.244.5 Member Data Documentation . . . 673
27.244.5.1 ItFileSetHt . . . 673
27.244.5.2 SingleSerieUIDFileSetHT . . . 673
27.245 gdcm::Series Class Reference . . . 674
27.245.1 Detailed Description . . . 674
27.245.2 Constructor & Destructor Documentation . . . 674
27.245.2.1 Series . . . 674
27.246 gdcm::network::ServiceClassApplicationInformation Class Reference . . . 674
27.246.1 Detailed Description . . . 674
27.246.2 Constructor & Destructor Documentation . . . 674
27.246.2.1 ServiceClassApplicationInformation . . . 674
27.246.3 Member Function Documentation . . . 674
27.246.3.1 Print . . . 674
27.246.3.2 Read . . . 675
27.246.3.3 SetTuple . . . 675
27.246.3.4 Size . . . 675
27.246.3.5 Write . . . 675
27.247 gdcm::ServiceClassUser Class Reference . . . 675
27.247.1 Detailed Description . . . 677
27.247.2 Constructor & Destructor Documentation . . . 677
27.247.2.1 ServiceClassUser . . . 677
27.247.2.2 ~ServiceClassUser . . . 677
27.247.3 Member Function Documentation . . . 677
27.247.3.1 GetAETitle . . . 677
27.247.3.2 GetCalledAETitle . . . 678
27.247.3.3 GetTimeout . . . 678
27.247.3.4 InitializeConnection . . . 678
27.247.3.5 IsPresentationContextAccepted . . . 678
27.247.3.6 New . . . 678
27.247.3.7 SendEcho . . . 678
27.247.3.8 SendFind . . . 678
27.247.3.9 SendMove . . . 678
27.247.3.10 SendMove . . . 678
27.247.3.11 SendMove . . . 678
27.247.3.12 SendStore . . . 678

27.247.3.15 SendStore . . . 679
27.247.3.16 SendStore . . . 679
27.247.3.17 SetAETitle . . . 679
27.247.3.18 SetCalledAETitle . . . 679
27.247.3.19 SetHostname . . . 679
27.247.3.20 SetPort . . . 679
27.247.3.21 SetPortSCP . . . 679
27.247.3.22 SetPresentationContexts . . . 679
27.247.3.23 SetTimeout . . . 680
27.247.3.24 StartAssociation . . . 680
27.247.3.25 StopAssociation . . . 680
27.248 gdcm::SHA1 Class Reference . . . 680
27.248.1 Detailed Description . . . 681
27.248.2 Constructor & Destructor Documentation . . . 681
27.248.2.1 SHA1 . . . 681
27.248.2.2 ~SHA1 . . . 681
27.248.3 Member Function Documentation . . . 681
27.248.3.1 Compute . . . 681
27.248.3.2 ComputeFile . . . 681
27.249 gdcm::SimpleMemberCommand< T > Class Template Reference . . . 681
27.249.1 Detailed Description . . . 683
27.249.2 Member Typedef Documentation . . . 683
27.249.2.1 Self . . . 683
27.249.2.2 TMemberFunctionPointer . . . 683
27.249.3 Constructor & Destructor Documentation . . . 684
27.249.3.1 SimpleMemberCommand . . . 684
27.249.3.2 ~SimpleMemberCommand . . . 684
27.249.4 Member Function Documentation . . . 684
27.249.4.1 Execute . . . 684
27.249.4.2 Execute . . . 684
27.249.4.3 New . . . 684
27.249.4.4 SetCallbackFunction . . . 684
27.249.5 Member Data Documentation . . . 684
27.249.5.1 m_MemberFunction . . . 684
27.249.5.2 m_This . . . 685
27.250 gdcm::SimpleSubjectWatcher Class Reference . . . 685
27.250.1 Detailed Description . . . 685

27.250.2 Constructor & Destructor Documentation . . . 685
27.250.2.1 SimpleSubjectWatcher . . . 685
27.250.2.2 ~SimpleSubjectWatcher . . . 685
27.250.3 Member Function Documentation . . . 685
27.250.3.1 EndFilter . . . 686
27.250.3.2 ShowAbort . . . 686
27.250.3.3 ShowAnonymization . . . 686
27.250.3.4 ShowData . . . 686
27.250.3.5 ShowDataSet . . . 686
27.250.3.6 ShowFileName . . . 686
27.250.3.7 ShowIteration . . . 686
27.250.3.8 ShowProgress . . . 686
27.250.3.9 StartFilter . . . 686
27.250.3.10 TestAbortOff . . . 686
27.250.3.11 TestAbortOn . . . 686
27.251 gdcm::SmartPointer< ObjectType > Class Template Reference . . . 686
27.251.1 Detailed Description . . . 688
27.251.2 Constructor & Destructor Documentation . . . 688
27.251.2.1 SmartPointer . . . 688
27.251.2.2 SmartPointer . . . 688
27.251.2.3 SmartPointer . . . 688
27.251.2.4 SmartPointer . . . 688
27.251.2.5 ~SmartPointer . . . 688
27.251.3 Member Function Documentation . . . 688
27.251.3.1 GetPointer . . . 689
27.251.3.2 operator ObjectType * . . . 689
27.251.3.3 operator* . . . 689
27.251.3.4 operator-> . . . 689
27.251.3.5 operator= . . . 689
27.251.3.6 operator= . . . 689
27.251.3.7 operator= . . . 689
27.252 gdcm::network::SOPClassExtendedNegociationSub Class Reference . . . 689
27.252.1 Detailed Description . . . 690
27.252.2 Constructor & Destructor Documentation . . . 690
27.252.2.1 SOPClassExtendedNegociationSub . . . 690
27.252.3 Member Function Documentation . . . 690
27.252.3.1 Print . . . 690

27.252.3.2 Read . . . 690
27.252.3.3 SetTuple . . . 690
27.252.3.4 Size . . . 690
27.252.3.5 Write . . . 690
27.253 gdcm::SOPClassUIDToIOD Class Reference . . . 690
27.253.1 Detailed Description . . . 691
27.253.2 Member Typedef Documentation . . . 691
27.253.2.1 const . . . 691
27.253.3 Member Function Documentation . . . 691
27.253.3.1 GetIOD . . . 691
27.253.3.2 GetIODFromSOPClassUID . . . 691
27.253.3.3 GetNumberOfSOPClassToIOD . . . 691
27.253.3.4 GetSOPClassUIDFromIOD . . . 691
27.253.3.5 GetSOPClassUIDToIOD . . . 691
27.253.3.6 GetSOPClassUIDToIODs . . . 691
27.254 gdcm::Sorter Class Reference . . . 691
27.254.1 Detailed Description . . . 693
27.254.2 Member Typedef Documentation . . . 693
27.254.2.1 SelectionMap . . . 693
27.254.2.2 SortFunction . . . 693
27.254.3 Constructor & Destructor Documentation . . . 694
27.254.3.1 Sorter . . . 694
27.254.3.2 ~Sorter . . . 694
27.254.4 Member Function Documentation . . . 694
27.254.4.1 AddSelect . . . 694
27.254.4.2 GetFilenames . . . 694
27.254.4.3 Print . . . 694
27.254.4.4 SetSortFunction . . . 694
27.254.4.5 Sort . . . 694
27.254.4.6 StableSort . . . 694
27.254.5 Friends And Related Function Documentation . . . 695
27.254.5.1 operator<< . . . 695
27.254.6 Member Data Documentation . . . 695
27.254.6.1 Filenames . . . 695
27.254.6.2 Selection . . . 695
27.254.6.3 SortFunc . . . 695
27.255 gdcm::Spacing Class Reference . . . 695

27.255.1 Detailed Description . . . 695
27.255.2 Member Enumeration Documentation . . . 696
27.255.2.1 SpacingType . . . 696
27.255.3 Constructor & Destructor Documentation . . . 696
27.255.3.1 Spacing . . . 696
27.255.3.2 ~Spacing . . . 696
27.255.4 Member Function Documentation . . . 696
27.255.4.1 ComputePixelAspectRatioFromPixelSpacing . . . 696
27.256 gdcm::Spectroscopy Class Reference . . . 697
27.256.1 Detailed Description . . . 697
27.256.2 Constructor & Destructor Documentation . . . 697
27.256.2.1 Spectroscopy . . . 697
27.257 gdcm::SplitMosaicFilter Class Reference . . . 697
27.257.1 Detailed Description . . . 698
27.257.2 Constructor & Destructor Documentation . . . 698
27.257.2.1 SplitMosaicFilter . . . 698
27.257.2.2 ~SplitMosaicFilter . . . 698
27.257.3 Member Function Documentation . . . 698
27.257.3.1 ComputeMOSAICDimensions . . . 698
27.257.3.2 GetFile . . . 698
27.257.3.3 GetFile . . . 698
27.257.3.4 GetImage . . . 698
27.257.3.5 GetImage . . . 698
27.257.3.6 SetFile . . . 698
27.257.3.7 SetImage . . . 698
27.257.3.8 Split . . . 698
27.258 gdcm::StartEvent Class Reference . . . 698
27.259 gdcm::static_assert_test< x > Struct Template Reference . . . 700
27.260 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference . . . 700
27.261 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference . . . 700
27.261.1 Member Enumeration Documentation . . . 700
27.261.1.1 anonymous enum . . . 700
27.262 gdcm::StreamImageReader Class Reference . . . 700
27.262.1 Detailed Description . . . 701
27.262.2 Constructor & Destructor Documentation . . . 701
27.262.2.1 StreamImageReader . . . 701
27.262.2.2 ~StreamImageReader . . . 701

27.262.3 Member Function Documentation . . . 701
27.262.3.1 CanReadImage . . . 701
27.262.3.2 DefinePixelExtent . . . 701
27.262.3.3 DefineProperBufferLength . . . 702
27.262.3.4 GetDimensionsValueForResolution . . . 702
27.262.3.5 GetFile . . . 702
27.262.3.6 Read . . . 702
27.262.3.7 ReadImageInformation . . . 702
27.262.3.8 SetFileName . . . 703
27.262.3.9 SetStream . . . 703
27.263 gdcm::StreamImageWriter Class Reference . . . 703
27.263.1 Detailed Description . . . 705
27.263.2 Constructor & Destructor Documentation . . . 705
27.263.2.1 StreamImageWriter . . . 705
27.263.2.2 ~StreamImageWriter . . . 705
27.263.3 Member Function Documentation . . . 706
27.263.3.1 CanWriteFile . . . 706
27.263.3.2 DefinePixelExtent . . . 706
27.263.3.3 DefineProperBufferLength . . . 706
27.263.3.4 SetFile . . . 706
27.263.3.5 SetFileName . . . 706
27.263.3.6 SetStream . . . 706
27.263.3.7 Write . . . 707
27.263.3.8 WriteImageInformation . . . 707
27.263.3.9 WriteImageSubregionRAW . . . 707
27.263.3.10 WriteRawHeader . . . 707
27.263.4 Member Data Documentation . . . 707
27.263.4.1 mElementOffsets . . . 707
27.263.4.2 mElementOffsets1 . . . 708
27.263.4.3 mspFile . . . 708
27.263.4.4 mWriter . . . 708
27.263.4.5 mXMax . . . 708
27.263.4.6 mXMin . . . 708
27.263.4.7 mYMax . . . 708
27.263.4.8 mYMin . . . 708
27.263.4.9 mZMax . . . 708
27.263.4.10 mZMin . . . 708

27.264gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference . . . 708
27.264.1Detailed Description . . . 710
27.264.2Member Typedef Documentation . . . 710
27.264.2.1const_iterator . . . 710
27.264.2.2const_reference . . . 710
27.264.2.3const_reverse_iterator . . . 710
27.264.2.4difference_type . . . 710
27.264.2.5iterator . . . 710
27.264.2.6pointer . . . 710
27.264.2.7reference . . . 710
27.264.2.8reverse_iterator . . . 710
27.264.2.9size_type . . . 710
27.264.2.10value_type . . . 711
27.264.3Constructor & Destructor Documentation . . . 711
27.264.3.1String . . . 711
27.264.3.2String . . . 711
27.264.3.3String . . . 711
27.264.3.4String . . . 711
27.264.4Member Function Documentation . . . 711
27.264.4.1IsValid . . . 711
27.264.4.2operator const char * . . . 711
27.264.4.3Trim . . . 711
27.264.4.4Trim . . . 711
27.264.4.5Truncate . . . 711
27.265gdcm::StringFilter Class Reference . . . 712
27.265.1Detailed Description . . . 712
27.265.2Constructor & Destructor Documentation . . . 713
27.265.2.1StringFilter . . . 713
27.265.2.2~StringFilter . . . 713
27.265.3Member Function Documentation . . . 713
27.265.3.1ExecuteQuery . . . 713
27.265.3.2ExecuteQuery . . . 713
27.265.3.3FromString . . . 713
27.265.3.4FromString . . . 713
27.265.3.5GetFile . . . 713
27.265.3.6GetFile . . . 713
27.265.3.7SetDicts . . . 713

27.265.3.8 SetFile . . . 713
27.265.3.9 ToString . . . 713
27.265.3.10 ToString . . . 713
27.265.3.11 ToStringPair . . . 714
27.265.3.12 ToStringPair . . . 714
27.265.3.13 ToStringPair . . . 714
27.265.3.14 UseDictAlways . . . 714
27.266 gdcm::Study Class Reference . . . 714
27.266.1 Detailed Description . . . 714
27.266.2 Constructor & Destructor Documentation . . . 714
27.266.2.1 Study . . . 714
27.267 gdcm::Subject Class Reference . . . 715
27.267.1 Detailed Description . . . 716
27.267.2 Constructor & Destructor Documentation . . . 716
27.267.2.1 Subject . . . 716
27.267.2.2 ~Subject . . . 716
27.267.3 Member Function Documentation . . . 716
27.267.3.1 AddObserver . . . 716
27.267.3.2 AddObserver . . . 716
27.267.3.3 GetCommand . . . 716
27.267.3.4 HasObserver . . . 716
27.267.3.5 InvokeEvent . . . 716
27.267.3.6 InvokeEvent . . . 717
27.267.3.7 RemoveAllObservers . . . 717
27.267.3.8 RemoveObserver . . . 717
27.268 gdcm::Surface Class Reference . . . 717
27.268.1 Detailed Description . . . 720
27.268.2 Member Enumeration Documentation . . . 720
27.268.2.1 STATES . . . 720
27.268.2.2 VIEWType . . . 720
27.268.3 Constructor & Destructor Documentation . . . 720
27.268.3.1 Surface . . . 720
27.268.3.2 ~Surface . . . 720
27.268.4 Member Function Documentation . . . 721
27.268.4.1 GetAlgorithmFamily . . . 721
27.268.4.2 GetAlgorithmFamily . . . 721
27.268.4.3 GetAlgorithmName . . . 721

27.268.4.4 GetAlgorithmVersion . . . 721
27.268.4.5 GetAxisOfRotation . . . 721
27.268.4.6 GetCenterOfRotation . . . 721
27.268.4.7 GetFiniteVolume . . . 721
27.268.4.8 GetManifold . . . 721
27.268.4.9 GetMaximumPointDistance . . . 721
27.268.4.10 GetMeanPointDistance . . . 721
27.268.4.11 GetMeshPrimitive . . . 721
27.268.4.12 GetMeshPrimitive . . . 721
27.268.4.13 GetNumberOfSurfacePoints . . . 721
27.268.4.14 GetNumberOfVectors . . . 721
27.268.4.15 GetPointCoordinatesData . . . 721
27.268.4.16 GetPointCoordinatesData . . . 721
27.268.4.17 GetPointPositionAccuracy . . . 721
27.268.4.18 GetPointsBoundingBoxCoordinates . . . 721
27.268.4.19 GetProcessingAlgorithm . . . 722
27.268.4.20 GetProcessingAlgorithm . . . 722
27.268.4.21 GetRecommendedDisplayCIELabValue . . . 722
27.268.4.22 GetRecommendedDisplayCIELabValue . . . 722
27.268.4.23 GetRecommendedDisplayGrayscaleValue . . . 722
27.268.4.24 GetRecommendedPresentationOpacity . . . 722
27.268.4.25 GetRecommendedPresentationType . . . 722
27.268.4.26 GetSTATES . . . 722
27.268.4.27 GetSTATESString . . . 722
27.268.4.28 GetSurfaceComments . . . 722
27.268.4.29 GetSurfaceNumber . . . 722
27.268.4.30 GetSurfaceProcessing . . . 722
27.268.4.31 GetSurfaceProcessingDescription . . . 722
27.268.4.32 GetSurfaceProcessingRatio . . . 722
27.268.4.33 GetVectorAccuracy . . . 722
27.268.4.34 GetVectorCoordinateData . . . 722
27.268.4.35 GetVectorCoordinateData . . . 722
27.268.4.36 GetVectorDimensionality . . . 722
27.268.4.37 GetVIEWType . . . 722
27.268.4.38 GetVIEWTypeString . . . 722
27.268.4.39 SetAlgorithmFamily . . . 722
27.268.4.40 SetAlgorithmName . . . 722

27.268.4.43 SetAlgorithmVersion . . . 722
27.268.4.44 SetAxisOfRotation . . . 722
27.268.4.45 SetCenterOfRotation . . . 723
27.268.4.46 SetFiniteVolume . . . 723
27.268.4.47 SetManifold . . . 723
27.268.4.48 SetMaximumPointDistance . . . 723
27.268.4.49 SetMeanPointDistance . . . 723
27.268.4.50 SetMeshPrimitive . . . 723
27.268.4.51 SetNumberOfSurfacePoints . . . 723
27.268.4.52 SetNumberOfVectors . . . 723
27.268.4.53 SetPointCoordinatesData . . . 723
27.268.4.54 SetPointPositionAccuracy . . . 723
27.268.4.55 SetPointsBoundingBoxCoordinates . . . 723
27.268.4.56 SetProcessingAlgorithm . . . 723
27.268.4.57 SetRecommendedDisplayCIELabValue . . . 723
27.268.4.58 SetRecommendedDisplayCIELabValue . . . 723
27.268.4.59 SetRecommendedDisplayCIELabValue . . . 723
27.268.4.60 SetRecommendedDisplayGrayscaleValue . . . 723
27.268.4.61 SetRecommendedPresentationOpacity . . . 723
27.268.4.62 SetRecommendedPresentationType . . . 723
27.268.4.63 SetSurfaceComments . . . 723
27.268.4.64 SetSurfaceNumber . . . 723
27.268.4.65 SetSurfaceProcessing . . . 723
27.268.4.66 SetSurfaceProcessingDescription . . . 723
27.268.4.67 SetSurfaceProcessingRatio . . . 723
27.268.4.68 SetVectorAccuracy . . . 723
27.268.4.69 SetVectorCoordinateData . . . 723
27.268.4.70 SetVectorDimensionality . . . 724
27.269 gdcm::SurfaceHelper Class Reference . . . 724
27.269.1 Detailed Description . . . 724
27.269.2 Member Typedef Documentation . . . 724
27.269.2.1 ColorArray . . . 724
27.269.3 Member Function Documentation . . . 724
27.269.3.1 RecommendedDisplayCIELabToRGB . . . 724
27.269.3.2 RecommendedDisplayCIELabToRGB . . . 725
27.269.3.3 RGBToRecommendedDisplayCIELab . . . 725
27.269.3.4 RGBToRecommendedDisplayGrayscale . . . 726

27.270gdcm::SurfaceReader Class Reference . . . 726
27.270.1Detailed Description . . . 727
27.270.2Constructor & Destructor Documentation . . . 728
27.270.2.1SurfaceReader . . . 728
27.270.2.2∼SurfaceReader . . . 728
27.270.3Member Function Documentation . . . 728
27.270.3.1GetNumberOfSurfaces . . . 728
27.270.3.2Read . . . 728
27.270.3.3ReadPointMacro . . . 728
27.270.3.4ReadSurface . . . 728
27.270.3.5ReadSurfaces . . . 728
27.271gdcm::SurfaceWriter Class Reference . . . 728
27.271.1Detailed Description . . . 730
27.271.2Constructor & Destructor Documentation . . . 730
27.271.2.1SurfaceWriter . . . 730
27.271.2.2∼SurfaceWriter . . . 730
27.271.3Member Function Documentation . . . 730
27.271.3.1ComputeNumberOfSurfaces . . . 730
27.271.3.2GetNumberOfSurfaces . . . 730
27.271.3.3PrepareWrite . . . 730
27.271.3.4PrepareWritePointMacro . . . 730
27.271.3.5SetNumberOfSurfaces . . . 730
27.271.3.6Write . . . 730
27.271.4Member Data Documentation . . . 730
27.271.4.1NumberOfSurfaces . . . 730
27.272gdcm::SwapCode Class Reference . . . 730
27.272.1Detailed Description . . . 731
27.272.2Member Enumeration Documentation . . . 731
27.272.2.1SwapCodeType . . . 731
27.272.3Constructor & Destructor Documentation . . . 732
27.272.3.1SwapCode . . . 732
27.272.4Member Function Documentation . . . 732
27.272.4.1GetIndex . . . 732
27.272.4.2GetSwapCodeString . . . 732
27.272.4.3operator SwapCode::SwapCodeType . . . 732
27.272.5Friends And Related Function Documentation . . . 732
27.272.5.1operator<< . . . 732

27.273 gdcm::SwapperDoOp Class Reference . . . 732
27.273.1 Member Function Documentation . . . 732
27.273.1.1 Swap . . . 732
27.273.1.2 SwapArray . . . 732
27.274 gdcm::SwapperNoOp Class Reference . . . 733
27.274.1 Detailed Description . . . 733
27.274.2 Member Function Documentation . . . 733
27.274.2.1 Swap . . . 733
27.274.2.2 SwapArray . . . 733
27.275 gdcm::System Class Reference . . . 733
27.275.1 Detailed Description . . . 734
27.275.2 Member Function Documentation . . . 734
27.275.2.1 DeleteDirectory . . . 734
27.275.2.2 EncodeBytes . . . 734
27.275.2.3 FileExists . . . 735
27.275.2.4 FileIsDirectory . . . 735
27.275.2.5 FileIsSymlink . . . 735
27.275.2.6 FileSize . . . 735
27.275.2.7 FileTime . . . 735
27.275.2.8 FormatDateTime . . . 735
27.275.2.9 GetCurrentDateTime . . . 735
27.275.2.10 GetCurrentModuleFileName . . . 736
27.275.2.11 GetCurrentProcessFileName . . . 736
27.275.2.12 GetCurrentResourcesDirectory . . . 736
27.275.2.13 GetCWD . . . 736
27.275.2.14 GetHostName . . . 736
27.275.2.15 GetLastSystemError . . . 736
27.275.2.16 GetLocaleCharset . . . 736
27.275.2.17 GetPermissions . . . 736
27.275.2.18 GetTimezoneOffsetFromUTC . . . 736
27.275.2.19 MakeDirectory . . . 736
27.275.2.20 ParseDateTime . . . 737
27.275.2.21 ParseDateTime . . . 737
27.275.2.22 RemoveFile . . . 737
27.275.2.23 SetPermissions . . . 737
27.275.2.24 StrCaseCmp . . . 737
27.275.2.25 StrNCaseCmp . . . 737

27.275.2.26 StrSep . . . 737
27.275.2.27 StrTokR . . . 737
27.276 gdcm::Table Class Reference . . . 737
27.276.1 Detailed Description . . . 738
27.276.2 Member Typedef Documentation . . . 738
27.276.2.1 MapTableEntry . . . 738
27.276.3 Constructor & Destructor Documentation . . . 738
27.276.3.1 Table . . . 738
27.276.3.2 ~Table . . . 738
27.276.4 Member Function Documentation . . . 738
27.276.4.1 GetTableEntry . . . 738
27.276.4.2 InsertEntry . . . 738
27.276.5 Friends And Related Function Documentation . . . 738
27.276.5.1 operator<< . . . 738
27.277 gdcm::TableEntry Class Reference . . . 738
27.277.1 Detailed Description . . . 739
27.277.2 Constructor & Destructor Documentation . . . 739
27.277.2.1 TableEntry . . . 739
27.277.2.2 ~TableEntry . . . 739
27.278 gdcm::TableReader Class Reference . . . 739
27.278.1 Detailed Description . . . 740
27.278.2 Constructor & Destructor Documentation . . . 740
27.278.2.1 TableReader . . . 740
27.278.2.2 ~TableReader . . . 740
27.278.3 Member Function Documentation . . . 740
27.278.3.1 CharacterDataHandler . . . 740
27.278.3.2 EndElement . . . 740
27.278.3.3 GetDefs . . . 740
27.278.3.4 GetFilename . . . 740
27.278.3.5 HandleIOD . . . 740
27.278.3.6 HandleIODEntry . . . 740
27.278.3.7 HandleMacro . . . 740
27.278.3.8 HandleMacroEntry . . . 740
27.278.3.9 HandleMacroEntryDescription . . . 741
27.278.3.10 HandleModule . . . 741
27.278.3.11 HandleModuleEntry . . . 741
27.278.3.12 HandleModuleEntryDescription . . . 741

27.278.3.13 HandleModuleInclude . . . 741
27.278.3.14 Read . . . 741
27.278.3.15 SetFilename . . . 741
27.278.3.16 StartElement . . . 741
27.279 gdcm::network::TableRow Class Reference . . . 741
27.279.1 Constructor & Destructor Documentation . . . 742
27.279.1.1 TableRow . . . 742
27.279.1.2 ~TableRow . . . 742
27.279.2 Member Data Documentation . . . 742
27.279.2.1 transitions . . . 742
27.280 gdcm::Tag Class Reference . . . 742
27.280.1 Detailed Description . . . 744
27.280.2 Constructor & Destructor Documentation . . . 744
27.280.2.1 Tag . . . 744
27.280.2.2 Tag . . . 744
27.280.2.3 Tag . . . 745
27.280.3 Member Function Documentation . . . 745
27.280.3.1 GetElement . . . 745
27.280.3.2 GetElementTag . . . 745
27.280.3.3 GetGroup . . . 745
27.280.3.4 GetLength . . . 745
27.280.3.5 GetPrivateCreator . . . 745
27.280.3.6 IsGroupLength . . . 745
27.280.3.7 IsGroupXX . . . 746
27.280.3.8 IsIllegal . . . 746
27.280.3.9 IsPrivate . . . 746
27.280.3.10 IsPrivateCreator . . . 746
27.280.3.11 IsPublic . . . 746
27.280.3.12 operator"!= . . . 746
27.280.3.13 operator< . . . 746
27.280.3.14 operator<= . . . 746
27.280.3.15 operator= . . . 746
27.280.3.16 operator== . . . 747
27.280.3.17 operator[] . . . 747
27.280.3.18 operator[] . . . 747
27.280.3.19 PrintAsContinuousString . . . 747
27.280.3.20 PrintAsContinuousUpperCaseString . . . 747

27.280.3.21 PrintAsPipeSeparatedString . . . 747
27.280.3.22 Read . . . 747
27.280.3.23 ReadFromCommaSeparatedString . . . 747
27.280.3.24 ReadFromContinuousString . . . 747
27.280.3.25 ReadFromPipeSeparatedString . . . 748
27.280.3.26 SetElement . . . 748
27.280.3.27 SetElementTag . . . 748
27.280.3.28 SetElementTag . . . 748
27.280.3.29 SetGroup . . . 748
27.280.3.30 SetPrivateCreator . . . 748
27.280.3.31 Write . . . 748
27.280.4 Friends And Related Function Documentation . . . 749
27.280.4.1 operator<< . . . 749
27.280.4.2 operator>> . . . 749
27.280.5 Member Data Documentation . . . 749
27.280.5.1 bytes . . . 749
27.280.5.2 tag . . . 749
27.280.5.3 tags . . . 749
27.281 gdcm::TagPath Class Reference . . . 749
27.281.1 Detailed Description . . . 749
27.281.2 Constructor & Destructor Documentation . . . 750
27.281.2.1 TagPath . . . 750
27.281.2.2 ~TagPath . . . 750
27.281.3 Member Function Documentation . . . 750
27.281.3.1 ConstructFromString . . . 750
27.281.3.2 ConstructFromTagList . . . 750
27.281.3.3 IsValid . . . 750
27.281.3.4 Print . . . 750
27.281.3.5 Push . . . 750
27.281.3.6 Push . . . 750
27.282 gdcm::Testing Class Reference . . . 750
27.282.1 Detailed Description . . . 751
27.282.2 Member Typedef Documentation . . . 751
27.282.2.1 MD5DataImagesType . . . 751
27.282.2.2 MediaStorageDataFilesType . . . 752
27.282.3 Constructor & Destructor Documentation . . . 752
27.282.3.1 Testing . . . 752

27.282.3.2 ~Testing . . . 752
27.282.4 Member Function Documentation . . . 752
27.282.4.1 ComputeFileMD5 . . . 752
27.282.4.2 ComputeMD5 . . . 752
27.282.4.3 GetDataExtraRoot . . . 752
27.282.4.4 GetDataRoot . . . 752
27.282.4.5 GetFileName . . . 752
27.282.4.6 GetFileNames . . . 753
27.282.4.7 GetLossyFlagFromFile . . . 753
27.282.4.8 GetMD5DataImage . . . 753
27.282.4.9 GetMD5DataImages . . . 753
27.282.4.10 GetMD5FromBrokenFile . . . 753
27.282.4.11 GetMD5FromFile . . . 753
27.282.4.12 GetMediaStorageDataFile . . . 753
27.282.4.13 GetMediaStorageDataFiles . . . 753
27.282.4.14 GetMediaStorageFromFile . . . 753
27.282.4.15 GetNumberOfFileNames . . . 753
27.282.4.16 GetNumberOfMD5DataImages . . . 753
27.282.4.17 GetNumberOfMediaStorageDataFiles . . . 753
27.282.4.18 GetPixelSpacingDataRoot . . . 753
27.282.4.19 GetSelectedPrivateGroupOffsetFromFile . . . 754
27.282.4.20 GetSelectedTagsOffsetFromFile . . . 754
27.282.4.21 GetSourceDirectory . . . 754
27.282.4.22 GetStreamOffsetFromFile . . . 754
27.282.4.23 GetTempDirectory . . . 754
27.282.4.24 GetTempDirectoryW . . . 754
27.282.4.25 GetTempFilename . . . 754
27.282.4.26 GetTempFilenameW . . . 754
27.282.4.27 Print . . . 755
27.283 gdcm::Trace Class Reference . . . 755
27.283.1 Detailed Description . . . 756
27.283.2 Constructor & Destructor Documentation . . . 756
27.283.2.1 Trace . . . 756
27.283.2.2 ~Trace . . . 756
27.283.3 Member Function Documentation . . . 756
27.283.3.1 DebugOff . . . 756
27.283.3.2 DebugOn . . . 756

27.283.3.3 ErrorOff . . . 756
27.283.3.4 ErrorOn . . . 756
27.283.3.5 GetDebugFlag . . . 756
27.283.3.6 GetDebugStream . . . 756
27.283.3.7 GetErrorFlag . . . 756
27.283.3.8 GetErrorStream . . . 757
27.283.3.9 GetStream . . . 757
27.283.3.10 GetWarningFlag . . . 757
27.283.3.11 GetWarningStream . . . 757
27.283.3.12 SetDebug . . . 757
27.283.3.13 SetDebugStream . . . 757
27.283.3.14 SetError . . . 757
27.283.3.15 SetErrorStream . . . 757
27.283.3.16 SetStream . . . 757
27.283.3.17 SetStreamToFile . . . 757
27.283.3.18 SetWarning . . . 757
27.283.3.19 SetWarningStream . . . 758
27.283.3.20 WarningOff . . . 758
27.283.3.21 WarningOn . . . 758
27.284 gdcm::TransferSyntax Class Reference . . . 758
27.284.1 Detailed Description . . . 760
27.284.2 Member Enumeration Documentation . . . 760
27.284.2.1 NegociatedType . . . 760
27.284.2.2 TSType . . . 760
27.284.3 Constructor & Destructor Documentation . . . 761
27.284.3.1 TransferSyntax . . . 761
27.284.4 Member Function Documentation . . . 761
27.284.4.1 CanStoreLossy . . . 761
27.284.4.2 GetNegociatedType . . . 761
27.284.4.3 GetString . . . 761
27.284.4.4 GetSwapCode . . . 761
27.284.4.5 GetTSString . . . 761
27.284.4.6 GetTSType . . . 761
27.284.4.7 IsEncapsulated . . . 761
27.284.4.8 IsEncoded . . . 762
27.284.4.9 IsExplicit . . . 762
27.284.4.10 IsImplicit . . . 762

27.284.4.1 IsLossless . . . 762
27.284.4.2 IsLossy . . . 762
27.284.4.3 IsValid . . . 762
27.284.4.4 operator TSType . . . 762
27.284.5 Friends And Related Function Documentation . . . 762
27.284.5.1 operator<< . . . 762
27.285 gdcm::network::TransferSyntaxSub Class Reference . . . 762
27.285.1 Detailed Description . . . 762
27.285.2 Constructor & Destructor Documentation . . . 763
27.285.2.1 TransferSyntaxSub . . . 763
27.285.3 Member Function Documentation . . . 763
27.285.3.1 GetName . . . 763
27.285.3.2 operator== . . . 763
27.285.3.3 Print . . . 763
27.285.3.4 Read . . . 763
27.285.3.5 SetName . . . 763
27.285.3.6 SetNameFromUID . . . 763
27.285.3.7 Size . . . 763
27.285.3.8 Write . . . 763
27.286 gdcm::network::Transition Struct Reference . . . 763
27.286.1 Constructor & Destructor Documentation . . . 764
27.286.1.1 Transition . . . 764
27.286.1.2 ~Transition . . . 764
27.286.1.3 Transition . . . 764
27.286.2 Member Function Documentation . . . 764
27.286.2.1 MakeNew . . . 764
27.286.3 Member Data Documentation . . . 764
27.286.3.1 mAction . . . 764
27.286.3.2 mEnd . . . 764
27.287 gdcm::Type Class Reference . . . 765
27.287.1 Detailed Description . . . 765
27.287.2 Member Enumeration Documentation . . . 766
27.287.2.1 TypeType . . . 766
27.287.3 Constructor & Destructor Documentation . . . 766
27.287.3.1 Type . . . 766
27.287.4 Member Function Documentation . . . 766
27.287.4.1 GetTypeString . . . 766

27.287.4.2 GetTypeType . . . 766
27.287.4.3 operator TypeType . . . 766
27.287.5 Friends And Related Function Documentation . . . 766
27.287.5.1 operator<< . . . 766
27.288 gdcm::UI Struct Reference . . . 766
27.288.1 Friends And Related Function Documentation . . . 767
27.288.1.1 operator<< . . . 767
27.288.2 Member Data Documentation . . . 767
27.288.2.1 Internal . . . 767
27.289 gdcm::UIDGenerator Class Reference . . . 767
27.289.1 Detailed Description . . . 767
27.289.2 Constructor & Destructor Documentation . . . 768
27.289.2.1 UIDGenerator . . . 768
27.289.3 Member Function Documentation . . . 768
27.289.3.1 Generate . . . 768
27.289.3.2 GenerateUUID . . . 768
27.289.3.3 GetGDCMUID . . . 768
27.289.3.4 GetRoot . . . 768
27.289.3.5 IsValid . . . 768
27.289.3.6 SetRoot . . . 769
27.290 gdcm::UIDs Class Reference . . . 769
27.290.1 Detailed Description . . . 773
27.290.2 Member Typedef Documentation . . . 773
27.290.2.1 TransferSyntaxStringsType . . . 773
27.290.3 Member Enumeration Documentation . . . 773
27.290.3.1 TSName . . . 773
27.290.3.2 TSType . . . 780
27.290.4 Member Function Documentation . . . 786
27.290.4.1 GetName . . . 786
27.290.4.2 GetNumberOfTransferSyntaxStrings . . . 787
27.290.4.3 GetString . . . 787
27.290.4.4 GetTransferSyntaxString . . . 787
27.290.4.5 GetTransferSyntaxStrings . . . 787
27.290.4.6 GetUIDName . . . 787
27.290.4.7 GetUIDString . . . 787
27.290.4.8 operator TSType . . . 787
27.290.4.9 SetFromUID . . . 787

27.291 gdcm::network::ULAction Class Reference . . . 787
27.291.1 Detailed Description . . . 789
27.291.2 Constructor & Destructor Documentation . . . 789
27.291.2.1 ULAction . . . 789
27.291.2.2 ~ULAction . . . 789
27.291.3 Member Function Documentation . . . 789
27.291.3.1 PerformAction . . . 789
27.292 gdcm::network::ULActionAA1 Class Reference . . . 790
27.292.1 Member Function Documentation . . . 790
27.292.1.1 PerformAction . . . 790
27.293 gdcm::network::ULActionAA2 Class Reference . . . 791
27.293.1 Member Function Documentation . . . 791
27.293.1.1 PerformAction . . . 791
27.294 gdcm::network::ULActionAA3 Class Reference . . . 792
27.294.1 Member Function Documentation . . . 792
27.294.1.1 PerformAction . . . 792
27.295 gdcm::network::ULActionAA4 Class Reference . . . 793
27.295.1 Member Function Documentation . . . 793
27.295.1.1 PerformAction . . . 793
27.296 gdcm::network::ULActionAA5 Class Reference . . . 794
27.296.1 Member Function Documentation . . . 794
27.296.1.1 PerformAction . . . 794
27.297 gdcm::network::ULActionAA6 Class Reference . . . 795
27.297.1 Member Function Documentation . . . 795
27.297.1.1 PerformAction . . . 795
27.298 gdcm::network::ULActionAA7 Class Reference . . . 796
27.298.1 Member Function Documentation . . . 796
27.298.1.1 PerformAction . . . 796
27.299 gdcm::network::ULActionAA8 Class Reference . . . 797
27.299.1 Member Function Documentation . . . 797
27.299.1.1 PerformAction . . . 797
27.300 gdcm::network::ULActionAE1 Class Reference . . . 798
27.300.1 Member Function Documentation . . . 798
27.300.1.1 PerformAction . . . 798
27.301 gdcm::network::ULActionAE2 Class Reference . . . 799
27.301.1 Member Function Documentation . . . 799
27.301.1.1 PerformAction . . . 799

27.302gdcm::network::ULActionAE3 Class Reference . . . 800
27.302.1Member Function Documentation . . . 800
27.302.1.1PerformAction . . . 800
27.303gdcm::network::ULActionAE4 Class Reference . . . 801
27.303.1Member Function Documentation . . . 801
27.303.1.1PerformAction . . . 801
27.304gdcm::network::ULActionAE5 Class Reference . . . 802
27.304.1Member Function Documentation . . . 802
27.304.1.1PerformAction . . . 802
27.305gdcm::network::ULActionAE6 Class Reference . . . 803
27.305.1Member Function Documentation . . . 803
27.305.1.1PerformAction . . . 803
27.306gdcm::network::ULActionAE7 Class Reference . . . 804
27.306.1Member Function Documentation . . . 804
27.306.1.1PerformAction . . . 804
27.307gdcm::network::ULActionAE8 Class Reference . . . 805
27.307.1Member Function Documentation . . . 805
27.307.1.1PerformAction . . . 805
27.308gdcm::network::ULActionAR1 Class Reference . . . 806
27.308.1Member Function Documentation . . . 806
27.308.1.1PerformAction . . . 806
27.309gdcm::network::ULActionAR10 Class Reference . . . 807
27.309.1Member Function Documentation . . . 807
27.309.1.1PerformAction . . . 807
27.310gdcm::network::ULActionAR2 Class Reference . . . 808
27.310.1Member Function Documentation . . . 808
27.310.1.1PerformAction . . . 808
27.311gdcm::network::ULActionAR3 Class Reference . . . 809
27.311.1Member Function Documentation . . . 809
27.311.1.1PerformAction . . . 809
27.312gdcm::network::ULActionAR4 Class Reference . . . 810
27.312.1Member Function Documentation . . . 810
27.312.1.1PerformAction . . . 810
27.313gdcm::network::ULActionAR5 Class Reference . . . 811
27.313.1Member Function Documentation . . . 811
27.313.1.1PerformAction . . . 811
27.314gdcm::network::ULActionAR6 Class Reference . . . 812

27.314.1 Member Function Documentation . . . 812
27.314.1.1 PerformAction . . . 812
27.315 gdcm::network::ULActionAR7 Class Reference . . . 813
27.315.1 Member Function Documentation . . . 813
27.315.1.1 PerformAction . . . 813
27.316 gdcm::network::ULActionAR8 Class Reference . . . 814
27.316.1 Member Function Documentation . . . 814
27.316.1.1 PerformAction . . . 814
27.317 gdcm::network::ULActionAR9 Class Reference . . . 815
27.317.1 Member Function Documentation . . . 815
27.317.1.1 PerformAction . . . 815
27.318 gdcm::network::ULActionDT1 Class Reference . . . 816
27.318.1 Member Function Documentation . . . 816
27.318.1.1 PerformAction . . . 816
27.319 gdcm::network::ULActionDT2 Class Reference . . . 817
27.319.1 Member Function Documentation . . . 817
27.319.1.1 PerformAction . . . 817
27.320 gdcm::network::ULBasicCallback Class Reference . . . 818
27.320.1 Detailed Description . . . 819
27.320.2 Constructor & Destructor Documentation . . . 819
27.320.2.1 ULBasicCallback . . . 819
27.320.2.2 ~ULBasicCallback . . . 819
27.320.3 Member Function Documentation . . . 819
27.320.3.1 GetDataSets . . . 819
27.320.3.2 GetResponses . . . 819
27.320.3.3 HandleDataSet . . . 819
27.320.3.4 HandleResponse . . . 819
27.321 gdcm::network::ULConnection Class Reference . . . 819
27.321.1 Detailed Description . . . 820
27.321.2 Constructor & Destructor Documentation . . . 820
27.321.2.1 ULConnection . . . 820
27.321.2.2 ~ULConnection . . . 821
27.321.3 Member Function Documentation . . . 821
27.321.3.1 AddAcceptedPresentationContext . . . 821
27.321.3.2 FindContext . . . 821
27.321.3.3 GetAcceptedPresentationContexts . . . 821
27.321.3.4 GetAcceptedPresentationContexts . . . 821

27.321.3.5 GetConnectionInfo . . . 821
27.321.3.6 GetMaxPDUSize . . . 821
27.321.3.7 GetPresentationContextACByID . . . 821
27.321.3.8 GetPresentationContextIDFromPresentationContext . . . 821
27.321.3.9 GetPresentationContextRQByID . . . 821
27.321.3.10 GetPresentationContexts . . . 821
27.321.3.11 GetProtocol . . . 821
27.321.3.12 GetState . . . 821
27.321.3.13 GetTimer . . . 821
27.321.3.14 InitializeConnection . . . 821
27.321.3.15 InitializeIncomingConnection . . . 821
27.321.3.16 SetMaxPDUSize . . . 821
27.321.3.17 SetPresentationContexts . . . 821
27.321.3.18 SetPresentationContexts . . . 821
27.321.3.19 SetState . . . 822
27.321.3.20 StopProtocol . . . 822
27.321.4 Friends And Related Function Documentation . . . 822
27.321.4.1 ULActionAE6 . . . 822
27.321.4.2 ULConnectionManager . . . 822
27.322 gdcm::network::ULConnectionCallback Class Reference . . . 822
27.322.1 Detailed Description . . . 823
27.322.2 Constructor & Destructor Documentation . . . 823
27.322.2.1 ULConnectionCallback . . . 823
27.322.2.2 ~ULConnectionCallback . . . 823
27.322.3 Member Function Documentation . . . 823
27.322.3.1 DataSetHandled . . . 823
27.322.3.2 DataSetHandles . . . 823
27.322.3.3 HandleDataSet . . . 823
27.322.3.4 HandleResponse . . . 823
27.322.3.5 ResetHandledDataSet . . . 823
27.322.3.6 SetImplicitFlag . . . 823
27.322.4 Member Data Documentation . . . 823
27.322.4.1 mImplicit . . . 823
27.323 gdcm::network::ULConnectionInfo Class Reference . . . 824
27.323.1 Detailed Description . . . 824
27.323.2 Constructor & Destructor Documentation . . . 824
27.323.2.1 ULConnectionInfo . . . 824

27.323.3 Member Function Documentation . . . 824
27.323.3.1 GetCalledAETitle . . . 824
27.323.3.2 GetCalledComputerName . . . 824
27.323.3.3 GetCalledIPAddress . . . 824
27.323.3.4 GetCalledIPPort . . . 824
27.323.3.5 GetCallingAETitle . . . 824
27.323.3.6 GetMaxPDULength . . . 824
27.323.3.7 Initialize . . . 824
27.323.3.8 SetMaxPDULength . . . 825
27.324 gdcm::network::ULConnectionManager Class Reference . . . 825
27.324.1 Detailed Description . . . 827
27.324.2 Constructor & Destructor Documentation . . . 827
27.324.2.1 ULConnectionManager . . . 827
27.324.2.2 ~ULConnectionManager . . . 827
27.324.3 Member Function Documentation . . . 827
27.324.3.1 BreakConnection . . . 827
27.324.3.2 BreakConnectionNow . . . 827
27.324.3.3 EstablishConnection . . . 827
27.324.3.4 EstablishConnectionMove . . . 827
27.324.3.5 SendEcho . . . 827
27.324.3.6 SendFind . . . 827
27.324.3.7 SendFind . . . 827
27.324.3.8 SendMove . . . 827
27.324.3.9 SendMove . . . 827
27.324.3.10 SendStore . . . 827
27.324.3.11 SendStore . . . 828
27.325 gdcm::network::ULEvent Class Reference . . . 828
27.325.1 Detailed Description . . . 828
27.325.2 Constructor & Destructor Documentation . . . 828
27.325.2.1 ULEvent . . . 828
27.325.2.2 ULEvent . . . 828
27.325.2.3 ~ULEvent . . . 828
27.325.3 Member Function Documentation . . . 828
27.325.3.1 GetEvent . . . 828
27.325.3.2 GetPDUs . . . 828
27.325.3.3 SetEvent . . . 828
27.325.3.4 SetPDU . . . 829

27.326gdcm::network::ULTransitionTable Class Reference . . . 829
27.326.1Detailed Description . . . 829
27.326.2Constructor & Destructor Documentation . . . 829
27.326.2.1ULTransitionTable . . . 829
27.326.3Member Function Documentation . . . 829
27.326.3.1HandleEvent . . . 829
27.326.3.2PrintTable . . . 829
27.327gdcm::network::ULWritingCallback Class Reference . . . 829
27.327.1Constructor & Destructor Documentation . . . 830
27.327.1.1ULWritingCallback . . . 831
27.327.1.2~ULWritingCallback . . . 831
27.327.2Member Function Documentation . . . 831
27.327.2.1HandleDataSet . . . 831
27.327.2.2HandleResponse . . . 831
27.327.2.3SetDirectory . . . 831
27.328gdcm::UNExplicitDataElement Class Reference . . . 831
27.328.1Detailed Description . . . 832
27.328.2Member Function Documentation . . . 832
27.328.2.1GetLength . . . 832
27.328.2.2Read . . . 833
27.328.2.3ReadPreValue . . . 833
27.328.2.4ReadValue . . . 833
27.328.2.5ReadWithLength . . . 833
27.329gdcm::UNExplicitImplicitDataElement Class Reference . . . 833
27.329.1Detailed Description . . . 834
27.329.2Member Function Documentation . . . 834
27.329.2.1GetLength . . . 834
27.329.2.2Read . . . 835
27.329.2.3ReadPreValue . . . 835
27.329.2.4ReadValue . . . 835
27.330gdcm::Unpacker12Bits Class Reference . . . 835
27.330.1Detailed Description . . . 835
27.330.2Member Function Documentation . . . 835
27.330.2.1Pack . . . 835
27.330.2.2Unpack . . . 835
27.331gdcm::Usage Class Reference . . . 836
27.331.1Detailed Description . . . 836

27.331.2 Member Enumeration Documentation . . . 837
27.331.2.1 UsageType . . . 837
27.331.3 Constructor & Destructor Documentation . . . 837
27.331.3.1 Usage . . . 837
27.331.4 Member Function Documentation . . . 837
27.331.4.1 GetUsageString . . . 837
27.331.4.2 GetUsageType . . . 837
27.331.4.3 operator UsageType . . . 837
27.331.5 Friends And Related Function Documentation . . . 837
27.331.5.1 operator<< . . . 837
27.332 gdcm::UserEvent Class Reference . . . 837
27.333 gdcm::network::UserInformation Class Reference . . . 839
27.333.1 Detailed Description . . . 839
27.333.2 Constructor & Destructor Documentation . . . 839
27.333.2.1 UserInformation . . . 839
27.333.2.2 ~UserInformation . . . 839
27.333.3 Member Function Documentation . . . 839
27.333.3.1 AddRoleSelectionSub . . . 839
27.333.3.2 AddSOPClassExtendedNegociationSub . . . 839
27.333.3.3 GetMaximumLengthSub . . . 839
27.333.3.4 GetMaximumLengthSub . . . 839
27.333.3.5 operator= . . . 839
27.333.3.6 Print . . . 839
27.333.3.7 Read . . . 839
27.333.3.8 Size . . . 840
27.333.3.9 Write . . . 840
27.334 gdcm::UUIDGenerator Class Reference . . . 840
27.334.1 Detailed Description . . . 840
27.334.2 Member Function Documentation . . . 840
27.334.2.1 Generate . . . 840
27.334.2.2 IsValid . . . 840
27.335 gdcm::Validate Class Reference . . . 840
27.335.1 Detailed Description . . . 841
27.335.2 Constructor & Destructor Documentation . . . 841
27.335.2.1 Validate . . . 841
27.335.2.2 ~Validate . . . 841
27.335.3 Member Function Documentation . . . 841

27.335.3.1 GetValidatedFile . . . 841
27.335.3.2 SetFile . . . 842
27.335.3.3 Validation . . . 842
27.335.4 Member Data Documentation . . . 842
27.335.4.1 F . . . 842
27.335.4.2 V . . . 842
27.336 gdcm::Value Class Reference . . . 842
27.336.1 Detailed Description . . . 843
27.336.2 Constructor & Destructor Documentation . . . 843
27.336.2.1 Value . . . 843
27.336.2.2 ~Value . . . 843
27.336.3 Member Function Documentation . . . 843
27.336.3.1 Clear . . . 843
27.336.3.2 GetLength . . . 843
27.336.3.3 operator== . . . 843
27.336.3.4 SetLength . . . 844
27.336.3.5 SetLengthOnly . . . 844
27.336.4 Friends And Related Function Documentation . . . 844
27.336.4.1 DataElement . . . 844
27.337 gdcm::ValueIO< TDE, TSwap, TType > Class Template Reference . . . 844
27.337.1 Detailed Description . . . 844
27.337.2 Member Function Documentation . . . 844
27.337.2.1 Read . . . 844
27.337.2.2 Write . . . 844
27.338 gdcm::Version Class Reference . . . 845
27.338.1 Detailed Description . . . 845
27.338.2 Constructor & Destructor Documentation . . . 845
27.338.2.1 Version . . . 845
27.338.2.2 ~Version . . . 845
27.338.3 Member Function Documentation . . . 845
27.338.3.1 GetBuildVersion . . . 845
27.338.3.2 GetMajorVersion . . . 845
27.338.3.3 GetMinorVersion . . . 845
27.338.3.4 GetVersion . . . 845
27.338.3.5 Print . . . 845
27.338.4 Friends And Related Function Documentation . . . 846
27.338.4.1 operator<< . . . 846

27.339 gdcm::VL Class Reference . . . 846
27.339.1 Detailed Description . . . 847
27.339.2 Member Typedef Documentation . . . 847
27.339.2.1 Type . . . 847
27.339.3 Constructor & Destructor Documentation . . . 847
27.339.3.1 VL . . . 847
27.339.4 Member Function Documentation . . . 847
27.339.4.1 GetLength . . . 847
27.339.4.2 GetVL16Max . . . 847
27.339.4.3 GetVL32Max . . . 847
27.339.4.4 IsOdd . . . 847
27.339.4.5 IsUndefined . . . 847
27.339.4.6 operator uint32_t . . . 847
27.339.4.7 operator++ . . . 847
27.339.4.8 operator++ . . . 847
27.339.4.9 operator+= . . . 847
27.339.4.10 Read . . . 848
27.339.4.11 Read16 . . . 848
27.339.4.12 SetToUndefined . . . 848
27.339.4.13 Write . . . 848
27.339.4.14 Write16 . . . 848
27.339.5 Friends And Related Function Documentation . . . 848
27.339.5.1 operator<< . . . 848
27.340 gdcm::VM Class Reference . . . 848
27.340.1 Detailed Description . . . 850
27.340.2 Member Enumeration Documentation . . . 850
27.340.2.1 VMType . . . 850
27.340.3 Constructor & Destructor Documentation . . . 851
27.340.3.1 VM . . . 851
27.340.4 Member Function Documentation . . . 851
27.340.4.1 Compatible . . . 851
27.340.4.2 GetIndex . . . 851
27.340.4.3 GetLength . . . 851
27.340.4.4 GetNumberOfElementsFromArray . . . 851
27.340.4.5 GetVMString . . . 851
27.340.4.6 GetVMType . . . 852
27.340.4.7 GetVMTypeFromLength . . . 852

27.340.4.8 IsValid . . . 852
27.340.4.9 operator VMType . . . 852
27.340.5 Friends And Related Function Documentation . . . 852
27.340.5.1 operator<< . . . 852
27.341 gdcm::VMToLength< T > Struct Template Reference . . . 852
27.342 gdcm::VR Class Reference . . . 852
27.342.1 Detailed Description . . . 854
27.342.2 Member Enumeration Documentation . . . 854
27.342.2.1 VRType . . . 854
27.342.3 Constructor & Destructor Documentation . . . 856
27.342.3.1 VR . . . 856
27.342.4 Member Function Documentation . . . 856
27.342.4.1 CanDisplay . . . 856
27.342.4.2 Compatible . . . 856
27.342.4.3 GetLength . . . 856
27.342.4.4 GetLength . . . 856
27.342.4.5 GetSize . . . 856
27.342.4.6 GetSizeof . . . 856
27.342.4.7 GetVRString . . . 856
27.342.4.8 GetVRStringFromFile . . . 856
27.342.4.9 GetVRType . . . 856
27.342.4.10 GetVRTypeFromFile . . . 856
27.342.4.11 IsASCII . . . 856
27.342.4.12 IsASCII2 . . . 856
27.342.4.13 IsBinary . . . 856
27.342.4.14 IsBinary2 . . . 856
27.342.4.15 IsDual . . . 856
27.342.4.16 IsSwap . . . 856
27.342.4.17 IsValid . . . 856
27.342.4.18 IsValid . . . 856
27.342.4.19 IsVRFile . . . 856
27.342.4.20 operator VRType . . . 857
27.342.4.21 Read . . . 857
27.342.4.22 Write . . . 857
27.342.5 Friends And Related Function Documentation . . . 857
27.342.5.1 operator<< . . . 857
27.343 gdcm::VR16ExplicitDataElement Class Reference . . . 857

27.343.1 Detailed Description . . . 858
27.343.2 Member Function Documentation . . . 858
27.343.2.1 GetLength . . . 858
27.343.2.2 Read . . . 859
27.343.2.3 ReadPreValue . . . 859
27.343.2.4 ReadValue . . . 859
27.343.2.5 ReadWithLength . . . 859
27.344 gdcm::VRToEncoding< T > Struct Template Reference . . . 859
27.345 gdcm::VRToType< T > Struct Template Reference . . . 859
27.345.1 Detailed Description . . . 859
27.346 gdcm::VRVLSize< T > Singleton Reference . . . 860
27.347 gdcm::VRVLSize< 0 > Class Template Reference . . . 860
27.347.1 Member Function Documentation . . . 860
27.347.1.1 Read . . . 860
27.347.1.2 Write . . . 860
27.348 gdcm::VRVLSize< 1 > Class Template Reference . . . 860
27.348.1 Member Function Documentation . . . 860
27.348.1.1 Read . . . 860
27.348.1.2 Write . . . 860
27.349 vtkGDCMImageReader Class Reference . . . 861
27.349.1 Detailed Description . . . 863
27.349.2 Constructor & Destructor Documentation . . . 863
27.349.2.1 vtkGDCMImageReader . . . 863
27.349.2.2 ~vtkGDCMImageReader . . . 864
27.349.3 Member Function Documentation . . . 864
27.349.3.1 CanReadFile . . . 864
27.349.3.2 ExecuteData . . . 864
27.349.3.3 ExecuteInformation . . . 864
27.349.3.4 FillMedicalImageInformation . . . 864
27.349.3.5 GetDescriptiveName . . . 864
27.349.3.6 GetFileExtensions . . . 864
27.349.3.7 GetIconImage . . . 864
27.349.3.8 GetOverlay . . . 864
27.349.3.9 LoadSingleFile . . . 864
27.349.3.10 New . . . 864
27.349.3.11 PrintSelf . . . 864
27.349.3.12 RequestDataCompat . . . 864

27.349.3.18 RequestInformationCompat . . . 864
27.349.3.19 SetCurve . . . 864
27.349.3.15 SetFileNames . . . 865
27.349.3.16 SetFilePattern . . . 865
27.349.3.17 SetFilePrefix . . . 865
27.349.3.18 SetMedicalImageProperties . . . 865
27.349.3.19 vtkBooleanMacro . . . 865
27.349.3.20 vtkBooleanMacro . . . 865
27.349.3.21 vtkBooleanMacro . . . 865
27.349.3.22 vtkBooleanMacro . . . 865
27.349.3.23 vtkBooleanMacro . . . 865
27.349.3.24 vtkGetMacro . . . 865
27.349.3.25 vtkGetMacro . . . 865
27.349.3.26 vtkGetMacro . . . 865
27.349.3.27 vtkGetMacro . . . 865
27.349.3.28 vtkGetMacro . . . 865
27.349.3.29 vtkGetMacro . . . 865
27.349.3.30 vtkGetMacro . . . 865
27.349.3.31 vtkGetMacro . . . 865
27.349.3.32 vtkGetMacro . . . 865
27.349.3.33 vtkGetMacro . . . 865
27.349.3.34 vtkGetMacro . . . 865
27.349.3.35 vtkGetObjectMacro . . . 865
27.349.3.36 vtkGetObjectMacro . . . 865
27.349.3.37 vtkGetObjectMacro . . . 865
27.349.3.38 vtkGetObjectMacro . . . 866
27.349.3.39 vtkGetStringMacro . . . 866
27.349.3.40 vtkGetStringMacro . . . 866
27.349.3.41 vtkGetVector3Macro . . . 866
27.349.3.42 vtkGetVector6Macro . . . 866
27.349.3.43 vtkSetMacro . . . 866
27.349.3.44 vtkSetMacro . . . 866
27.349.3.45 vtkSetMacro . . . 866
27.349.3.46 vtkSetMacro . . . 866
27.349.3.47 vtkSetVector6Macro . . . 866
27.349.3.48 vtkTypeRevisionMacro . . . 866
27.349.4 Member Data Documentation . . . 866

27.349.4.1 ApplyInverseVideo . . . 866
27.349.4.2 ApplyLookupTable . . . 866
27.349.4.3 ApplyPlanarConfiguration . . . 866
27.349.4.4 ApplyShiftScale . . . 866
27.349.4.5 ApplyYBRToRGB . . . 866
27.349.4.6 Curve . . . 866
27.349.4.7 DirectionCosines . . . 866
27.349.4.8 FileNames . . . 866
27.349.4.9 ForceRescale . . . 866
27.349.4.10 IconDataScalarType . . . 866
27.349.4.11 IconImageDataExtent . . . 866
27.349.4.12 IconNumberOfScalarComponents . . . 866
27.349.4.13 ImageFormat . . . 866
27.349.4.14 ImageOrientationPatient . . . 866
27.349.4.15 ImagePositionPatient . . . 867
27.349.4.16 LoadIconImage . . . 867
27.349.4.17 LoadOverlays . . . 867
27.349.4.18 LossyFlag . . . 867
27.349.4.19 MedicalImageProperties . . . 867
27.349.4.20 NumberOfIconImages . . . 867
27.349.4.21 NumberOfOverlays . . . 867
27.349.4.22 PlanarConfiguration . . . 867
27.349.4.23 Scale . . . 867
27.349.4.24 Shift . . . 867
27.350 vtkGDCMImageReader2 Class Reference . . . 867
27.350.1 Detailed Description . . . 870
27.350.2 Constructor & Destructor Documentation . . . 870
27.350.2.1 vtkGDCMImageReader2 . . . 870
27.350.2.2 ~vtkGDCMImageReader2 . . . 870
27.350.3 Member Function Documentation . . . 870
27.350.3.1 CanReadFile . . . 870
27.350.3.2 FillMedicalImageInformation . . . 870
27.350.3.3 GetDescriptiveName . . . 870
27.350.3.4 GetFileExtensions . . . 870
27.350.3.5 GetIconImage . . . 870
27.350.3.6 GetIconImagePort . . . 870
27.350.3.7 GetOverlay . . . 870

27.350.3.8 GetOverlayPort . . . 870
27.350.3.9 LoadSingleFile . . . 870
27.350.3.10 New . . . 870
27.350.3.11 PrintSelf . . . 870
27.350.3.12 ProcessRequest . . . 870
27.350.3.13 RequestData . . . 870
27.350.3.14 RequestDataCompat . . . 870
27.350.3.15 RequestInformation . . . 870
27.350.3.16 RequestInformationCompat . . . 871
27.350.3.17 SetCurve . . . 871
27.350.3.18 SetFilePattern . . . 871
27.350.3.19 SetFilePrefix . . . 871
27.350.3.20 SetMedicalImageProperties . . . 871
27.350.3.21 vtkBooleanMacro . . . 871
27.350.3.22 vtkBooleanMacro . . . 871
27.350.3.23 vtkBooleanMacro . . . 871
27.350.3.24 vtkBooleanMacro . . . 871
27.350.3.25 vtkBooleanMacro . . . 871
27.350.3.26 vtkGetMacro . . . 871
27.350.3.27 vtkGetMacro . . . 871
27.350.3.28 vtkGetMacro . . . 871
27.350.3.29 vtkGetMacro . . . 871
27.350.3.30 vtkGetMacro . . . 871
27.350.3.31 vtkGetMacro . . . 871
27.350.3.32 vtkGetMacro . . . 871
27.350.3.33 vtkGetMacro . . . 871
27.350.3.34 vtkGetMacro . . . 871
27.350.3.35 vtkGetMacro . . . 871
27.350.3.36 vtkGetMacro . . . 871
27.350.3.37 vtkGetObjectMacro . . . 871
27.350.3.38 vtkGetObjectMacro . . . 871
27.350.3.39 vtkGetStringMacro . . . 871
27.350.3.40 vtkGetStringMacro . . . 871
27.350.3.41 vtkGetVector3Macro . . . 871
27.350.3.42 vtkGetVector6Macro . . . 872
27.350.3.43 vtkSetMacro . . . 872
27.350.3.44 vtkSetMacro . . . 872

27.350.3.45 vtkSetMacro . . . 872
27.350.3.46 vtkSetMacro . . . 872
27.350.3.47 vtkSetVector6Macro . . . 872
27.350.3.48 vtkTypeRevisionMacro . . . 872
27.350.4 Member Data Documentation . . . 872
27.350.4.1 ApplyInverseVideo . . . 872
27.350.4.2 ApplyLookupTable . . . 872
27.350.4.3 ApplyPlanarConfiguration . . . 872
27.350.4.4 ApplyShiftScale . . . 872
27.350.4.5 ApplyYBRToRGB . . . 872
27.350.4.6 Curve . . . 872
27.350.4.7 DirectionCosines . . . 872
27.350.4.8 ForceRescale . . . 872
27.350.4.9 IconDataScalarType . . . 872
27.350.4.10 IconImageDataExtent . . . 872
27.350.4.11 IconNumberOfScalarComponents . . . 872
27.350.4.12 ImageFormat . . . 872
27.350.4.13 ImageOrientationPatient . . . 872
27.350.4.14 ImagePositionPatient . . . 872
27.350.4.15 LoadIconImage . . . 872
27.350.4.16 LoadOverlays . . . 872
27.350.4.17 LossyFlag . . . 872
27.350.4.18 NumberOfIconImages . . . 872
27.350.4.19 NumberOfOverlays . . . 873
27.350.4.20 PlanarConfiguration . . . 873
27.350.4.21 Scale . . . 873
27.350.4.22 Shift . . . 873
27.351 vtkGDCMImageWriter Class Reference . . . 873
27.351.1 Detailed Description . . . 875
27.351.2 Member Enumeration Documentation . . . 875
27.351.2.1 CompressionTypes . . . 875
27.351.3 Constructor & Destructor Documentation . . . 875
27.351.3.1 vtkGDCMImageWriter . . . 875
27.351.3.2 ~vtkGDCMImageWriter . . . 875
27.351.4 Member Function Documentation . . . 875
27.351.4.1 GetDescriptiveName . . . 875
27.351.4.2 GetFileExtensions . . . 875

27.351.4.3 GetFileName . . . 875
27.351.4.4 New . . . 875
27.351.4.5 PrintSelf . . . 876
27.351.4.6 SetDirectionCosines . . . 876
27.351.4.7 SetDirectionCosinesFromImageOrientationPatient . . . 876
27.351.4.8 SetFileNames . . . 876
27.351.4.9 SetMedicalImageProperties . . . 876
27.351.4.10 vtkBooleanMacro . . . 876
27.351.4.11 vtkBooleanMacro . . . 876
27.351.4.12 vtkGetMacro . . . 876
27.351.4.13 vtkGetMacro . . . 876
27.351.4.14 vtkGetMacro . . . 876
27.351.4.15 vtkGetMacro . . . 876
27.351.4.16 vtkGetMacro . . . 876
27.351.4.17 vtkGetMacro . . . 876
27.351.4.18 vtkGetMacro . . . 876
27.351.4.19 vtkGetObjectMacro . . . 876
27.351.4.20 vtkGetObjectMacro . . . 876
27.351.4.21 vtkGetObjectMacro . . . 876
27.351.4.22 vtkGetStringMacro . . . 876
27.351.4.23 vtkGetStringMacro . . . 877
27.351.4.24 vtkSetMacro . . . 877
27.351.4.25 vtkSetMacro . . . 877
27.351.4.26 vtkSetMacro . . . 877
27.351.4.27 vtkSetMacro . . . 877
27.351.4.28 vtkSetMacro . . . 877
27.351.4.29 vtkSetMacro . . . 877
27.351.4.30 vtkSetMacro . . . 877
27.351.4.31 vtkSetStringMacro . . . 877
27.351.4.32 vtkSetStringMacro . . . 877
27.351.4.33 vtkTypeRevisionMacro . . . 877
27.351.4.34 Write . . . 877
27.351.4.35 WriteGDCMData . . . 877
27.351.4.36 WriteSlice . . . 877
27.352 vtkGDCMMedicalImageProperties Class Reference . . . 877
27.352.1 Constructor & Destructor Documentation . . . 879
27.352.1.1 vtkGDCMMedicalImageProperties . . . 879

27.352.1.2 ~vtkGDCMMedicalImageProperties . . . 879
27.352.2 Member Function Documentation . . . 879
27.352.2.1 Clear . . . 879
27.352.2.2 GetFile . . . 879
27.352.2.3 New . . . 879
27.352.2.4 PrintSelf . . . 879
27.352.2.5 PushBackFile . . . 879
27.352.2.6 vtkTypeRevisionMacro . . . 879
27.352.3 Friends And Related Function Documentation . . . 879
27.352.3.1 vtkGDCMImageReader . . . 879
27.352.3.2 vtkGDCMImageReader2 . . . 879
27.352.3.3 vtkGDCMImageWriter . . . 879
27.353 vtkGDCMPolyDataReader Class Reference . . . 879
27.353.1 Detailed Description . . . 881
27.353.2 Constructor & Destructor Documentation . . . 881
27.353.2.1 vtkGDCMPolyDataReader . . . 881
27.353.2.2 ~vtkGDCMPolyDataReader . . . 881
27.353.3 Member Function Documentation . . . 881
27.353.3.1 FillMedicalImageInformation . . . 881
27.353.3.2 New . . . 881
27.353.3.3 PrintSelf . . . 881
27.353.3.4 RequestData . . . 881
27.353.3.5 RequestData_HemodynamicWaveformStorage . . . 881
27.353.3.6 RequestData_RTStructureSetStorage . . . 882
27.353.3.7 RequestInformation . . . 882
27.353.3.8 RequestInformation_HemodynamicWaveformStorage . . . 882
27.353.3.9 RequestInformation_RTStructureSetStorage . . . 882
27.353.3.10 vtkGetObjectMacro . . . 882
27.353.3.11 vtkGetObjectMacro . . . 882
27.353.3.12 vtkGetStringMacro . . . 882
27.353.3.13 vtkSetStringMacro . . . 882
27.353.3.14 vtkTypeRevisionMacro . . . 882
27.353.4 Member Data Documentation . . . 882
27.353.4.1 FileName . . . 882
27.353.4.2 MedicalImageProperties . . . 882
27.353.4.3 RTStructSetProperties . . . 882
27.354 vtkGDCMPolyDataWriter Class Reference . . . 882

27.354.1 Detailed Description . . . 884
27.354.2 Constructor & Destructor Documentation . . . 884
27.354.2.1 vtkGDCMPolyDataWriter . . . 884
27.354.2.2 ~vtkGDCMPolyDataWriter . . . 884
27.354.3 Member Function Documentation . . . 884
27.354.3.1 InitializeRTStructSet . . . 884
27.354.3.2 New . . . 884
27.354.3.3 PrintSelf . . . 884
27.354.3.4 SetMedicalImageProperties . . . 884
27.354.3.5 SetNumberOfInputPorts . . . 885
27.354.3.6 SetRTStructSetProperties . . . 885
27.354.3.7 vtkTypeRevisionMacro . . . 885
27.354.3.8 WriteData . . . 885
27.354.3.9 WriteRTSTRUCTData . . . 885
27.354.3.10 WriteRTSTRUCTInfo . . . 885
27.354.4 Member Data Documentation . . . 885
27.354.4.1 MedicalImageProperties . . . 885
27.354.4.2 RTStructSetProperties . . . 885
27.355 vtkGDCMTesting Class Reference . . . 885
27.355.1 Detailed Description . . . 886
27.355.2 Member Typedef Documentation . . . 887
27.355.2.1 MD5MetaImagesType . . . 887
27.355.3 Constructor & Destructor Documentation . . . 887
27.355.3.1 vtkGDCMTesting . . . 887
27.355.3.2 ~vtkGDCMTesting . . . 887
27.355.4 Member Function Documentation . . . 887
27.355.4.1 GetGDCMDataRoot . . . 887
27.355.4.2 GetMD5MetaImage . . . 887
27.355.4.3 GetMHDMD5FromFile . . . 887
27.355.4.4 GetNumberOfMD5MetaImages . . . 887
27.355.4.5 GetRAWMD5FromFile . . . 887
27.355.4.6 GetVTKDataRoot . . . 887
27.355.4.7 New . . . 887
27.355.4.8 PrintSelf . . . 888
27.355.4.9 vtkTypeRevisionMacro . . . 888
27.356 vtkGDCMThreadedImageReader Class Reference . . . 888
27.356.1 Constructor & Destructor Documentation . . . 889

27.356.1.1 vtkGDCMThreadedImageReader . . . 889
27.356.1.2 ~vtkGDCMThreadedImageReader . . . 890
27.356.2 Member Function Documentation . . . 890
27.356.2.1 ExecuteData . . . 890
27.356.2.2 ExecuteInformation . . . 890
27.356.2.3 New . . . 890
27.356.2.4 PrintSelf . . . 890
27.356.2.5 ReadFiles . . . 890
27.356.2.6 RequestDataCompat . . . 890
27.356.2.7 vtkBooleanMacro . . . 890
27.356.2.8 vtkGetMacro . . . 890
27.356.2.9 vtkSetMacro . . . 890
27.356.2.10 vtkSetMacro . . . 890
27.356.2.11 vtkSetMacro . . . 890
27.356.2.12 vtkTypeRevisionMacro . . . 890
27.357 vtkGDCMThreadedImageReader2 Class Reference . . . 890
27.357.1 Constructor & Destructor Documentation . . . 892
27.357.1.1 vtkGDCMThreadedImageReader2 . . . 892
27.357.1.2 ~vtkGDCMThreadedImageReader2 . . . 892
27.357.2 Member Function Documentation . . . 892
27.357.2.1 GetFileName . . . 892
27.357.2.2 New . . . 892
27.357.2.3 PrintSelf . . . 892
27.357.2.4 RequestInformation . . . 892
27.357.2.5 SetFileName . . . 892
27.357.2.6 SetFileNames . . . 893
27.357.2.7 SplitExtent . . . 893
27.357.2.8 ThreadedRequestData . . . 893
27.357.2.9 vtkBooleanMacro . . . 893
27.357.2.10 vtkBooleanMacro . . . 893
27.357.2.11 vtkBooleanMacro . . . 893
27.357.2.12 vtkGetMacro . . . 893
27.357.2.13 vtkGetMacro . . . 893
27.357.2.14 vtkGetMacro . . . 893
27.357.2.15 vtkGetMacro . . . 893
27.357.2.16 vtkGetMacro . . . 893
27.357.2.17 vtkGetMacro . . . 893

27.357.2.18 vtkGetMacro . . . 893
27.357.2.19 vtkGetMacro . . . 893
27.357.2.20 vtkGetObjectMacro . . . 893
27.357.2.21 vtkGetVector3Macro . . . 893
27.357.2.22 vtkGetVector3Macro . . . 893
27.357.2.23 vtkGetVector6Macro . . . 893
27.357.2.24 vtkSetMacro . . . 893
27.357.2.25 vtkSetMacro . . . 893
27.357.2.26 vtkSetMacro . . . 893
27.357.2.27 vtkSetMacro . . . 893
27.357.2.28 vtkSetMacro . . . 893
27.357.2.29 vtkSetMacro . . . 893
27.357.2.30 vtkSetMacro . . . 893
27.357.2.31 vtkSetVector3Macro . . . 894
27.357.2.32 vtkSetVector3Macro . . . 894
27.357.2.33 vtkSetVector6Macro . . . 894
27.357.2.34 vtkTypeRevisionMacro . . . 894
27.358 vtkImageColorViewer Class Reference . . . 894
27.358.1 Detailed Description . . . 897
27.358.2 Member Enumeration Documentation . . . 897
27.358.2.1 anonymous enum . . . 897
27.358.3 Constructor & Destructor Documentation . . . 897
27.358.3.1 vtkImageColorViewer . . . 897
27.358.3.2 ~vtkImageColorViewer . . . 897
27.358.4 Member Function Documentation . . . 897
27.358.4.1 AddInput . . . 897
27.358.4.2 AddInputConnection . . . 897
27.358.4.3 GetColorLevel . . . 897
27.358.4.4 GetColorWindow . . . 897
27.358.4.5 GetInput . . . 897
27.358.4.6 GetOffScreenRendering . . . 898
27.358.4.7 GetOverlayVisibility . . . 898
27.358.4.8 GetPosition . . . 898
27.358.4.9 GetSize . . . 898
27.358.4.10 GetSliceMax . . . 898
27.358.4.11 GetSliceMin . . . 898
27.358.4.12 GetSliceRange . . . 898

27.358.4.13 GetSliceRange . . . 898
27.358.4.14 GetSliceRange . . . 898
27.358.4.15 GetWindowName . . . 898
27.358.4.16 InstallPipeline . . . 898
27.358.4.17 New . . . 898
27.358.4.18 PrintSelf . . . 898
27.358.4.19 Render . . . 898
27.358.4.20 SetColorLevel . . . 898
27.358.4.21 SetColorWindow . . . 898
27.358.4.22 SetDisplayId . . . 898
27.358.4.23 SetInput . . . 898
27.358.4.24 SetInputConnection . . . 899
27.358.4.25 SetOffScreenRendering . . . 899
27.358.4.26 SetOverlayVisibility . . . 899
27.358.4.27 SetParentId . . . 899
27.358.4.28 SetPosition . . . 899
27.358.4.29 SetPosition . . . 899
27.358.4.30 SetRenderer . . . 899
27.358.4.31 SetRenderWindow . . . 899
27.358.4.32 SetSize . . . 899
27.358.4.33 SetSize . . . 899
27.358.4.34 SetSlice . . . 899
27.358.4.35 SetSliceOrientation . . . 899
27.358.4.36 SetSliceOrientationToXY . . . 899
27.358.4.37 SetSliceOrientationToXZ . . . 899
27.358.4.38 SetSliceOrientationToYZ . . . 899
27.358.4.39 SetupInteractor . . . 900
27.358.4.40 SetWindowId . . . 900
27.358.4.41 UnInstallPipeline . . . 900
27.358.4.42 UpdateDisplayExtent . . . 900
27.358.4.43 UpdateOrientation . . . 900
27.358.4.44 VTK_LEGACY . . . 900
27.358.4.45 VTK_LEGACY . . . 900
27.358.4.46 VTK_LEGACY . . . 900
27.358.4.47 VTK_LEGACY . . . 900
27.358.4.48 vtkBooleanMacro . . . 900
27.358.4.49 vtkGetMacro . . . 900

27.358.4.50 vtkGetMacro . . . 900
27.358.4.51 vtkGetObjectMacro . . . 900
27.358.4.52 vtkGetObjectMacro . . . 900
27.358.4.53 vtkGetObjectMacro . . . 900
27.358.4.54 vtkGetObjectMacro . . . 900
27.358.4.55 vtkGetObjectMacro . . . 900
27.358.4.56 vtkTypeRevisionMacro . . . 900
27.358.5 Friends And Related Function Documentation . . . 900
27.358.5.1 vtkImageColorViewerCallback . . . 900
27.358.6 Member Data Documentation . . . 900
27.358.6.1 FirstRender . . . 900
27.358.6.2 ImageActor . . . 900
27.358.6.3 Interactor . . . 900
27.358.6.4 InteractorStyle . . . 901
27.358.6.5 OverlayImageActor . . . 901
27.358.6.6 Renderer . . . 901
27.358.6.7 RenderWindow . . . 901
27.358.6.8 Slice . . . 901
27.358.6.9 SliceOrientation . . . 901
27.358.6.10 WindowLevel . . . 901
27.359 vtkImageMapToColors16 Class Reference . . . 901
27.359.1 Constructor & Destructor Documentation . . . 903
27.359.1.1 vtkImageMapToColors16 . . . 903
27.359.1.2 ~vtkImageMapToColors16 . . . 903
27.359.2 Member Function Documentation . . . 903
27.359.2.1 GetMTime . . . 903
27.359.2.2 New . . . 903
27.359.2.3 PrintSelf . . . 903
27.359.2.4 RequestData . . . 903
27.359.2.5 RequestInformation . . . 903
27.359.2.6 SetLookupTable . . . 903
27.359.2.7 SetOutputFormatToLuminance . . . 903
27.359.2.8 SetOutputFormatToLuminanceAlpha . . . 903
27.359.2.9 SetOutputFormatToRGB . . . 903
27.359.2.10 SetOutputFormatToRGBA . . . 903
27.359.2.11 ThreadedRequestData . . . 903
27.359.2.12 vtkBooleanMacro . . . 903

27.359.2.13 vtkGetMacro . . . 903
27.359.2.14 vtkGetMacro . . . 903
27.359.2.15 vtkGetMacro . . . 903
27.359.2.16 vtkGetObjectMacro . . . 903
27.359.2.17 vtkSetMacro . . . 904
27.359.2.18 vtkSetMacro . . . 904
27.359.2.19 vtkSetMacro . . . 904
27.359.2.20 vtkTypeRevisionMacro . . . 904
27.359.3 Member Data Documentation . . . 904
27.359.3.1 ActiveComponent . . . 904
27.359.3.2 DataWasPassed . . . 904
27.359.3.3 LookupTable . . . 904
27.359.3.4 OutputFormat . . . 904
27.359.3.5 PassAlphaToOutput . . . 904
27.360 vtkImageMapToWindowLevelColors2 Class Reference . . . 904
27.360.1 Constructor & Destructor Documentation . . . 906
27.360.1.1 vtkImageMapToWindowLevelColors2 . . . 906
27.360.1.2 ~vtkImageMapToWindowLevelColors2 . . . 906
27.360.2 Member Function Documentation . . . 906
27.360.2.1 New . . . 906
27.360.2.2 PrintSelf . . . 906
27.360.2.3 RequestData . . . 906
27.360.2.4 RequestInformation . . . 906
27.360.2.5 ThreadedRequestData . . . 906
27.360.2.6 vtkGetMacro . . . 906
27.360.2.7 vtkGetMacro . . . 906
27.360.2.8 vtkSetMacro . . . 906
27.360.2.9 vtkSetMacro . . . 906
27.360.2.10 vtkTypeRevisionMacro . . . 906
27.360.3 Member Data Documentation . . . 906
27.360.3.1 Level . . . 906
27.360.3.2 Window . . . 906
27.361 vtkImagePlanarComponentsToComponents Class Reference . . . 906
27.361.1 Constructor & Destructor Documentation . . . 908
27.361.1.1 vtkImagePlanarComponentsToComponents . . . 908
27.361.1.2 ~vtkImagePlanarComponentsToComponents . . . 908
27.361.2 Member Function Documentation . . . 908

27.361.2.1 New . . . 908
27.361.2.2 PrintSelf . . . 908
27.361.2.3 RequestData . . . 908
27.361.2.4 vtkTypeRevisionMacro . . . 908
27.362 vtkImageRGBToYBR Class Reference . . . 908
27.362.1 Constructor & Destructor Documentation . . . 909
27.362.1.1 vtkImageRGBToYBR . . . 909
27.362.1.2 ~vtkImageRGBToYBR . . . 909
27.362.2 Member Function Documentation . . . 909
27.362.2.1 New . . . 909
27.362.2.2 PrintSelf . . . 909
27.362.2.3 ThreadedExecute . . . 909
27.362.2.4 vtkTypeRevisionMacro . . . 909
27.363 vtkImageYBRToRGB Class Reference . . . 910
27.363.1 Constructor & Destructor Documentation . . . 911
27.363.1.1 vtkImageYBRToRGB . . . 911
27.363.1.2 ~vtkImageYBRToRGB . . . 911
27.363.2 Member Function Documentation . . . 911
27.363.2.1 New . . . 911
27.363.2.2 PrintSelf . . . 911
27.363.2.3 ThreadedExecute . . . 911
27.363.2.4 vtkTypeRevisionMacro . . . 911
27.364 vtkLookupTable16 Class Reference . . . 911
27.364.1 Constructor & Destructor Documentation . . . 912
27.364.1.1 vtkLookupTable16 . . . 912
27.364.1.2 ~vtkLookupTable16 . . . 912
27.364.2 Member Function Documentation . . . 912
27.364.2.1 Build . . . 913
27.364.2.2 GetPointer . . . 913
27.364.2.3 MapScalarsThroughTable2 . . . 913
27.364.2.4 New . . . 913
27.364.2.5 PrintSelf . . . 913
27.364.2.6 SetNumberOfTableValues . . . 913
27.364.2.7 vtkTypeRevisionMacro . . . 913
27.364.2.8 WritePointer . . . 913
27.364.3 Member Data Documentation . . . 913
27.364.3.1 Table16 . . . 913

27.365 vtkRTStructSetProperties Class Reference . . . 913
27.365.1 Detailed Description . . . 915
27.365.2 Constructor & Destructor Documentation . . . 915
27.365.2.1 vtkRTStructSetProperties . . . 915
27.365.2.2 ~vtkRTStructSetProperties . . . 915
27.365.3 Member Function Documentation . . . 915
27.365.3.1 AddContourReferencedFrameOfReference . . . 916
27.365.3.2 AddReferencedFrameOfReference . . . 916
27.365.3.3 AddStructureSetROI . . . 916
27.365.3.4 AddStructureSetROIObservation . . . 916
27.365.3.5 Clear . . . 916
27.365.3.6 DeepCopy . . . 916
27.365.3.7 GetContourReferencedFrameOfReferenceClassUID . . . 916
27.365.3.8 GetContourReferencedFrameOfReferenceInstanceUID . . . 916
27.365.3.9 GetNumberOfContourReferencedFrameOfReferences . . . 916
27.365.3.10 GetNumberOfContourReferencedFrameOfReferences . . . 916
27.365.3.11 GetNumberOfReferencedFrameOfReferences . . . 916
27.365.3.12 GetNumberOfStructureSetROIs . . . 916
27.365.3.13 GetReferencedFrameOfReferenceClassUID . . . 916
27.365.3.14 GetReferencedFrameOfReferenceInstanceUID . . . 916
27.365.3.15 GetStructureSetObservationNumber . . . 916
27.365.3.16 GetStructureSetROIDescription . . . 916
27.365.3.17 GetStructureSetROIGenerationAlgorithm . . . 916
27.365.3.18 GetStructureSetROIName . . . 916
27.365.3.19 GetStructureSetROINumber . . . 916
27.365.3.20 GetStructureSetROIObservationLabel . . . 916
27.365.3.21 GetStructureSetROIRefFrameRefUID . . . 916
27.365.3.22 GetStructureSetRTROIInterpretedType . . . 916
27.365.3.23 New . . . 917
27.365.3.24 PrintSelf . . . 917
27.365.3.25 vtkGetStringMacro . . . 917
27.365.3.26 vtkGetStringMacro . . . 917
27.365.3.27 vtkGetStringMacro . . . 917
27.365.3.28 vtkGetStringMacro . . . 917
27.365.3.29 vtkGetStringMacro . . . 917
27.365.3.30 vtkGetStringMacro . . . 917
27.365.3.31 vtkGetStringMacro . . . 917

27.365.3.32 vtkGetStringMacro . . . 917
27.365.3.33 vtkGetStringMacro . . . 917
27.365.3.34 vtkSetStringMacro . . . 917
27.365.3.35 vtkSetStringMacro . . . 917
27.365.3.36 vtkSetStringMacro . . . 917
27.365.3.37 vtkSetStringMacro . . . 917
27.365.3.38 vtkSetStringMacro . . . 917
27.365.3.39 vtkSetStringMacro . . . 917
27.365.3.40 vtkSetStringMacro . . . 917
27.365.3.41 vtkSetStringMacro . . . 917
27.365.3.42 vtkSetStringMacro . . . 917
27.365.3.43 vtkTypeRevisionMacro . . . 917
27.365.4 Member Data Documentation . . . 917
27.365.4.1 Internals . . . 917
27.365.4.2 ReferenceFrameOfReferenceUID . . . 917
27.365.4.3 ReferenceSeriesInstanceUID . . . 918
27.365.4.4 SeriesInstanceUID . . . 918
27.365.4.5 SOPInstanceUID . . . 918
27.365.4.6 StructureSetDate . . . 918
27.365.4.7 StructureSetLabel . . . 918
27.365.4.8 StructureSetName . . . 918
27.365.4.9 StructureSetTime . . . 918
27.365.4.10 StudyInstanceUID . . . 918
27.366 gdcm::Waveform Class Reference . . . 918
27.366.1 Detailed Description . . . 918
27.366.2 Constructor & Destructor Documentation . . . 918
27.366.2.1 Waveform . . . 918
27.367 gdcm::Writer Class Reference . . . 918
27.367.1 Detailed Description . . . 921
27.367.2 Constructor & Destructor Documentation . . . 922
27.367.2.1 Writer . . . 922
27.367.2.2 ~Writer . . . 922
27.367.3 Member Function Documentation . . . 922
27.367.3.1 CheckFileMetaInformationOff . . . 922
27.367.3.2 CheckFileMetaInformationOn . . . 922
27.367.3.3 GetFile . . . 922
27.367.3.4 GetStreamPtr . . . 922

27.367.3.5 SetCheckFileMetaInformation . . . 922
27.367.3.6 SetFile . . . 922
27.367.3.7 SetFileName . . . 922
27.367.3.8 SetStream . . . 923
27.367.3.9 SetWriteDataSetOnly . . . 923
27.367.3.10 Write . . . 923
27.367.4 Friends And Related Function Documentation . . . 923
27.367.4.1 StreamImageWriter . . . 923
27.367.5 Member Data Documentation . . . 923
27.367.5.1 Ofstream . . . 923
27.367.5.2 Stream . . . 923
27.368 gdcm::XMLDictReader Class Reference . . . 923
27.368.1 Detailed Description . . . 924
27.368.2 Constructor & Destructor Documentation . . . 925
27.368.2.1 XMLDictReader . . . 925
27.368.2.2 ~XMLDictReader . . . 925
27.368.3 Member Function Documentation . . . 925
27.368.3.1 CharacterDataHandler . . . 925
27.368.3.2 EndElement . . . 925
27.368.3.3 GetDict . . . 925
27.368.3.4 HandleDescription . . . 925
27.368.3.5 HandleEntry . . . 925
27.368.3.6 StartElement . . . 925
27.369 gdcm::XMLPrinter Class Reference . . . 925
27.369.1 Member Enumeration Documentation . . . 927
27.369.1.1 PrintStyles . . . 927
27.369.2 Constructor & Destructor Documentation . . . 927
27.369.2.1 XMLPrinter . . . 927
27.369.2.2 ~XMLPrinter . . . 927
27.369.3 Member Function Documentation . . . 927
27.369.3.1 GetPrintStyle . . . 927
27.369.3.2 HandleBulkData . . . 927
27.369.3.3 Print . . . 927
27.369.3.4 PrintDataElement . . . 927
27.369.3.5 PrintDataSet . . . 927
27.369.3.6 PrintSQ . . . 927
27.369.3.7 SetFile . . . 927

27.369.3.8 SetStyle . . . 927
27.369.4 Member Data Documentation . . . 927
27.369.4.1 F . . . 927
27.369.4.2 PrintStyle . . . 927
27.370 gdcm::XMLPrivateDictReader Class Reference . . . 928
27.370.1 Detailed Description . . . 929
27.370.2 Constructor & Destructor Documentation . . . 929
27.370.2.1 XMLPrivateDictReader . . . 929
27.370.2.2 ~XMLPrivateDictReader . . . 929
27.370.3 Member Function Documentation . . . 929
27.370.3.1 CharacterDataHandler . . . 929
27.370.3.2 EndElement . . . 929
27.370.3.3 GetPrivateDict . . . 929
27.370.3.4 HandleDescription . . . 929
27.370.3.5 HandleEntry . . . 929
27.370.3.6 StartElement . . . 929

28 File Documentation **931**

28.1 gdcm2pnm.dox File Reference . . . 931
28.2 gdcm2vtk.dox File Reference . . . 931
28.3 gdcmAAbortPDU.h File Reference . . . 931
28.4 gdcmAAssociateACPDU.h File Reference . . . 932
28.5 gdcmAAssociateRJPDU.h File Reference . . . 932
28.6 gdcmAAssociateRQPDU.h File Reference . . . 933
28.7 gdcmAbstractSyntax.h File Reference . . . 934
28.8 gdcmanon.dox File Reference . . . 935
28.9 gdcmAnonymizeEvent.h File Reference . . . 935
28.10 gdcmAnonymizer.h File Reference . . . 936
28.11 gdcmApplicationContext.h File Reference . . . 937
28.12 gdcmApplicationEntity.h File Reference . . . 938
28.13 gdcmAReleaseRPPDU.h File Reference . . . 939
28.14 gdcmAReleaseRQPDU.h File Reference . . . 940
28.15 gdcmARTIMTimer.h File Reference . . . 941
28.16 gdcmASN1.h File Reference . . . 942
28.17 gdcmAsynchronousOperationsWindowSub.h File Reference . . . 943
28.18 gdcmAttribute.h File Reference . . . 943
28.19 gdcmAudioCodec.h File Reference . . . 945

28.20gdcmBase64.h File Reference . . . 945
28.21gdcmBaseCompositeMessage.h File Reference . . . 946
28.22gdcmBasePDU.h File Reference . . . 947
28.23gdcmBaseRootQuery.h File Reference . . . 948
28.24gdcmBasicOffsetTable.h File Reference . . . 949
28.25gdcmBitmap.h File Reference . . . 951
28.26gdcmBitmapToBitmapFilter.h File Reference . . . 952
28.27gdcmBoxRegion.h File Reference . . . 953
28.28gdcmByteBuffer.h File Reference . . . 953
28.29gdcmByteSwap.h File Reference . . . 954
28.30gdcmByteSwapFilter.h File Reference . . . 955
28.31gdcmByteValue.h File Reference . . . 956
28.32gdcmCAPICryptoFactory.h File Reference . . . 957
28.33gdcmCAPICryptographicMessageSyntax.h File Reference . . . 957
28.34gdcmCEchoMessages.h File Reference . . . 958
28.35gdcmCFindMessages.h File Reference . . . 959
28.36gdcmCMoveMessages.h File Reference . . . 960
28.37gdcmCodec.h File Reference . . . 961
28.38gdcmCoder.h File Reference . . . 962
28.39gdcmCodeString.h File Reference . . . 963
28.40gdcmCommand.h File Reference . . . 964
28.41gdcmCommandDataSet.h File Reference . . . 966
28.42gdcmCompositeMessageFactory.h File Reference . . . 966
28.43gdcmCompositeNetworkFunctions.h File Reference . . . 967
28.44gdcmConstCharWrapper.h File Reference . . . 968
28.45gdcmconv.dox File Reference . . . 968
28.46gdcmCP246ExplicitDataElement.h File Reference . . . 969
28.47gdcmCryptoFactory.h File Reference . . . 969
28.48gdcmCryptographicMessageSyntax.h File Reference . . . 970
28.49gdcmCSAElement.h File Reference . . . 971
28.50gdcmCSAHeader.h File Reference . . . 972
28.51gdcmCSAHeaderDict.h File Reference . . . 973
28.52gdcmCSAHeaderDictEntry.h File Reference . . . 975
28.53gdcmCStoreMessages.h File Reference . . . 976
28.54gdcmCurve.h File Reference . . . 977
28.55gdcmDataElement.h File Reference . . . 978
28.56gdcmDataEvent.h File Reference . . . 980

28.57gdcmDataSet.h File Reference . . . 981
28.58gdcmDataSetEvent.h File Reference . . . 982
28.59gdcmDataSetHelper.h File Reference . . . 982
28.60gdcmDecoder.h File Reference . . . 983
28.61gdcmDefinedTerms.h File Reference . . . 984
28.62gdcmDeflateStream.h File Reference . . . 985
28.63gdcmDefs.h File Reference . . . 985
28.64gdcmDeltaEncodingCodec.h File Reference . . . 987
28.65gdcmDICOMDIR.h File Reference . . . 987
28.66gdcmDICOMDIRGenerator.h File Reference . . . 988
28.67gdcmDict.h File Reference . . . 989
28.68gdcmDictConverter.h File Reference . . . 991
28.69gdcmDictEntry.h File Reference . . . 991
28.70gdcmDictPrinter.h File Reference . . . 993
28.71gdcmDicts.h File Reference . . . 993
28.72gdcmdiff.dox File Reference . . . 994
28.73gdcmDIMSE.h File Reference . . . 995
28.74gdcmDirectionCosines.h File Reference . . . 995
28.75gdcmDirectory.h File Reference . . . 996
28.76gdcmDirectoryHelper.h File Reference . . . 997
28.77gdcmDummyValueGenerator.h File Reference . . . 998
28.78gdcmdump.dox File Reference . . . 998
28.79gdcmDumper.h File Reference . . . 999
28.80gdcmElement.h File Reference . . . 999
28.81gdcmEncapsulatedDocument.h File Reference . . . 1001
28.82gdcmEnumeratedValues.h File Reference . . . 1001
28.83gdcmEvent.h File Reference . . . 1002
28.83.1 Macro Definition Documentation . . . 1003
28.83.1.1 gdcmEventMacro . . . 1003
28.84gdcmException.h File Reference . . . 1004
28.85gdcmExplicitDataElement.h File Reference . . . 1004
28.86gdcmExplicitImplicitDataElement.h File Reference . . . 1005
28.87gdcmFiducials.h File Reference . . . 1006
28.88gdcmFile.h File Reference . . . 1007
28.89gdcmFileAnonymizer.h File Reference . . . 1008
28.90gdcmFileChangeTransferSyntax.h File Reference . . . 1008
28.91gdcmFileDerivation.h File Reference . . . 1009

28.92gdcmFileExplicitFilter.h File Reference . . . 1010
28.93gdcmFileMetaInformation.h File Reference . . . 1011
28.94gdcmFilename.h File Reference . . . 1012
28.95gdcmFileNameEvent.h File Reference . . . 1013
28.96gdcmFilenameGenerator.h File Reference . . . 1014
28.97gdcmFileSet.h File Reference . . . 1015
28.98gdcmFileStreamer.h File Reference . . . 1016
28.99gdcmFindPatientRootQuery.h File Reference . . . 1017
28.100gdcmFindStudyRootQuery.h File Reference . . . 1018
28.101gdcmFragment.h File Reference . . . 1019
28.102gdcmgendir.dox File Reference . . . 1021
28.103gdcmGlobal.h File Reference . . . 1021
28.104gdcmGroupDict.h File Reference . . . 1022
28.105gdcmIconImage.h File Reference . . . 1022
28.106gdcmIconImageFilter.h File Reference . . . 1023
28.107gdcmIconImageGenerator.h File Reference . . . 1024
28.108gdcmImage.h File Reference . . . 1025
28.109gdcmImageApplyLookupTable.h File Reference . . . 1026
28.110gdcmImageChangePhotometricInterpretation.h File Reference . . . 1027
28.111gdcmImageChangePlanarConfiguration.h File Reference . . . 1028
28.112gdcmImageChangeTransferSyntax.h File Reference . . . 1029
28.113gdcmImageCodec.h File Reference . . . 1030
28.114gdcmImageConverter.h File Reference . . . 1031
28.115gdcmImageFragmentSplitter.h File Reference . . . 1032
28.116gdcmImageHelper.h File Reference . . . 1033
28.117gdcmImageReader.h File Reference . . . 1034
28.118gdcmImageRegionReader.h File Reference . . . 1034
28.119gdcmImageToImageFilter.h File Reference . . . 1035
28.120gdcmImageWriter.h File Reference . . . 1036
28.121gdcmimg.dox File Reference . . . 1037
28.122gdcmImplementationClassUIDSub.h File Reference . . . 1037
28.123gdcmImplementationUIDSub.h File Reference . . . 1038
28.124gdcmImplementationVersionNameSub.h File Reference . . . 1039
28.125gdcmImplicitDataElement.h File Reference . . . 1040
28.126gdcminfo.dox File Reference . . . 1041
28.127gdcmIOD.h File Reference . . . 1041
28.128gdcmIODEntry.h File Reference . . . 1043

28.129 gdcmIODs.h File Reference . . . 1045
28.130 gdcmIPPSorter.h File Reference . . . 1046
28.131 gdcmItem.h File Reference . . . 1047
28.132 gdcmJPEG12Codec.h File Reference . . . 1048
28.133 gdcmJPEG16Codec.h File Reference . . . 1049
28.134 gdcmJPEG2000Codec.h File Reference . . . 1049
28.135 gdcmJPEG8Codec.h File Reference . . . 1050
28.136 gdcmJPEGCodec.h File Reference . . . 1051
28.137 gdcmJPEGLSCodec.h File Reference . . . 1052
28.138 gdcmJSON.h File Reference . . . 1053
28.139 gdcmKAKADUCodec.h File Reference . . . 1054
28.140 gdcmLegacyMacro.h File Reference . . . 1055
28.140.1 Macro Definition Documentation . . . 1055
28.140.1.1 GDCM_LEGACY . . . 1055
28.140.1.2 GDCM_LEGACY_BODY . . . 1056
28.140.1.3 GDCM_LEGACY_REPLACED_BODY . . . 1056
28.141 gdcmLO.h File Reference . . . 1056
28.142 gdcmLookupTable.h File Reference . . . 1056
28.143 gdcmMacro.h File Reference . . . 1057
28.144 gdcmMacroEntry.h File Reference . . . 1059
28.144.1 Macro Definition Documentation . . . 1061
28.144.1.1 GDCMMACROENTRY_H . . . 1061
28.145 gdcmMacros.h File Reference . . . 1061
28.146 gdcmMaximumLengthSub.h File Reference . . . 1063
28.147 gdcmMD5.h File Reference . . . 1064
28.148 gdcmMediaStorage.h File Reference . . . 1065
28.149 gdcmMeshPrimitive.h File Reference . . . 1066
28.150 gdcmModule.h File Reference . . . 1067
28.151 gdcmModuleEntry.h File Reference . . . 1069
28.152 gdcmModules.h File Reference . . . 1071
28.153 gdcmMovePatientRootQuery.h File Reference . . . 1072
28.154 gdcmMoveStudyRootQuery.h File Reference . . . 1073
28.155 gdcmNestedModuleEntries.h File Reference . . . 1074
28.156 gdcmNetworkEvents.h File Reference . . . 1076
28.157 gdcmNetworkStateID.h File Reference . . . 1077
28.158 gdcmObject.h File Reference . . . 1078
28.159 gdcmOpenSSLCryptoFactory.h File Reference . . . 1079

28.160gdcmOpenSSLCryptographicMessageSyntax.h File Reference 1079
28.161gdcmOpenSSLP7CryptoFactory.h File Reference 1080
28.162gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference 1081
28.163gdcmOrientation.h File Reference 1083
28.164gdcmOverlay.h File Reference 1083
28.165gdcmpap3.dox File Reference 1084
28.166gdcmParseException.h File Reference 1084
28.167gdcmParser.h File Reference 1086
28.168gdcmPatient.h File Reference 1086
28.169gdcmPDataTFPDU.h File Reference 1087
28.170gdcmPDBElement.h File Reference 1088
28.171gdcmPDBHeader.h File Reference 1090
28.172gdcmpdf.dox File Reference 1090
28.173gdcmPDFCodec.h File Reference 1090
28.174gdcmPDUFactory.h File Reference 1091
28.175gdcmPersonName.h File Reference 1092
28.176gdcmPGXCodec.h File Reference 1093
28.177gdcmPhotometricInterpretation.h File Reference 1093
28.178gdcmPixelFormat.h File Reference 1094
28.179gdcmPixmap.h File Reference 1095
28.180gdcmPixmapReader.h File Reference 1096
28.181gdcmPixmapToPixmapFilter.h File Reference 1097
28.182gdcmPixmapWriter.h File Reference 1098
28.183gdcmPNMCodec.h File Reference 1099
28.184gdcmPreamble.h File Reference 1100
28.185gdcmPresentationContext.h File Reference 1101
28.186gdcmPresentationContextAC.h File Reference 1102
28.187gdcmPresentationContextGenerator.h File Reference 1104
28.188gdcmPresentationContextRQ.h File Reference 1104
28.189gdcmPresentationDataValue.h File Reference 1105
28.190gdcmPrinter.h File Reference 1106
28.191gdcmPrivateTag.h File Reference 1107
28.192gdcmProgressEvent.h File Reference 1109
28.193gdcmPVRGCodec.h File Reference 1109
28.194gdcmPythonFilter.h File Reference 1110
28.195gdcmQueryBase.h File Reference 1111
28.196gdcmQueryFactory.h File Reference 1112

28.197gdcmQueryImage.h File Reference . . . 1113
28.198gdcmQueryPatient.h File Reference . . . 1114
28.199gdcmQuerySeries.h File Reference . . . 1115
28.200gdcmQueryStudy.h File Reference . . . 1116
28.201gdcmraw.dox File Reference . . . 1117
28.202gdcmRAWCodec.h File Reference . . . 1117
28.203gdcmReader.h File Reference . . . 1118
28.204gdcmRegion.h File Reference . . . 1119
28.205gdcmRescaler.h File Reference . . . 1120
28.206gdcmRLECodec.h File Reference . . . 1121
28.207gdcmRoleSelectionSub.h File Reference . . . 1121
28.208gdcmscanner.dox File Reference . . . 1122
28.209gdcmScanner.h File Reference . . . 1122
28.210gdcmscu.dox File Reference . . . 1123
28.211gdcmSegment.h File Reference . . . 1123
28.212gdcmSegmentedPaletteColorLookupTable.h File Reference . . . 1125
28.213gdcmSegmentHelper.h File Reference . . . 1125
28.214gdcmSegmentReader.h File Reference . . . 1127
28.215gdcmSegmentWriter.h File Reference . . . 1128
28.216gdcmSequenceOfFragments.h File Reference . . . 1129
28.217gdcmSequenceOfItems.h File Reference . . . 1129
28.218gdcmSerieHelper.h File Reference . . . 1130
28.219gdcmSeries.h File Reference . . . 1132
28.220gdcmServiceClassApplicationInformation.h File Reference . . . 1133
28.221gdcmServiceClassUser.h File Reference . . . 1134
28.222gdcmSHA1.h File Reference . . . 1134
28.223gdcmSimpleSubjectWatcher.h File Reference . . . 1135
28.224gdcmSmartPointer.h File Reference . . . 1136
28.225gdcmSOPClassExtendedNegociationSub.h File Reference . . . 1137
28.226gdcmSOPClassUIDToIOD.h File Reference . . . 1138
28.227gdcmSorter.h File Reference . . . 1139
28.228gdcmSpacing.h File Reference . . . 1141
28.229gdcmSpectroscopy.h File Reference . . . 1141
28.230gdcmSplitMosaicFilter.h File Reference . . . 1142
28.231gdcmStaticAssert.h File Reference . . . 1143
28.231.1Macro Definition Documentation . . . 1143
28.231.1.1GDCM_DO_JOIN . . . 1143

28.231.1.2 GDCM_DO_JOIN2 . . . 1143
28.231.1.3 GDCM_JOIN . . . 1143
28.231.1.4 GDCM_STATIC_ASSERT . . . 1144
28.232 gdcmStreamImageReader.h File Reference . . . 1144
28.233 gdcmStreamImageWriter.h File Reference . . . 1144
28.234 gdcmString.h File Reference . . . 1145
28.235 gdcmStringFilter.h File Reference . . . 1146
28.236 gdcmStudy.h File Reference . . . 1147
28.237 gdcmSubject.h File Reference . . . 1148
28.238 gdcmSurface.h File Reference . . . 1149
28.239 gdcmSurfaceHelper.h File Reference . . . 1150
28.240 gdcmSurfaceReader.h File Reference . . . 1151
28.241 gdcmSurfaceWriter.h File Reference . . . 1152
28.242 gdcmSwapCode.h File Reference . . . 1153
28.243 gdcmSwapper.h File Reference . . . 1154
28.244 gdcmSystem.h File Reference . . . 1155
28.245 gdcmTable.h File Reference . . . 1156
28.246 gdcmTableEntry.h File Reference . . . 1157
28.247 gdcmTableReader.h File Reference . . . 1158
28.248 gdcmTag.h File Reference . . . 1160
28.249 gdcmTagPath.h File Reference . . . 1161
28.250 gdcmTagToVR.h File Reference . . . 1161
28.251 gdcmtar.dox File Reference . . . 1162
28.252 gdcmTerminal.h File Reference . . . 1162
28.253 gdcmTestDriver.h File Reference . . . 1164
28.254 gdcmTesting.h File Reference . . . 1164
28.255 gdcmTrace.h File Reference . . . 1165
28.255.1 Macro Definition Documentation . . . 1166
28.255.1.1 GDCM_FUNCTION . . . 1166
28.255.1.2 gdcmAssertAlwaysMacro . . . 1166
28.255.1.3 gdcmAssertMacro . . . 1166
28.255.1.4 gdcmDebugMacro . . . 1167
28.255.1.5 gdcmErrorMacro . . . 1167
28.255.1.6 gdcmWarningMacro . . . 1167
28.256 gdcmTransferSyntax.h File Reference . . . 1168
28.257 gdcmTransferSyntaxSub.h File Reference . . . 1169
28.258 gdcmType.h File Reference . . . 1170

28.259 gdcmTypes.h File Reference . . . 1171
28.260 gdcmUIDGenerator.h File Reference . . . 1172
28.261 gdcmUIDs.h File Reference . . . 1173
28.262 gdcmULAction.h File Reference . . . 1173
28.263 gdcmULActionAA.h File Reference . . . 1174
28.264 gdcmULActionAE.h File Reference . . . 1175
28.265 gdcmULActionAR.h File Reference . . . 1176
28.266 gdcmULActionDT.h File Reference . . . 1177
28.267 gdcmULBasicCallback.h File Reference . . . 1177
28.268 gdcmULConnection.h File Reference . . . 1178
28.269 gdcmULConnectionCallback.h File Reference . . . 1179
28.270 gdcmULConnectionInfo.h File Reference . . . 1180
28.271 gdcmULConnectionManager.h File Reference . . . 1182
28.272 gdcmULEvent.h File Reference . . . 1182
28.273 gdcmULTransitionTable.h File Reference . . . 1183
28.274 gdcmULWritingCallback.h File Reference . . . 1185
28.275 gdcmUNExplicitDataElement.h File Reference . . . 1185
28.276 gdcmUNExplicitImplicitDataElement.h File Reference . . . 1186
28.277 gdcmUnpacker12Bits.h File Reference . . . 1187
28.278 gdcmUsage.h File Reference . . . 1187
28.279 gdcmUserInformation.h File Reference . . . 1190
28.280 gdcmUUIDGenerator.h File Reference . . . 1191
28.281 gdcmValidate.h File Reference . . . 1191
28.282 gdcmValue.h File Reference . . . 1192
28.283 gdcmValueIO.h File Reference . . . 1193
28.284 gdcmVersion.h File Reference . . . 1194
28.285 gdcmviewer.dox File Reference . . . 1195
28.286 gdcmVL.h File Reference . . . 1195
28.287 gdcmVM.h File Reference . . . 1196
28.287.1 Macro Definition Documentation . . . 1197
28.287.1.1 TYPETOLENGTH . . . 1197
28.288 gdcmVR.h File Reference . . . 1197
28.288.1 Macro Definition Documentation . . . 1199
28.288.1.1 TYPETOENCODING . . . 1199
28.288.1.2 VRTypeTemplateCase . . . 1199
28.289 gdcmVR16ExplicitDataElement.h File Reference . . . 1200
28.290 gdcmWaveform.h File Reference . . . 1200

28.291gdcmWin32.h File Reference . . . 1201
28.291.1Macro Definition Documentation . . . 1201
28.291.1.1GDCM_EXPORT . . . 1201
28.292gdcmWriter.h File Reference . . . 1202
28.293gdcmxml.dox File Reference . . . 1203
28.294gdcmXMLDictReader.h File Reference . . . 1203
28.295gdcmXMLPrinter.h File Reference . . . 1203
28.296gdcmXMLPrivateDictReader.h File Reference . . . 1204
28.297README.txt File Reference . . . 1205
28.298TestsList.txt File Reference . . . 1205
28.299vtkGDCMImageReader.h File Reference . . . 1205
28.299.1Macro Definition Documentation . . . 1206
28.299.1.1VTK_CMYK . . . 1206
28.299.1.2VTK_INVERSE_LUMINANCE . . . 1206
28.299.1.3VTK_LOOKUP_TABLE . . . 1206
28.299.1.4VTK_YBR . . . 1206
28.300vtkGDCMImageReader2.h File Reference . . . 1206
28.300.1Macro Definition Documentation . . . 1207
28.300.1.1VTK_CMYK . . . 1207
28.300.1.2VTK_INVERSE_LUMINANCE . . . 1207
28.300.1.3VTK_LOOKUP_TABLE . . . 1207
28.300.1.4VTK_YBR . . . 1207
28.301vtkGDCMImageWriter.h File Reference . . . 1207
28.302vtkGDCMMedicalImageProperties.h File Reference . . . 1207
28.303vtkGDCMPolyDataReader.h File Reference . . . 1208
28.304vtkGDCMPolyDataWriter.h File Reference . . . 1209
28.305vtkGDCMTesting.h File Reference . . . 1209
28.306vtkGDCMThreadedImageReader.h File Reference . . . 1210
28.307vtkGDCMThreadedImageReader2.h File Reference . . . 1211
28.308vtkImageColorViewer.h File Reference . . . 1211
28.309vtkImageMapToColors16.h File Reference . . . 1212
28.310vtkImageMapToWindowLevelColors2.h File Reference . . . 1212
28.311vtkImagePlanarComponentsToComponents.h File Reference . . . 1213
28.312vtkImageRGBToYBR.h File Reference . . . 1213
28.313vtkImageYBRToRGB.h File Reference . . . 1214
28.314vtkLookupTable16.h File Reference . . . 1214
28.315vtkRTStructSetProperties.h File Reference . . . 1215

29 Example Documentation **1217**

29.1 AWTMedical3.java . . . 1217
29.2 BasicAnonymizer.cs . . . 1221
29.3 BasicImageAnonymizer.cs . . . 1222
29.4 CastConvertPhilips.py . . . 1224
29.5 ChangePrivateTags.cxx . . . 1226
29.6 ChangeSequenceUltrasound.cxx . . . 1227
29.7 CheckBigEndianBug.cxx . . . 1228
29.8 ClinicalTrialAnnotate.cxx . . . 1230
29.9 ClinicalTrialIdentificationWorkflow.cs . . . 1231
29.10 CompressImage.cxx . . . 1234
29.11 CompressLossyJPEG.cs . . . 1235
29.12 Compute3DSpacing.cxx . . . 1236
29.13 Convert16BitsTo8Bits.cxx . . . 1237
29.14 ConvertMPL.py . . . 1238
29.15 ConvertMultiFrameToSingleFrame.cxx . . . 1239
29.16 ConvertNumpy.py . . . 1240
29.17 ConvertPIL.py . . . 1241
29.18 ConvertRGBToLuminance.cxx . . . 1243
29.19 ConvertSingleBitTo8Bits.cxx . . . 1243
29.20 ConvertToQImage.cxx . . . 1245
29.21 CreateARGBImage.cxx . . . 1246
29.22 CreateCMYKImage.cxx . . . 1247
29.23 CreateFakeRTDOSE.cxx . . . 1248
29.24 CreateJPIPDataSet.cxx . . . 1250
29.25 CreateRAWStorage.py . . . 1251
29.26 csa2img.cxx . . . 1253
29.27 CStoreQtProgress.cxx . . . 1255
29.28 DecompressImage.cs . . . 1257
29.29 DecompressImage.java . . . 1259
29.30 DecompressImage.py . . . 1260
29.31 DecompressImageMultiframe.cs . . . 1260
29.32 DecompressJPEGFile.cs . . . 1262
29.33 DecompressPixmap.java . . . 1263
29.34 DiffFile.cxx . . . 1264
29.35 DiscriminateVolume.cxx . . . 1265
29.36 DumbAnonymizer.py . . . 1269

29.37DumpADAC.cxx . . . 1271
29.38DumpExamCard.cxx . . . 1275
29.39DumpGEMSMovieGroup.cxx . . . 1282
29.40DumpImageHeaderInfo.cxx . . . 1288
29.41DumpPhilipsECHO.cxx . . . 1291
29.42DumpToSQLITE3.cxx . . . 1296
29.43DuplicatePCDE.cxx . . . 1298
29.44ELSCINT1WaveToText.cxx . . . 1300
29.45EncapsulateFileInRawData.cxx . . . 1302
29.46ExtractEncapsulatedFile.cs . . . 1303
29.47ExtractEncryptedContent.cxx . . . 1304
29.48ExtractIconFromFile.cxx . . . 1305
29.49ExtractImageRegion.cs . . . 1306
29.50ExtractImageRegion.java . . . 1308
29.51ExtractImageRegionWithLUT.cs . . . 1309
29.52Extracting_All_Resolution.cxx . . . 1310
29.53ExtractOneFrame.cs . . . 1316
29.54Fake_Image_Using_Stream_Image_Writer.cxx . . . 1317
29.55FileAnonymize.cs . . . 1320
29.56FileAnonymize.java . . . 1321
29.57FileChangeTS.cs . . . 1322
29.58FileChangeTSLossy.cs . . . 1324
29.59FileStreaming.cs . . . 1327
29.60FindAllPatientName.py . . . 1328
29.61FixBrokenJ2K.cxx . . . 1328
29.62FixCommaBug.py . . . 1330
29.63FixJAIBugJPEGLS.cxx . . . 1331
29.64gdcmorthoplanes.cxx . . . 1334
29.65gdcmreslice.cxx . . . 1340
29.66gdcmrtionplan.cxx . . . 1342
29.67gdcmrtplan.cxx . . . 1347
29.68gdcmscene.cxx . . . 1351
29.69gdcmtexture.cxx . . . 1352
29.70gdcmvolume.cxx . . . 1354
29.71GenAllVR.cxx . . . 1355
29.72GenerateDICOMDIR.cs . . . 1358
29.73GenerateRTSTRUCT.cxx . . . 1358

29.74GenerateStandardSOPClasses.cxx . . . 1361
29.75GenFakeIdentifyFile.cxx . . . 1362
29.76GenFakeImage.cxx . . . 1365
29.77GenLongSeqs.cxx . . . 1366
29.78GenSeqs.cxx . . . 1368
29.79GetArray.cs . . . 1369
29.80GetJPEGSamplePrecision.cxx . . . 1370
29.81GetPortionCSAHeader.py . . . 1372
29.82GetSequenceUltrasound.cxx . . . 1373
29.83GetSubSequenceData.cxx . . . 1375
29.84headsq2dcm.py . . . 1377
29.85HelloActiviz.cs . . . 1378
29.86HelloActiviz2.cs . . . 1379
29.87HelloActiviz3.cs . . . 1381
29.88HelloActiviz4.cs . . . 1381
29.89HelloActiviz5.cs . . . 1382
29.90HelloSimple.java . . . 1383
29.91HelloVizWorld.cxx . . . 1384
29.92HelloVTKWorld.cs . . . 1385
29.93HelloVTKWorld.java . . . 1386
29.94HelloVTKWorld2.cs . . . 1387
29.95HelloWorld.cxx . . . 1388
29.96HelloWorld.py . . . 1389
29.97iU22tomultisc.cxx . . . 1390
29.98LargeVRDSExplicit.cxx . . . 1391
29.99MagnifyFile.cxx . . . 1393
29.100ManipulateFile.cs . . . 1394
29.101ManipulateFile.py . . . 1395
29.102ManipulateSequence.py . . . 1397
29.103MergeFile.py . . . 1398
29.104MergeTwoFiles.cxx . . . 1398
29.105MetaImageMD5Activiz.cs . . . 1400
29.106MIPViewer.java . . . 1401
29.107MpegVideoInfo.cs . . . 1403
29.108MPRViewer.java . . . 1408
29.109MPRViewer2.java . . . 1410
29.110MrProtocol.cxx . . . 1414

29.111 NewSequence.cs . . . 1421
29.112 NewSequence.py . . . 1422
29.113 offscreenimage.cxx . . . 1423
29.114 PatchFile.cxx . . . 1424
29.115 PhilipsPrivateRescaleInterceptSlope.py . . . 1426
29.116 PlaySound.py . . . 1427
29.117 pmsct_rgb1.cxx . . . 1428
29.118 PrivateDict.py . . . 1431
29.119 PublicDict.cxx . . . 1432
29.120 QIDO-RS.cxx . . . 1433
29.121 ReadAndDumpDICOMDIR.cxx . . . 1433
29.122 ReadAndDumpDICOMDIR.py . . . 1437
29.123 ReadAndPrintAttributes.cxx . . . 1439
29.124 ReadExplicitLengthSQIVR.cxx . . . 1441
29.125 ReadFiles.java . . . 1441
29.126 ReadGEMSSDO.cxx . . . 1442
29.127 ReadMultiTimesException.cxx . . . 1445
29.128 ReadSeriesIntoVTK.java . . . 1446
29.129 ReadUTF8QtDir.cxx . . . 1447
29.130 RefCounting.cs . . . 1448
29.131 ReformatFile.cs . . . 1449
29.132 RemovePrivateTags.py . . . 1450
29.133 RescaleImage.cs . . . 1451
29.134 reslicesphere.cxx . . . 1452
29.135 ReWriteSCAsMR.py . . . 1460
29.136 rle2img.cxx . . . 1461
29.137 rtstructapp.cxx . . . 1464
29.138 ScanDirectory.cs . . . 1465
29.139 ScanDirectory.java . . . 1466
29.140 ScanDirectory.py . . . 1470
29.141 SendFileSCU.cs . . . 1471
29.142 SimplePrint.cs . . . 1471
29.143 SimplePrintPatientName.cs . . . 1472
29.144 SimpleScanner.cxx . . . 1473
29.145 SortImage.cxx . . . 1475
29.146 SortImage.py . . . 1476
29.147 SortImage2.cs . . . 1477

29.148 StandardizeFiles.cs . . . 1478
29.149 StreamImageReaderTest.cxx . . . 1479
29.150 TestByteSwap.cxx . . . 1483
29.151 TestReader.cxx . . . 1485
29.152 TestReader.py . . . 1486
29.153 threadgdcm.cxx . . . 1487
29.154 TraverseModules.cxx . . . 1490
29.155 uid_unique.cxx . . . 1491
29.156 VolumeSorter.cxx . . . 1492
29.157 WriteBuffer.py . . . 1494

Index **1497**

Chapter 1

GDCM Documentation

This is the developpers documentation.

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

`http://gdcm.sourceforge.net/2.4/gdcm-2.4.4.pdf`

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

`http://gdcm.sourceforge.net/2.4/gdcm-2.4.4-doc.tar.gz`

Author

Mathieu Malaterre

Chapter 2

off-screen rendering of DICOM images

2.1 SYNOPSIS

```
gdcm2pnm [options] file-in bitmap-out
```

2.2 DESCRIPTION

The **gdcm2pnm** command line program takes as input a DICOM file and produces a rendered bitmap file.

2.3 PARAMETERS

```
file-in   DICOM input filename

bitmap-out  Bitmap output filename
```

2.4 OPTIONS

2.4.1 OPTIONS

2.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information
```

```
  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

2.5 Simple usage

gdcm2pnm will take as input DICOM and render it into a bitmap file using the window/level attributes value.

```
$ gdcm2pnm input.dcm output.png
```

It is much different from the gdcmraw or gdcmimg command line tool as it will render a DICOM image. This means that the output will be rendered in 8bits ready for display.

2.6 SEE ALSO

gdcm2vtk(1), **gdcmimg**(1)

2.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 3

Convert a file supported by VTK into DICOM.

3.1 SYNOPSIS

```
gdcm2vtk [options] file-in file-out
```

3.2 DESCRIPTION

The **gdcm2vtk** takes as input any file supported by VTK (including DICOM file) and will generate as output a DICOM file.

3.3 PARAMETERS

```
file-in   input filename (DICOM or VTK supported)

file-out  output filename (DICOM or VTK supported)
```

3.4 OPTIONS

3.4.1 OPTIONS

```
     --force-rescale    force rescale.
     --force-spacing    force spacing.
     --palette-color    when supported generate a PALETTE COLOR file.
     --argb             when supported generate a ARGB file.
     --compress         when supported generate a compressed file.
     --use-vtkdicom     Use vtkDICOMImageReader (instead of GDCM).
     --modality         set Modality.
     --lower-left       set lower left.
     --shift            set shift.
     --scale            set scale.
     --compress         set compression (MetaIO).
  -T --study-uid        Study UID.
  -S --series-uid       Series UID.
     --root-uid         Root UID.
```

3.4.2 compression options

```
-J --jpeg             Compress image in jpeg.
-K --j2k              Compress image in j2k.
-L --jpegls           Compress image in jpeg-ls.
-R --rle              Compress image in rle (lossless only).
```

3.4.3 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

3.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

3.5 DESCRIPTION

Convert a file supported by VTK into DICOM.

Typical usage is:

```
$ gdcm2vtk inputfile output.dcm
```

It uses the internal factory mechanism of VTK to recognize a file (CanRead function). See VTK supported file here:

```
What image file formats can VTK read and write? http://www.vtk.org/Wiki/VTK_FAQ#What_image_file_formats_can_VTK
```

If your input file has 4 components, the 4th comp (alpha) will be removed from the output file as DICOM does not support alpha component anymore (see –argb option).

Special care was taken for the following file format:

1. DICOM: Direction Cosines and vtkMedicalImageInformation are passed to the output
2. BMP: The file can be saved with a Lookup Table (see --palette-color)
3. GE Signa: vtkMedicalImageProperties is passed to the output
4. MINC: Direction Cosines is passed to the output
5. TIFF: vtkTIFFReader is currently in bad shape in VTK (different behavior in VTK 5.2 and git/master). Only u

3.5.1 CONVERT MetaImage (mhd, mha)

```
$ gdcm2vtk inputfile output.mha
```

This command will convert the input DICOM file: inputfile into a MetaImage .mha file. Same goes for .mhd file.

3.5.2 CONVERT MHA/MHD

```
$ gdcm2vtk inputfile output.mha
```

or

```
$ gdcm2vtk inputfile output.mhd
```

This command will convert the input DICOM file: inputfile into a MetaImageData .mha/.mhd file.

3.5.3 CONVERT VTI

```
$ gdcm2vtk inputfile output.vti
```

This command will convert the input DICOM file: inputfile into a XML VTK ImageData .vti file.

3.5.4 CONVERT VTK

```
$ gdcm2vtk inputfile output.vtk
```

This command will convert the input DICOM file: inputfile into an old VTK Structured PointSets .vtk file.

3.6 CONVERT DICOM

```
$ gdcm2vtk input.dcm output.dcm
```

vtkGDCMImageReader will be used to read in a DICOM file, not the default vtkDICOMImageReader. See option –use-vtkdicom to use vtkDICOMImageReader.

3.7 RoundTrip DICOM to MHD to DICOM

```
$ gdcm2vtk input_ybr.dcm output.mhd
$ gdcm2vtk --modality US --imageformat 7 output.mhd output.dcm
```

The above section shows how to convert a DICOM using the Photometric Interpretation of YBR_FULL (or even YBR↩ _FULL_422 is lossy) into another file format: MetaImage (mhd). Since this file format does not handle color space, we have to explicitly set it using the –imageformat command line option. The –modality command line option is required in this case since the default Secondary Capture Image Storage Class family does not allow for YBR Photometric Interpretation.

3.8 gdcm2vtk notes

IMPORTANT NOTE: The internal VTK structured will be filled from the input DICOM, and then pass to the output DIC↩ OM writer. Some information might be lost during the conversion DICOM to VTK to DICOM. This option is mostly used to test the vtkGDCMImageReader/vtkGDCMImageWriter combination.

IMPORTANT NOTE: When converting from a lossy format such as JPEG, the information of lossiness is important. The output DICOM will contains the required Lossy Image Compression attribute that indicates that image was lossy-compressed somewhere along the pipeline. See also gdcmimg (better handling of JPEG in general).

IMPORTANT NOTE: When using –use-vtkdicom the output DICOM file will always be written as MR Image Storage as this information is not available from the reader itself. This allow setting the Image Orientation (Patient) properly.

3.9 SEE ALSO

gdcmdump(1), **gdcmviewer**(1), **gdcmimg**(1)

3.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 4

Tool to anonymize a DICOM file.

4.1 SYNOPSIS

```
gdcmanon [options] file-in file-out
gdcmanon [options] dir-in  dir-out
```

4.2 DESCRIPTION

The **gdcmanon** tool is an implementation of PS 3.15 / E.1 / Basic Application Level Confidentiality Profile (Implementation of E.1.1 De-identify & E.1.2 Re-identify)

This tool is split into two very different operating mode:

- An implementation of PS 3.15, see -e and -d flags
- A dumb mode, see –dumb

Dumb mode and PS 3.15 do not work well together, you should really only use one type of anonymization. In case of doubt, avoid using –dumb.

In order to use the PS 3.15 implementation (-d & -e flag), you'll need a certificate to do de-identification operations, and the associated private key to do the re-identification operation. If you are only doing a one-shot anonymization and do not need to properly re-identify the DICOM file, you can safely discard the private key and only keep the certificate. See OpenSSL section below for an example on how to generate the private key/certificate pair.

gdcmanon will exit early if OpenSSL was not configured/build properly into the library (see GDCM_USE_SYSTEM_O↩ PENSSL in cmake).

4.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

or

```
file-in   DICOM input directory

file-out  DICOM output directory
```

4.4 OPTIONS

You need to specify at least one operating mode, from the following list (and only one):

4.4.1 Required parameters

```
-e --de-identify            De-identify DICOM (default)
-d --re-identify            Re-identify DICOM
   --dumb                   Dumb mode anonymizer
```

Warning when operating in dumb mode, you need to also specify an operation to do, such as 'remove' or 'empty' a tag, see below the dumb mode options.

4.4.2 OPTIONS

```
-i --input                  DICOM filename / directory
-o --output                 DICOM filename / directory
-r --recursive              recursively process (sub-)directories.
   --continue               Do not stop when file found is not DICOM.
   --root-uid               Root UID.
   --resources-path         Resources path.
-k --key                    Path to RSA Private Key.
-c --certificate            Path to Certificate.
```

4.4.3 encryption options

```
   --des            DES.
   --des3           Triple DES.
   --aes128         AES 128.
   --aes192         AES 192.
   --aes256         AES 256.
```

4.4.4 dumb mode options

```
   --empty   %d,%d           DICOM tag(s) to empty
   --remove  %d,%d           DICOM tag(s) to remove
   --replace %d,%d,%s        DICOM tag(s) to replace
```

4.4.5 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

4.4.6 environment variable

```
GDCM_ROOT_UID Root UID
GDCM_RESOURCES_PATH path pointing to resources files (Part3.xml, ...)
```

4.5 Typical usage

4.5.1 De-identification (anonymization, encrypt)

The only thing required for this operation is a certificate file (in PEM format).

```
$ gdcmanon --certificate certificate.pem -e original.dcm original_anonymized.dcm
```

You can use –asn1 option from gdcmdump to dump the generated DataSet as ASN1 structure (see **gdcmdump**(1) for example).

4.5.2 Re-identification (de-anonymization,decrypt)

The only thing required for this operation is a private key (in PEM format). It is required that the private key used for the re-identification process, was the actual private key used to generate the certificate file (certificate.pem) used during the de-identification step.

```
$ gdcmanon --key privatekey.pem -d original_anonymized.dcm original_copy.dcm
```

You can then check that original.dcm and original_copy.dcm are identical.

4.5.3 Multiple files caveat

It is very important to understand the following section, when anonymizing more than one single file. When anonymizing multiple DICOM files, you are required to use the directory input. You cannot call multiple time the gdcmanon command line tool. Indeed the tool stores in memory during the process only a hash table of conversion so that each time a particular value is found it get always replaced by the same de-identified value (think: consistent Series Instance UID).

4.5.4 Dumb mode

This functionality is not described in the DICOM standard. Users are advised that improper use of that mode is not recommended, meaning that important tag can be emptied/removed/replaced resulting in illegal/invalid DICOM file. Only use when you know what you are doing. If you delete a Type 1 attribute, chance is that your DICOM file will be not accepted in most DICOM third party viewer. Unfortunately this is often this mode that is implemented in popular DICOM Viewer, always prefer what the DICOM standard describes, and avoid the dumb mode.

The following example shows how to use dumb mode and achieve 5 operations at the same time:

- Empty the tag (0010,0010) Patient's Name,
- Empty the tag (0010,0020) Patient ID,
- Remove the tag (0010,0040) Patient's Sex
- Remove the tag (0010,1010) Patient's Age

- Replace the tag (0010,1030) Patient's Weight with the value '10'

You are required to check which DICOM attribute is Type 1 and Type 1C, before trying to **'Empty'** or **'Remove'** a particular DICOM attribute. For the same reason, you are required to check what are valid value in a replace operation.

```
$ gdcmanon --dumb --empty 10,10 --empty 10,20 --remove 10,40 --remove 10,1010 --replace 10,1030,10 012345.002.050
```

Multiple operation of –dumb mode can take place, just reuse the output of the previous operation. Always use gdcmdump on the input and output file to check what was actually achieved. You can use a diff program to check only what changed (see **gdcmdiff**(1) for example).

4.5.4.1 Irreversible Anonymization

In some very rare cases, one would want to anonymize using the PS 3.15 mode so as to take benefit of the automatic conversion of all content that could contain Patient related information.

In the end all Patient related information has been removed and has been secretly stored in the 0400,0500 DICOM attribute. However to make sure that no-one ever try to break that security using brute-force algorithm, one want want to remove completely this DICOM attribute. This will make the DICOM:

- Completely free of any Patient related information (as per PS 3.15 specification)
- Remove any mean of people to brute force attack the file to find out the identity of the Patient

In this case one could simply do, as a first step execute the reversible anonymizer:

```
$ gdcmanon -c certificate.pem input.dcm anonymized_reversible.dcm
```

and now completely remove the DICOM attribute containing the secretly encrypted Patient related information:

```
$ gdcmanon --dumb --remove 400,500 --remove 12,62 --remove 12,63 anonymized_reversible.dcm anonymized_irreversibl
```

Remarks

> As mentionned in DICOM Sup 142, this anonymization is preferred over de-identification since: It is not required that the Encrypted Attributes Data Set be created; indeed, there may be circumstances where the Dataset is expected to be archived long enough that any contemporary encryption technology may be inadequate to provide long term protection against unauthorized recovery of identification

4.6 OpenSSL

On most system you can have access to OpenSSL to generate the Private Key/Certificate pair.

4.6.1 Generating a Private Key

Command line to generate a rsa key (512bit)

```
$ openssl genrsa -out CA_key.pem
```

Command line to generate a rsa key (2048bit)

```
$ openssl genrsa -out CA_key.pem 2048
```

Command line to generate a rsa key (2048bit) + passphrase

```
$ openssl genrsa -des3 -out CA_key.pem 2048
```

4.6.2 Generating a Certificate

From your previously generated Private Key, you can now generate a certificate in PEM (DER format is currently not supported).

```
$ openssl req -new -key CA_key.pem -x509 -days 365 -out CA_cert.cer
```

4.7 DICOM Standard:

Page to the DICOM Standard:

`http://dicom.nema.org/`

The DICOM Standard at the time of releasing gdcmanon is:

`ftp://medical.nema.org/medical/dicom/2008/`

Direct link to PS 3.15-2008:

`ftp://medical.nema.org/medical/dicom/2008/08_15pu.pdf`

4.8 Warnings

Certain attributes may still contains Protected Health Information (PHI) after an anonymization step. This is typically the case for Patient's Address (0010,1040). The reason is that this particular attribute is not supposed to be in the composite IODs in the first place. DICOM Supp 142 includes it (however gdcmanon does not implement it).

4.9 SEE ALSO

gdcmconv(1), **gdcmdump**(1), **gdcmdiff**(1), **openssl**(1), **dumpasn1**(1)

4.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 5

Tool to convert DICOM to DICOM.

5.1 SYNOPSIS

```
gdcmconv [options] file-in file-out
```

5.2 DESCRIPTION

The **gdcmconv** command line program takes as input a DICOM file (file-in) and process it to generate an output DICOM file (file-out). The command line option dictate the type of operation(s) gdcmconv will use to generate the output file.

5.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

5.4 OPTIONS

5.4.1 PARAMETERS

```
  -i --input      DICOM filename
  -o --output     DICOM filename
```

5.4.2 OPTIONS

```
  -X --explicit            Change Transfer Syntax to explicit.
  -M --implicit            Change Transfer Syntax to implicit.
  -U --use-dict            Use dict for VR (only public by default).
     --with-private-dict   Use private dict for VR (advanced user only).
  -C --check-meta          Check File Meta Information (advanced user only).
     --root-uid            Root UID.
     --remove-gl           Remove group length (deprecated in DICOM 2008).
     --remove-private-tags Remove private tags.
     --remove-retired      Remove retired tags.
```

5.4.3 image options

```
-l --apply-lut                      Apply LUT (non-standard, advanced user only).
-P --photometric-interpretation %s  Change Photometric Interpretation (when possible).
-w --raw                            Decompress image.
-d --deflated                       Compress using deflated (gzip).
-J --jpeg                           Compress image in jpeg.
-K --j2k                            Compress image in j2k.
-L --jpegls                         Compress image in jpeg-ls.
-R --rle                            Compress image in rle (lossless only).
-F --force                          Force decompression/merging before recompression/splitting.
   --generate-icon                  Generate icon.
   --icon-minmax %d,%d              Min/Max value for icon.
   --icon-auto-minmax               Automatically commpute best Min/Max values for icon.
   --compress-icon                  Decide whether icon follows main TransferSyntax or remains uncompressed.
   --planar-configuration [01]      Change planar configuration.
-Y --lossy                          Use the lossy (if possible) compressor.
-S --split %d                       Write 2D image with multiple fragments (using max size)
```

5.4.4 JPEG options

```
-q --quality %*f           set quality.
```

5.4.5 JPEG-LS options

```
-e --lossy-error %*i       set error.
```

5.4.6 J2K options

```
-r --rate    %*f           set rate.
-q --quality %*f           set quality.
-t --tile %d,%d            set tile size.
-n --number-resolution %d  set number of resolution.
   --irreversible          set irreversible.
```

5.4.7 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

5.4.8 special options

```
-I --ignore-errors   convert even if file is corrupted (advanced users only, see disclaimers).
```

5.4.9 environment variable

```
GDCM_ROOT_UID Root UID
```

5.5 Simple usage

gdcmconv is a great tool to convert broken DICOM implementation into properly parsable DICOM file. Usage is simply:

```
$ gdcmconv input.dcm output.dcm
```

or if you prefer being explicit:

```
$ gdcmconv -i input.dcm -o output.dcm
```

Even though gdcmconv can overwrite directly on the same file (input.dcm = output.dcm), it is recommended that user should first convert into a different file to make sure the bug is properly handled by GDCM.

Typical cases where you would want to use gdcmconv in its simple form:

- convert non-cp246 conforming file into conforming cp246,
- convert implicit little endian transfer syntax file meta header into proper explicit little endian transfer syntax,
- convert the GE-13 bytes bug,
- convert dual syntax file: implicit/explicit,
- convert Philips dual Little Endian/Big Endian file,
- convert GDCM 1.2.0 broken UN-2-bytes fields,
- &...
- All other broken files listed in the supported section.

When no option other is used, only the dataset is inspected. So encapsulated Pixel Data, for instance, is not inspected for well known bugs.

When doing this kind of work, this is usually a good idea to perform some kind of quality control, see gdcmconv Quality Control section (down below).

5.6 Typical usage

5.6.1 File Meta Header

Running

```
$ gdcmconv input.dcm output.dcm
```

Is not enough to recompute file meta header, when input file is buggy. You may want to use: –check-meta

```
$ gdcmconv --check-meta input.dcm output.dcm
```

See typical cases such as: GE_DLX-8-MONO2-PrivateSyntax.dcm or PICKER-16-MONO2-No_DicomV3_Preamble.↩dcm from gdcmData.

5.6.2 Conversion to Explicit Transfer Syntax

To convert a file that was written using Implicit Transfer Syntax into Explicit Transfer Syntax simply use:

```
$ gdcmconv --explicit uncompressed.dcm compressed.dcm
```

5.6.3 Compressing to lossless JPEG

To compress an uncompressed DICOM file to a JPEG Lossless encapsulated format:

```
$ gdcmconv --jpeg uncompressed.dcm compressed.dcm
```

5.6.4 Compressing to lossy JPEG

To compress an uncompressed DICOM file to a JPEG Lossy encapsulated format:

```
$ gdcmconv --lossy --jpeg -q 90 uncompressed.dcm compressed.dcm
```

Note:

```
  -q is just one of the many way to specify lossy quality, you need to inspect the other cmd line flag to specify
```

5.6.5 Compressing to lossless JPEG-LS

To compress an uncompressed DICOM file to a JPEG-LS Lossless encapsulated format:

```
$ gdcmconv --jpegls uncompressed.dcm compressed.dcm
```

5.6.6 Compressing to lossy JPEG-LS

To compress an uncompressed DICOM file to a JPEG-LS Lossy encapsulated format:

```
$ gdcmconv --lossy --jpegls -e 2 uncompressed.dcm lossy_compressed.dcm
```

Note:

-e (or –lossy-error) means that the maximum tolerate error is 2 for each pixel value

5.6.7 Compressing to lossless J2K

To compress an uncompressed DICOM file to a JPEG-2000 Lossless encapsulated format:

```
$ gdcmconv --j2k uncompressed.dcm compressed.dcm
```

5.6.8 Compressing to lossy J2K

To compress an uncompressed DICOM file to a JPEG-2000 Lossy encapsulated format:

```
$ gdcmconv --lossy -q 55,50,45 --j2k uncompressed.dcm lossy_compressed.dcm
```

Note:

```
  -q is just one of the many way to specify lossy quality, you need to inspect the other cmd line flag to specify
```

5.6.9 Compressing to lossless RLE

To compress an uncompressed DICOM file to a RLE Lossless encapsulated format:

```
$ gdcmconv --rle uncompressed.dcm compressed.dcm
```

There is no such thing as lossy RLE compression.

5.6.10 Split encapsulated DICOM:

To split an encapsulated stream into smaller chunk (1024 bytes each):

```
$ gdcmconv --split 1024 rle.dcm rle_1024.dcm
```

If an odd number of bytes is passed it will be rounded down to the next even number (eg. 1025 -> 1024) since DICOM only allow even number for Value Length.

5.6.11 Forcing (re)compression

Sometime it is necessary to use the –force option. By default when user specify –j2k and input file is already in JPEG 2000 encapsulated DICOM format then no operation takes places. By using –force you make sure that (re)compression operation takes places.

Real life example of why you would use –force:

- When Pixel Data is missing data / is padded with junk
- When you would like to make sure GDCM can handle decompression & recompression cycle

5.6.12 Decompressing a Compressed DICOM

```
$ gdcmconv --raw compressed.dcm uncompressed.dcm
```

5.6.13 Compressing an uncompressed Icon

By default when compressing a DICOM Image file, gdcmconv will not compress the icon. A user option needs to be turned on to explicitly force the compression of the Icon Image Sequence Pixel Data

For example, by default we will not compress the Icon Image Sequence Pixel Data attribute:

```
$ gdcmconv --jpeg gdcmData/simpleImageWithIcon.dcm uncompressed_icon.dcm
```

In the following example we will explicitly compress the Icon Image Sequence Pixel Data attibute. In that case the same Transfer Syntax is being used for both the main Pixel Data and the Pixel Data from the Icon Image Sequence:

```
$ gdcmconv --jpeg --compress-icon gdcmData/simpleImageWithIcon.dcm compressed_icon.dcm
```

5.6.14 Generating an Icon

For some application it might be necessary to produce a small preview of the main image to be able to quickly load that short preview instead of the main image. In that case:

```
$ gdcmconv --raw --generate-icon gdcmData/test.acr test_icon.dcm
```

In some cases the main Pixel Data element is expressed as pixel defined on 16bits. Since Icon can only store at most pixel of size 8bits, a rescale operation needs to take place. In order to properly select a better interval for doing the rescale operation user can specify the min max used for the rescale operation:

```
$ gdcmconv --raw --generate-icon --icon-minmax 0,192 gdcmData/012345.002.050.dcm icon_minmax.dcm
```

5.6.15 Changing the planar Configuration

Often RLE files are compressed using a different Planar Configuration (RRR ... GGG... BBB...) instead of the usual triplet (RGB ... RGB ... RGB). So upon decompression the Planar Configuration is 1. This is perfectly legal in DICOM, however this is unconventional, and thus it may be a good idea to also change the planar configuration and set it to the default :

```
$ gdcmconv --raw --planar-configuration 0 compressed.dcm uncompressed1.dcm
```

To reinvert the planar configuration of file 'uncompressed1.dcm', simply do:

```
$ gdcmconv --raw --planar-configuration 1 uncompressed1.dcm uncompressed2.dcm
```

5.7 Lossless Conversion

When talking about lossless conversion, there is an ambiguity that need to be understood. To achieve higher compression ratio, the RGB color space is usually not used, in favor of a YBR one. Changing from one color space to the other is (bit level) not lossless.

For more detail, see what are the true lossless transformations as described:

`http://gdcm.sourceforge.net/wiki/index.php/Color_Space_Transformations`

5.8 Quality Control

One important part when using gdcmconv it to have a way to quality control the output.

You can use 3rd party tool to check the output of gdcmconv is correct.

5.8.1 DCMTK / dicom3tools

Using another DICOM implementation such as the one from DCMTK or dicom3tools can be a good process to check the output of gdcmconv.

- For DCMTK use: dcmdump
- For dicom3tools use: dcdump

For reference, gdcmconv –raw will act as dcmdjpeg +cn +px, since it never tries to convert color space.

5.8.2 VIM: vimdiff

You can setup your favorite editor to compare the output, for instance in vim:

```
autocmd BufReadPre *.dcm set ro
autocmd BufReadPost *.dcm silent %!dcmdump -M +uc "%"
```

then simply do:

```
$ vimdiff input.dcm output.dcm
```

5.8.3 vbindiff

On UNIX you can visually compare binary file using the vbindiff command:

```
$ vbindiff input.dcm output.dcm
```

5.9 SEE ALSO

gdcmdump(1), **gdcmraw**(1), **gdcminfo**(1), **gdcmdiff**(1)

5.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 6

dumps differences of two DICOM files

6.1 SYNOPSIS

```
gdcmdiff [options] file1 file2
```

6.2 DESCRIPTION

The **gdcmdiff** command line program takes as input two DICOM files: file1 and file2.

6.3 PARAMETERS

```
file1   DICOM input filename

file2   DICOM output filename
```

6.4 OPTIONS

6.4.1 OPTIONS

```
  -m      --meta          Compare metainformation. Default is off.
  -t <n>  --truncate <n>  String values trimmed to n characters.
```

6.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information
```

```
  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

6.5 Simple usage

gdcmdiff is a great tool to produce a diff in between two DICOM files. Usage is simply:

```
$ gdcmdiff input1.dcm input2.dcm
```

6.6 SEE ALSO

gdcmdump(1), **gdcminfo**(1)

6.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 7

dumps a DICOM file, it will display the structure and values contained in the specified DICOM file.

7.1 SYNOPSIS

```
gdcmdump [options] dcm_file
gdcmdump [options] dcm_directory
```

7.2 DESCRIPTION

The **gdcmdump** command line program dumps a DICOM file to the console. For those familiar with dcmdump (DCMTK) output, gdcmdump has some minor differences. Namely:

- For Implicit Transfer Syntax gdcmdump will print ?? instead of the dictionary VR

gdcmdump has a limited private dictionary that is used to lookup private element whenever possible.

7.3 PARAMETERS

```
dcm_file        DICOM input filename
dcm_directory   DICOM input directory
```

7.4 OPTIONS

7.4.1 OPTIONS

```
  -x --xml-dict       generate the XML dict (only private elements for now).
  -r --recursive      recursive (input is a directory)
  -d --dump           dump value (limited use).
  -p --print          print value instead of simply dumping (default).
  -c --color          print in color.
  -C --csa            print SIEMENS CSA Header (0029,[12]0,SIEMENS CSA HEADER).
  -P --pdb            print GEMS Protocol Data Block (0025,1b,GEMS_SERS_01).
     --elscint        print ELSCINT Protocol Information (01f7,26,ELSCINT1).
     --vepro          print VEPRO Protocol Information (0055,20,VEPRO VIF 3.0 DATA).
```

```
                        or VEPRO Protocol Information (0055,20,VEPRO VIM 5.0 DATA).
    --sds             print Philips MR Series Data Storage (1.3.46.670589.11.0.0.12.2) Information (2005,32,Philip
-A --asn1             print encapsulated ASN1 structure >(0400,0520).
    --map-uid-names   map UID to names.
```

7.4.2 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

7.4.3 special options

```
-I --ignore-errors   dumps even if file is corrupted (advanced users only, see disclaimers).
```

7.5 Typical usage

7.5.1 Printing Implicit Transfer Syntax

The VR are not found in the file, thus are presented with a "(??)", and right next to it (if found) the correct VR.

Eg.:

```
$ gdcmdump GE_DLX-8-MONO2-PrivateSyntax.dcm
```

```
# Dicom-File-Format
\&...
(0008,0000) ?? (UL) 434                                          # 4,1 Generic Group Length
(0008,0005) ?? (CS) [ISO_IR 100]                                 # 10,1-n Specific Character Set
(0008,0008) ?? (CS) [ORIGINAL\\PRIMARY\\SINGLE PLANE ]             # 30,2-n Image Type
(0008,0016) ?? (UI) [1.2.840.10008.5.1.4.1.1.12.1]               # 28,1 SOP Class UID
(0008,0018) ?? (UI) [1.2.840.113619.2.16.1.0.906539207.1.24207]  # 42,1 SOP Instance UID
(0008,0020) ?? (DA) [19980923]                                   # 8,1 Study Date
(0008,0021) ?? (DA) [19980923]                                   # 8,1 Series Date
(0008,0022) ?? (DA) [19980923]                                   # 8,1 Acquisition Date
(0008,0023) ?? (DA) [19980923]                                   # 8,1 Content Date
(0008,0030) ?? (TM) [101229.000]                                 # 10,1 Study Time
(0008,0031) ?? (TM) [101229.000]                                 # 10,1 Series Time
(0008,0032) ?? (TM) [102653.000]                                 # 10,1 Acquisition Time
(0008,0033) ?? (TM) [102653.000]                                 # 10,1 Content Time
\&...
```

7.5.2 Print Private Attributes

GDCM has a limited private dictionary. Whenever possible, it will try to lookup the private data element.

```
$ gdcmdump 012345.002.050.dcm

\&...
(0009,0010) LO [GEMS_IDEN_01]                                    # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ]                                  # 14,1 Full fidelity
(0009,1002) SH [MRCV]                                            # 4,1 Suite id
(0009,1004) SH [SIGNA ]                                          # 6,1 Product id
(0009,1027) SL 985968524                                         # 4,1 Image actual date
(0009,1030) SH [19356UMR2 ]                                      # 10,1 Service id
(0009,1031) SH [999 ]                                            # 4,1 Mobile location number
(0009,10e3) UI [1.2.840.113619.1.1.4.1762386977]                 # 32,1 Equipment UID
(0009,10e6) SH [08]                                              # 2,1 Genesis Version - now
(0009,10e7) UL 2757786872                                        # 4,1 Exam Record checksum
(0009,10e9) SL 985968523                                         # 4,1 Actual series data time stamp
\&...
(0019,0000) UL 1208                                              # 4,1 Generic Group Length
(0019,0010) LO [GEMS_ACQU_01]                                    # 12,1 Private Creator
(0019,100f) DS [424.399994]                                      # 10,1 Horiz. Frame of ref.
(0019,1011) SS 0                                                 # 2,1 Series contrast
\&...
(0019,10e0) DS [0.000000]                                        # 8,1 User data 24 {# DTI Diffusion Dir., relea
(0019,10e2) DS [0.000000]                                        # 8,1 Velocity Encode Scale
(0019,10f2) SS 0                                                 # 2,1 Fast phases
(0019,10f9) DS [98]                                              # 2,1 Transmit gain
\&...
(0021,0000) UL 372                                               # 4,1 Generic Group Length
(0021,0010) LO [GEMS_RELA_01]                                    # 12,1 Private Creator
(0021,1003) SS 0                                                 # 2,1 Series from which Prescribed
\&...
```

7.5.3 SIEMENS CSA Header

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical SIEMENS MR DICOM file.

Eg.:

```
$ gdcmdump --csa MR_SIEMENS_forceLoad29-1010_29-1020.dcm

(0029,0010)siemens csa header
Image shadow data (0029,xx10)

0 - 'EchoLinePosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '64      '
1 - 'EchoColumnPosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '64      '
2 - 'EchoPartitionPosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '32      '
3 - 'UsedChannelMask' VM 1, VR UL, SyngoDT 9, NoOfItems 6, Data '255     '
4 - 'Actual3DImaPartNumber' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
5 - 'ICE_Dims' VM 1, VR LO, SyngoDT 19, NoOfItems 6, Data 'X_1_1_1_1_1_1_31_1_1_1_1_19'
6 - 'B_value' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '0       '
7 - 'Filter1' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
8 - 'Filter2' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
\&...
```

7.5.4 GEMS Protocol Data Block

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical GEMS MR DICOM file.

Protocol Data Block : 0025,xx1b,GEMS_SERS_01

```
$ gdcmdump --pdb GE_MR_0025xx1bProtocolDataBlock.dcm


ENTRY "Head First"
POSITION "Supine"
ANREF "NA"
COIL "HEAD"
PLANE "OBLIQUE"
SEDESCFLAG "1"
SEDESC "AX FSE T2"
IMODE "2D"
PSEQ "FSE-XL"
IOPT "FC, EDR, TRF, Fast"
PLUG "22"
FILTCHOICE "None"
BWRT "-1"
TRICKSIMG "1"
TAG_SPACE "7"
TAG_TYPE "None"
\&...
```

7.5.5 ELSCINT Protocol Information

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical ELSCINT CT DICOM file.

ELSCINT Protocol Information: (01f7,26,ELSCINT1)

```
$ gdcmdump --elscint ELSCINT1_ProtocolInformation.dcm


ELSCINT1 Dumping info from tag (01f7,26,elscint1)
ELSCINT1/Item name: []
  ApprovedStep [yes]
  RefSurview [1\0]
  STD-first-img-pos [11.5]
  current-step [yes]
  ntimed-steps [0]
  orig-n-slices [390]
  protocol-file [Head_Multi_1032_usr.proc]
  protocol-name [FACE-TRAUMA/Head/Hx]
  protocol-path [/usr/diamond.root/spr/]
  protocol-step [1]
  protocol-version [2.51]

ELSCINT1/Item name: [doseright]
  ACS [n/a]
  ACS-bed-position [0]
  ACS-calc-mas [0]
  ACS-iq-parameter [0]
  ACS-learn-allowed [no]
  ACS-water-radius [-1.000000]
  ACS-water-radius-scan [-1]
\&...
```

7.5.6 VEPRO Protocol Information

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical VEPRO CT DICOM file.

ELSCINT Protocol Information: (0055,20,VEPRO VIM 5.0 DATA)

```
$ gdcmdump --vepro VEPRO_ProtocolInformation.dcm


VIMDATA2: (0055,20,VEPRO VIM 5.0 DATA)
  ID: VIM
  Version: 5.0
  UserName:
  UserAdress1: Name of Institution
  UserAdress2: Street of Institution
  UserAdress3: City of Institution
  UserAdress4:
  UserAdress5:
  RecDate: 20101001
  RecTime: 211321
  RecPlace:
  RecSource: DICOM Distributor
  DF1: P-09/10-41808
  DF2: Sultana Razia
  DF3: 19411001
  DF4: F
  DF5:
  DF6:
  DF7:
  DF8: CT Scan Brain without Contrast
  DF9: 10/10-0034873
  DF10: 10/10-00348
  DF11:
  DF12:
  DF13:
  DF14:  Head 0.5
  DF15: 4
  DF16:
  DF17:
  DF18:
  DF19:
  DF20:
  StudyUID: 1.2.392.200036.9116.2.6.1.48.1214228007.1285934880.206831
  SeriesUID: 1.2.392.200036.9116.2.6.1.48.1214228007.1285935201.938653
  Modality: CT
```

7.5.7 Philips Private MR Series Data Storage (1.3.46.670589.11.0.0.12.2)

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical Philips Private MR Series Data Storage file.

PMS Series Data Storage (2005,32,Philips MR Imaging DD 002)

```
$ gdcmdump --sds PMS_SeriesDataStorage.dcm


\&...
PMS/Item name: [PDF_CONTROL_GEN_PARS/IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_PREP_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_RECON_PARS/IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_SCAN_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_EXAM_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_HARDWARE_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_PREP_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_SPT_PARS/IEEE_PDF/Y ]
```

```
  SP_scan_resol [256\256] # 2
  SP_pda_profiles [0\0] # 2
  SP_filter [324074] # 1
  SP_analyse_with_iqt [0] # 1
  SP_main_system_type [3] # 1
  SP_gradient_system [6] # 1
  SP_coil_type [2\2\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_coil_id [2\34\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_coil_part [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_act_q [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_act_coil_freq [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_coil_m_pos [255\255\255\0\0\0\0\0\0\0\0\0\0\0\255] # 16
  SP_coil_t_pos [255\128\255\0\0\0\0\0\0\0\0\0\0\0\255] # 16
  SP_surface_coil_con [0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
  SP_proton_freq [127801349] # 1
  SP_tm_result [2\2\2\2\2\2\2\2\2\2\2\2\2\2\2\2] # 16
  SP_f0_result [0] # 1
  SP_as_result [0] # 1
  SP_po_result [0] # 1
  SP_rg_result [0] # 1
  SP_dc_result [0] # 1
  SP_ph_result [0] # 1
\&...
```

7.5.8 Encapsulated ASN1 Structure

This option is mainly used for dumping the ASN1 structure of the encrypted Attribute (0040,0520)

```
$ gdcmdump encrypted.dcm
```

```
\&...
(0400,0500) SQ                                                          # u/l,1 Encrypted Attributes Sequence
  (fffe,e000) na (Item with undefined length)
    (0400,0510) UI [1.2.840.10008.1.2]                                  # 18,1 Encrypted Content Transfer Syntax UID
    (0400,0520) OB 30\\82\\03\\ba\\06\\09\\2a\\86\\48\\55\\04\\08\\13         # 958,1 Encrypted Content
  (fffe,e00d)
(fffe,e0dd)
\&...
```

```
$ gdcmdump --asn1 encrypted.dcm
```

```
    0:d=0  hl=4 l= 954 cons: SEQUENCE
    4:d=1  hl=2 l=   9 prim:  OBJECT            :pkcs7-envelopedData
   15:d=1  hl=4 l= 939 cons:  cont [ 0 ]
   19:d=2  hl=4 l= 935 cons:   SEQUENCE
   23:d=3  hl=2 l=   1 prim:    INTEGER           :00
   26:d=3  hl=4 l= 366 cons:    SET
   30:d=4  hl=4 l= 362 cons:     SEQUENCE
   34:d=5  hl=2 l=   1 prim:      INTEGER           :00
   37:d=5  hl=2 l=  82 cons:      SEQUENCE
   39:d=6  hl=2 l=  69 cons:       SEQUENCE
   41:d=7  hl=2 l=  11 cons:        SET
   43:d=8  hl=2 l=   9 cons:         SEQUENCE
   45:d=9  hl=2 l=   3 prim:          OBJECT            :countryName
   50:d=9  hl=2 l=   2 prim:          PRINTABLESTRING   :AU
   54:d=7  hl=2 l=  19 cons:        SET
   56:d=8  hl=2 l=  17 cons:         SEQUENCE
   58:d=9  hl=2 l=   3 prim:          OBJECT            :stateOrProvinceName
   63:d=9  hl=2 l=  10 prim:          PRINTABLESTRING   :Some-State
   75:d=7  hl=2 l=  33 cons:        SET
   77:d=8  hl=2 l=  31 cons:         SEQUENCE
   79:d=9  hl=2 l=   3 prim:          OBJECT            :organizationName
   84:d=9  hl=2 l=  24 prim:          PRINTABLESTRING   :Internet Widgits Pty Ltd
  110:d=6  hl=2 l=   9 prim:       INTEGER           :AC966D88787A51B4
```

```
  121:d=5  hl=2 l=  13 cons:      SEQUENCE
  123:d=6  hl=2 l=   9 prim:       OBJECT            :rsaEncryption
  134:d=6  hl=2 l=   0 prim:       NULL
  136:d=5  hl=4 l= 256 prim:      OCTET STRING      [HEX DUMP]:822368070285AD756C962ECB973514B291F946...
  396:d=3  hl=4 l= 558 cons:    SEQUENCE
  400:d=4  hl=2 l=   9 prim:     OBJECT            :pkcs7-data
  411:d=4  hl=2 l=  29 cons:     SEQUENCE
  413:d=5  hl=2 l=   9 prim:      OBJECT            :aes-256-cbc
  424:d=5  hl=2 l=  16 prim:      OCTET STRING      [HEX DUMP]:3B49AFE71749F2BFF1519EBAEA95A393
  442:d=4  hl=4 l= 512 prim:     cont [ 0 ]
```

7.6 SEE ALSO

gdcmdump(1), **gdcmraw**(1), **gdcmanon**(1)

7.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 8

Tool to generate a DICOMDIR file from a File-Set.

8.1 SYNOPSIS

```
gdcmgendir [options] file-in file-out
```

8.2 DESCRIPTION

8.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

8.4 OPTIONS

8.4.1 Parameters

8.4.2 OPTIONS

```
  -i --input              DICOM filename or directory
  -o --output             DICOM filename or directory
  -r --recursive          recursive.
     --descriptor         descriptor.
     --root-uid           Root UID.
```

8.4.3 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit
```

```
-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

8.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

8.5 Typical usage

8.6 NOTE

One may have to run some preliminary steps in order to get gdcmgendir to generate the DICOMDIR file. Namely two steps:

- Batch renaming of the DICOM filename into something compatible with ISO 9660 filename convention
- Convert all DICOM file into the Explicit VR Little Endian Uncompressed (1.2.840.10008.1.2.1)

Step 1. can be solved in a numerous way. Eg. on UNIX environment this could either be solved using the mkisofs command line tool. Filenames should not contains any extension since the VR CS does not allow for the '.' character. Only upper case, digit 0-9, the space ' ' and the underscore '_' character are valid in VR CS, with a maximum of 8 bytes. Another simple tool that can be handy is 'rename' in conjunction with 'basename'.

Step 2. can simply be achieved using the gdcmconv command line tool:

```
$ for i in `ls IMG*`; do gdcmconv --raw --force $i /tmp/out/$i; done
```

8.7 SEE ALSO

gdcmconv(1), **gdcmanon**(1), **rename**(1), **mkisofs**(1)

8.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 9

Manipulate DICOM image file.

gdcmimg is a low level tool to allow de-/encapsulation from/to DICOM image. This tool does not understand Transfer Syntax conversion. It will encapsulate the raw data as-is. This has some impact in some cases, see special warnings below.

It is important to note that gdcmimg can only encapsulate proper input file, for instance JPG and or JP2 are accepted since an associated DICOM Transfer Syntax can be found. However input such as TIFF and/or PNG are not, since DICOM does not support those. See instead a tool such as gdcm2vtk.

9.1 SYNOPSIS

```
gdcmimg [options] file-in file-out
```

9.2 DESCRIPTION

The **gdcmimg** command line tool can be used in two fashions:

- 1. Converting a recognized file format into its encapsulated DICOM counterpart,
- 2. Anonymizing a rectangular portion of a DICOM file.

9.3 PARAMETERS

```
file-in   input filename (non-DICOM)

file-out  DICOM output filename
```

9.4 OPTIONS

9.4.1 PARAMETERS

```
  -i --input     Input filename
  -o --output    Output filename
```

9.4.2 OPTIONS

```
   --endian %s       Endianness (LSB/MSB).
-d --depth %d        Depth (Either 8/16/32 or BitsAllocated eg. 12 when known).
   --sign %s         Pixel sign (0/1).
   --spp  %d         Sample Per Pixel (1/3).
-s --size %d,%d      Size.
-C --sop-class-uid   SOP Class UID (name or value).
-T --study-uid       Study UID.
-S --series-uid      Series UID.
   --root-uid        Root UID.
```

9.4.3 fill options

```
-R --region %d,%d    Region.
-F --fill %d         Fill with pixel value specified.
```

9.4.4 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

9.4.5 environment variable

```
GDCM_ROOT_UID Root UID
```

9.5 Supported File Format (appropriate file extension) gdcmimg

will base it's conversion process based on the file extension. Follows the list of recognized file extension. When no extension is found, DICOM file is assumed.

input format

```
* RAW       (raw, rawl, gray, rgb)
* RLE       (rle)
* PNM       (pgm, pnm, ppm)
* JPEG-LS   (jls)
* JPEG 2000 (jp2, j2k, j2c, jpx, jpc)
* JPEG      (jpg, jpeg, ljpg, ljpeg)
* DICOM     ()
```

output format:

```
  * PGM        (pgm, pnm, ppm)
  * DICOM      ()
```

For RAW file format, you should take special care of the –endian option. For the (old) JPEG file format, both the lossy and lossless format are supported, user should pay attention to the –sign option. For file format such as RLE or RAW, user is expected to fill in information required to find the dimension and type of input data as there is no other way to find this information. For all other file format, the properties are derived from the file format itself.

PNM file are supposed to be big endian (important for depth $>$ 8)

9.6 Typical usage

9.6.1 Remove a rectangular part of the image

To fill the region [0,100]x[0,100] of a DICOM image simply do:

```
$ gdcmimg --fill 0 --region 0,100,0,100 -i input.dcm -o output_black.dcm
```

Warning: if the Pixel Data is compressed, the image is first decompressed so that pixel can be set to 0, but it is not re-compressed.

9.6.2 Convert RAW to DICOM

Recognized extension is .raw, .rawl, .gray or .rgb (case insensitive)

```
$ gdcmimg --size 512,512 --depth 16 -i input.raw -o output.dcm
```

the image will be a Secondary Capture.

When the input is 3 component, one need to specify explicitly the Samples Per Pixel:

```
$ gdcmimg --size 512,512 --spp 3  input_rgb.raw output_rgb.dcm
```

When the filename contains .rgb as file extension output is automatically recognized as RGB no need to specify –spp

```
$ gdcmimg --size 512,512 input.rgb output_rgb.dcm
```

You can use the **dd** cmd line to skip any header you would like to discard, for instance, if you would like to skip the first 108 bytes, simply do:

```
$ dd skip=108 bs=1 if=input.raw of=output.raw
```

.raw and .rawl extension are equivalent. You need to explicitly specify the endianess manually:

```
$ gdcmimg --endian MSB --size 512,512 --depth 16 -i input.raw -o output.dcm
```

or

```
$ gdcmimg --endian LSB --size 512,512 --depth 16 -i input.raw -o output.dcm
```

9.6.3 Convert PGM/PNM/PPM to DICOM

Recognized extensions are .pgm, .pnm, .ppm (case insensitive)

```
$ gdcmimg -i input.pgm -o output.dcm
```

the image will be a Secondary Capture

9.6.4 Convert RLE to DICOM

Recognized extension is .rle (case insensitive)

```
$ gdcmimg --size 512,512 --depth 16 -i input.rle -o output.dcm
```

the image will be a Secondary Capture

9.6.5 Convert JPEG to DICOM

Recognized extensions are .jpg, .jpeg, .ljpg, .ljpeg (case insensitive)

```
$ gdcmimg -i input.ljpeg -o output.dcm
```

the image will be a Secondary Capture

9.6.6 Convert J2K to DICOM

Recognized extensions are .j2k, .jp2, .jpc, jpx, j2c (case insensitive)

```
$ gdcmimg -i input.j2k -o output.dcm
```

the image will be a Secondary Capture.

All Pixel informations (Bits Stored/Allocated...) will be derived from the image itself, and not from the command line options.

9.6.7 Specifying a SOP Class UID

Instead of the default Secondary Capture Image Storage, one may want to specify, say VL Photographic Image Storage.

```
$ gdcmimg --sop-class-uid 1.2.840.10008.5.1.4.1.1.77.1.4 input.jpg output.dcm
```

9.7 Multiple Files

gdcmimg handle nicely a set of files (for instance jpeg):

```
$ gdcmimg 1.jpg 2.jpg 3.jpg 4.jpg output.dcm
```

9.8 Start Offset

In some case, one may want to create a 2D slice from an arbitrary volume (e.g 3D). In which case –offset becomes handy:

```
$ gdcmimg --offset 4954104330 --size 1673,1673 Input3D_1673_1673_1775.raw slice_1770.dcm
```

9.9 Warning

There are a couple of issues with gdcmimg implementation:

For RAW file, one should pay attention that when using –endian MSB the Pixel Data will be encapsulated as is (not touched by gdcmimg). Therefore the only possible transfer syntax available is Implicit VR Big Endian DLX (G.E Private). GDCM does handle this private Transfer Syntax. So if you need to convert this Transfer Syntax to another one (and allow Pixel Data manipulation), you can use:

```
$ gdcmconv --raw --force input_big_endian_dlx.raw -o output_implicit_vr_little_endian.dcm
```

For JFIF file and JP2 file (with header) the header is copied into the Pixel Data element which is illegal for JP2. Use gdcmconv to properly re-encode a JP2/JFIF file into J2K/JPG.

```
$ gdcmimg input.jp2 output_jp2.dcm
$ gdcmconv --j2k --force output_jp2.dcm output_j2k.dcm
```

For RLE file, no check is done for crossing the row boundary. It is recommended to use gdcmconv –rle to re-encode into a proper RLE file in case of doubt.

Of course if the compression is not ok with your setup, you can always de-encapsulated the DICOM file (typically JPEG) to a non-encapsulated form, using gdcmconv:

```
$ gdcmconv --raw input_jpeg.dcm output_raw.dcm
```

9.10 SEE ALSO

gdcmdump(1), **gdcm2vtk**(1), **gdcmraw**(1), **convert**(1), **dd**(1)

9.11 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 10

Display meta info about the input DICOM file.

10.1 SYNOPSIS

```
gdcminfo [options] file-in
```

10.2 DESCRIPTION

The **gdcminfo** command line program takes as input a DICOM file, or a directory and process it to extract meta-information about the DICOM file processed.

10.3 PARAMETERS

```
file-in   DICOM input filename
```

10.4 OPTIONS

10.4.1 OPTIONS

```
  -r --recursive          recursive.
  -d --check-deflated     check if file is proper deflated syntax.
     --resources-path     Resources path.
     --md5sum             Compute md5sum of Pixel Data attribute value.
     --check-compression  check the encapsulated stream compression (lossless/lossy).
```

10.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).
```

```
  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

10.4.3 environment variable

GDCM_RESOURCES_PATH path pointing to resources files (Part3.xml, ...)

10.5 Simple usage

10.5.1 gdcmData

Using data from gdcmData:

```
$ gdcminfo gdcmData/012345.002.050.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
NumberOfDimensions: 2
Dimensions: (256,256)
Origin: (-85,21.6,108.7)
Spacing: (0.664062,0.664062,1.5)
DirectionCosines: (1,0,0,0,0,-1)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel    :1
BitsAllocated      :16
BitsStored         :16
HighBit            :15
PixelRepresentation:1
Orientation Label: CORONAL
```

10.5.2 Davie Clunie datasets:

Using data from David Clunie datasets:

```
$ gdcminfo BRTUM001.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4.1 [Enhanced MR Image Storage]
NumberOfDimensions: 3
Dimensions: (256,256,15)
Origin: (40,-105,105)
Spacing: (0.820312,0.820312,6)
DirectionCosines: (0,1,0,0,0,-1)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel    :1
BitsAllocated      :16
BitsStored         :16
HighBit            :15
PixelRepresentation:1
Orientation Label: SAGITTAL
```

10.5.3 Checking the md5sum of the Pixel Data

After compressing a DICOM file (see gdcmconv) using a lossless compression algorithm, it is fairly easy to compare the two files for differences at DICOM attribute level. However one operation is slightly easier to do: how to make sure the compression was actually lossless ? In this case one could use the –md5sum operation.

Take an uncompressed DICOM image file:

```
$ gdcminfo --md5sum SIEMENS_ImageLocationUN.dcm
```

The tool return: 0621954acd5815e0b4f7b65fcc6506b1

Now compress this file:

```
$ gdcmconv --jpegls SIEMENS_ImageLocationUN.dcm lossless_compressed.dcm
```

and then check again the md5sum:

```
$ gdcminfo --md5sum lossless_compressed.dcm
```

The tool return: 0621954acd5815e0b4f7b65fcc6506b1

10.5.4 Checking if Pixel Data is lossless

In some environment one wish to check whether or not the DICOM file is lossless. It is fairly easy to do that in most cases. Only in two occasion this is not clear from the sole DICOM Attribute. When the Transfer Syntax is JPEG 2000 Image Compression (1.2.840.10008.1.2.4.91) and when the Transfer Syntax is JPEG-LS Lossy (Near-Lossless) Image Compression (1.2.840.10008.1.2.4.81).

In this case, the only solution is to open the Pixel Data element, read the specific JPEG header and check whether or not the JPEG transformation was lossless or not:

```
$ gdcminfo --check-compression gdcmData/MAROTECH_CT_JP2Lossy.dcm
```

The tool returns: "Encapsulated Stream was found to be: lossy"

10.6 SEE ALSO

gdcmdump(1), **gdcmraw**(1), **gdcmconv**(1)

10.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 11

Tool to convert PAPYRUS 3.0 to DICOM.

11.1 SYNOPSIS

```
gdcmconv [options] file-in file-out
```

11.2 DESCRIPTION

The **gdcmconv** command line program takes as input a PAPYRUS 3.0 file (file-in) and process it to generate an output (pseudo) DICOM file (file-out). The command line option dictate the type of operation(s) gdcmconv will use to generate the output file.

11.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

11.4 OPTIONS

11.4.1 PARAMETERS

```
  -i --input      DICOM filename
  -o --output     DICOM filename
```

11.4.2 OPTIONS

```
  -S --split            Split multiframes PAPYRUS 3.0 into multiples DICOM files
     --decomp-pap3      Use PAPYRUS 3.0 for decompressing (can be combined with --split).
     --check-iop        Check that the Image Orientation (Patient) Attribute is ok (see --split).
```

11.4.3 general options

```
  -h   --help
```

```
        print this help text and exit

  -v   --version
        print version information and exit

  -V   --verbose
        verbose mode (warning+error).

  -W   --warning
        warning mode, print warning information

  -E   --error
        error mode, print error information

  -D   --debug
        debug mode, print debug information
```

11.4.4 environment variable

```
  GDCM_ROOT_UID Root UID
```

11.5 Simple usage

gdcmconv is a great tool to convert broken PAPYRUS 3.0 implementation into properly parsable DICOM file. Usage is simply:

```
$ gdcmconv input.pa3 output.dcm
```

or if you prefer being explicit:

```
$ gdcmconv -i input.pa3 -o output.dcm
```

Even though gdcmpap3 can overwrite directly on the same file (input.pa3 = output.dcm), it is recommended that user should first convert into a different file to make sure the bug is properly handled by GDCM.

11.6 SEE ALSO

gdcmdump(1), **gdcmconv**(1), **gdcminfo**(1)

11.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 12

Tool to convert PDF to PDF/DICOM.

12.1 SYNOPSIS

```
gdcmpdf [options] file-in file-out
```

12.2 DESCRIPTION

The **gdcmpdf** tool convert a PDF file (any PDF version) into an encapsulated PDF/DICOM file. By default it will try to read the PDF meta information stored in the PDF and convert this information to some specific DICOM fields (see below). However it may fails (eg. wrong password on encrypted PDF file) in which case empty value are used.

12.3 PARAMETERS

```
file-in   PDF input filename

file-out  DICOM output filename
```

12.4 OPTIONS

12.4.1 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
```

```
         debug mode, print debug information
```

12.5 Usage Example

```
$ wget http://gdcm.sourceforge.net/gdcm.pdf
$ gdcmpdf gdcm.pdf gdcm.dcm
```

To re-extract the encapsulated PDF file:

```
 $ gdcmraw -i gdcm.dcm -t 42,11 -o gdcm.dcm.pdf
 $ diff gdcm.pdf gdcm.dcm.pdf
```

12.6 PDF Info Mapping

Here is how the PDF info is mapped to DICOM information (typical pdfinfo output):

```
Title:          GDCM Reference Manual
Subject:        Grassroots DICOM API reference
Keywords:       GDCM,DICOM,JPEG,Lossless JPEG,JPEG-LS,J2K,JPEG 2000,RLE
Author:         Mathieu Malaterre and co.
Creator:        LaTeX with hyperref package
Producer:       pdfeTeX-1.21a
CreationDate:   Tue Apr 28 15:34:26 2009
Tagged:         no
Pages:          1188
Encrypted:      no
Page size:      612 x 792 pts (letter)
File size:      13756841 bytes
Optimized:      yes
PDF version:    1.4
```

Converted to DICOM this leads to:

```
# Dicom-Data-Set
# Used TransferSyntax: Little Endian Explicit
(0008,0005) CS [ISO_IR 100]                              #  10, 1 SpecificCharacterSet
(0008,0012) DA [20090428]                                #   8, 1 InstanceCreationDate
(0008,0013) TM [182550.302631]                           #  14, 1 InstanceCreationTime
(0008,0016) UI =EncapsulatedPDFStorage                   #  30, 1 SOPClassUID
(0008,0018) UI [1.2.826.0.1.3680043.2.1143.776842935192792959289022034349197114] #  64, 1 SOPInstanceUID
(0008,0020) DA [20090428]                                #   8, 1 StudyDate
(0008,0023) DA [20090428]                                #   8, 1 ContentDate
(0008,002a) DT [20090428153437.000000]                   #  22, 1 AcquisitionDateTime
(0008,0030) TM [182550.302160]                           #  14, 1 StudyTime
(0008,0033) TM [153426.000000]                           #  14, 1 ContentTime
(0008,0050) SH (no value available)                      #   0, 0 AccessionNumber
(0008,0060) CS [OT]                                      #   2, 1 Modality
(0008,0064) CS [WSD]                                     #   4, 1 ConversionType
(0008,0070) LO [LaTeX with hyperref package]             #  28, 1 Manufacturer
(0008,0090) PN (no value available)                      #   0, 0 ReferringPhysiciansName
(0010,0010) PN [Mathieu Malaterre and co.]               #  26, 1 PatientsName
(0010,0020) LO (no value available)                      #   0, 0 PatientID
(0010,0030) DA (no value available)                      #   0, 0 PatientsBirthDate
(0010,0040) CS (no value available)                      #   0, 0 PatientsSex
(0018,1020) LO [pdfeTeX-1.21a]                           #  14, 1 SoftwareVersions
(0020,000d) UI [1.2.826.0.1.3680043.2.1143.186812183222341735165423248075512313 3] #  64, 1 StudyInstanceUID
(0020,000e) UI [1.2.826.0.1.3680043.2.1143.133009915082574661750784610766396431 1] #  64, 1 SeriesInstanceUID
(0020,0010) SH (no value available)                      #   0, 0 StudyID
(0020,0011) IS [1]                                       #   2, 1 SeriesNumber
```

```
(0020,0013) IS [1]                                      #   2, 1 InstanceNumber
(0028,0301) CS [YES]                                    #   4, 1 BurnedInAnnotation
(0040,a043) SQ (Sequence with explicit length #=0)      #   0, 1 ConceptNameCodeSequence
(fffe,e0dd) na (SequenceDelimitationItem for re-encod.) #   0, 0 SequenceDelimitationItem
(0042,0010) ST [GDCM Reference Manual]                  #  22, 1 DocumentTitle
(0042,0011) OB 25\\50\\44\\46\\2d\\31\\2e\\34\\0a\\25\\e7\\f3\\cf\\d3\\0a\\33\\32\\30\\37\\37\\20\\30... # 137568
(0042,0012) LO [application/pdf]                        #  16, 1 MIMETypeOfEncapsulatedDocument
```

```
$ stat gdcm.pdf
  File: 'gdcm.pdf'
  Size: 13756841        Blocks: 26912      IO Block: 4096   regular file
Device: fe01h/65025d    Inode: 2675750     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1002/mmalaterre)   Gid: ( 1002/mmalaterre)
Access: 2009-04-28 16:05:00.000000000 +0200
Modify: 2009-04-28 15:34:37.000000000 +0200
Change: 2009-04-28 16:05:00.000000000 +0200
```

Explanation for the different Date/Time mappings:

- Study Date/Time, Instance Creation Date/Time are both equal to the current time gdcmpdf tool was run,
- Acquisition Date Time is set to the Modify Time of the actual PDF file,
- Content Date/Time are set from the actual PDF header info: CreationDate.

12.7 SEE ALSO

gdcmconv(1), **gdcmraw**(1), **pdfinfo**(1)

12.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 13

Extract Data Element Value Field.

13.1 SYNOPSIS

```
gdcmraw [options] file-in file-out
```

13.2 DESCRIPTION

The **gdcmraw** tool is mostly used for development purpose. It is used to extract a specific binary field from a DICOM DataSet.

13.3 PARAMETERS

```
file-in   DICOM input filename

file-out  output filename
```

13.4 OPTIONS

13.4.1 PARAMETERS

```
  -i --input     Input filename
  -o --output    Output filename
  -t --tag       Specify tag to extract value from.
```

13.4.2 OPTIONS

```
  -S --split-frags  Split fragments into multiple files.
  -p --pattern      Specify trailing file pattern (see split-frags).
  -P --pixel-data   Pixel Data trailing 0.
```

13.4.3 general options

```
  -h   --help
```

```
        print this help text and exit

  -v   --version
        print version information and exit

  -V   --verbose
        verbose mode (warning+error).

  -W   --warning
        warning mode, print warning information

  -E   --error
        error mode, print error information

  -D   --debug
        debug mode, print debug information
```

13.5 Typical usage

13.5.1 Copy Attribute Value to file

This will extract the value at Tag (0025,101b):

```
$ gdcmraw -i GE_MR_0025xx1bProtocolDataBlock.dcm -t 25,101b -o pdb.raw
```

13.5.2 Extract Pixel Data

If you do not specify any tag, the Pixel Data element is the default one. So for instance to grab the Pixel Data from an image:

```
$ gdcmraw -i test.acr -o test.raw
```

You can then for example compute the md5sum of this pixel data (very useful):

```
$ md5sum test.raw
f845c8f283d39a0204c325654493ba53  test.raw
```

13.5.3 Encapsulated Syntax

When the Pixel Data is encapsulated, multiple fragments can be used to store a single slice image:

```
$ gdcmdump D_CLUNIE_CT1_J2KR.dcm
```

```
\&...
(7fe0,0010) OB                                          # u/l,1 Pixel Data
  (fffe,e000) ?? []                           # 0,1 Item
  (fffe,e000) ?? ff\\4f\\ff\\51\\00\\29\\00\\00\\00\\00\\02\\00\\00\\00\\02\\00\\00\\00\\00\\00\\00\\00\\
  (fffe,e000) ?? 2c\\b7\\ee\\68\\de\\e3\\93\\2d\\b3\\b8\\ba\\90\\7b\\42\\3e\\f8\\42\\16\\64\\88\\46\\30\\37\\d4\\
  (fffe,e000) ?? 48\\3c\\03\\e8\\c4\\3f\\44\\e1\\8a\\5c\\73\\3b\\02\\0a\\ad\\a5\\8f\\e4\\0c\\81\\76\\a2\\d7\\1b\\
(fffe,e0dd) 0
```

In order to create a J2K image out of it, we need to extract each fragments and concatenate them:

```
$ gdcmraw -i D_CLUNIE_CT1_J2KR.dcm -o D_CLUNIE_CT1_J2KR.j2k
```

This is a valid J2K file, using the Kakadu software package:

```
$ kdu_expand -i D_CLUNIE_CT1_J2KR.j2k -o D_CLUNIE_CT1_J2KR.tiff -record D_CLUNIE_CT1_J2KR.txt
```

```
$ cat D_CLUNIE_CT1_J2KR.txt
```

```
Sprofile=PROFILE2
Scap=no
Sextensions=0
Ssize={512,512}
Sorigin={0,0}
Stiles={512,512}
Stile_origin={0,0}
Scomponents=1
Ssigned=yes
Sprecision=16
Ssampling={1,1}
Sdims={512,512}
Cycc=no
Cmct=0
Clayers=1
Cuse_sop=no
Cuse_eph=no
Corder=LRCP
Calign_blk_last={no,no}
Clevels=5
Cads=0
Cdfs=0
Cdecomp=B(-:-:-)
Creversible=yes
Ckernels=W5X3
Catk=0
Cuse_precincts=no
Cblk={64,64}
Cmodes=0
Qguard=1
Qabs_ranges=18,19,19,20,19,19,20,19,19,20,19,19,20,19,19,20

>> New attributes for tile 0:
```

13.5.4 Extract fragments as single file

Sometimes each fragments is in fact a single slice, so we would not need to concatenate them:

```
$ gdcmdump 00191113.dm
```

```
\&...
(7fe0,0010) OB                                                  # u/l,1 Pixel Data
  (fffe,e000) ?? 00\\00\\00\\00\\6b\\38\\01\\00\\10\\77\\02\\00\\37\\b6\\03\\00\\a7\\f4\\04\\00         # 20,1 It
  (fffe,e000) ?? ff\\d8\\ff\\c3\\00\\0b\\08\\02\\00\\02\\00\\01\\00\\11\\00\\ff\\c4\\00\\1b\\00\\01\\01\\01\\01\\
  (fffe,e000) ?? ff\\d8\\ff\\c3\\00\\0b\\08\\02\\00\\02\\00\\01\\00\\11\\00\\ff\\c4\\00\\1b\\00\\01\\01\\01\\01\\
  (fffe,e000) ?? ff\\d8\\ff\\c3\\00\\0b\\08\\02\\00\\02\\00\\01\\00\\11\\00\\ff\\c4\\00\\1b\\00\\01\\01\\01\\01\\
  (fffe,e000) ?? ff\\d8\\ff\\c3\\00\\0b\\08\\02\\00\\02\\00\\01\\00\\11\\00\\ff\\c4\\00\\1b\\00\\01\\01\\01\\01\\
(fffe,e0dd) 0
```

Let's try to extract those 4 individual Lossless jpeg individually:

```
$ gdcmraw --split-frags -i 00191113.dcm -o jpeg --pattern %02d.ljpeg
```

This will output 4 files:

```
-rw-r--r--  1 mathieu mathieu   81512 2008-08-08 22:10 jpeg03.ljpeg
-rw-r--r--  1 mathieu mathieu   81694 2008-08-08 22:10 jpeg02.ljpeg
-rw-r--r--  1 mathieu mathieu   81564 2008-08-08 22:10 jpeg01.ljpeg
-rw-r--r--  1 mathieu mathieu   79970 2008-08-08 22:10 jpeg00.ljpeg
```

13.6 Footnote about JPEG files

It is a common misunderstanding to interchange 'JPEG 8bits lossy' with simply JPEG file. The JPEG specification is much broader than simply the common lossy 8bits file (as found on internet).

You can have:

- JPEG Lossy 8bits
- JPEG Lossy 12bits
- JPEG Lossless 2-16bits

Those are what is defined in ITU-T T.81, ISO/IEC IS 10918-1.

13.7 SEE ALSO

gdcmdump(1), **gdcmraw**(1)

13.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 14

Scan a directory containing DICOM files.

14.1 SYNOPSIS

```
gdcmscanner [options] directory
```

14.2 DESCRIPTION

The **gdcmscanner** is a command line tool to quickly extract value from a set of DICOM attribute in a DICOM File-Set.

14.2.1 PARAMETERS

```
  -d --dir       DICOM directory
  -t --tag %d,%d DICOM tag(s) to look for
```

14.2.2 OPTIONS

```
  -p --print      Print output.
  -r --recursive  Recusively descend directory.
```

14.2.3 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).

  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

14.3 Typical usage

14.4 Simple usage

In order to display all the value for Patient Name (0010,0010) in the directory name **gdcmData**, simply do:

```
$ gdcmscanner -t 10,10 -d gdcmData -p
```

14.5 Complex usage

Because gdcmscanner does not support progress, you have to wait until all files are traversed to see any results. This is quite cumbersome, on UNIX this can be worked around with the following trick:

```
$ find gdcmData -type d -exec gdcmscanner -t 10,10 -d {} -p ';'
```

So all directory are locally traversed (no child directory are recursively traversed), which means results comes out much faster.

14.6 SEE ALSO

gdcmdump(1), **gdcmraw**(1)

14.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 15

Tool to execute a DICOM Query/Retrieve operation

15.1 SYNOPSIS

```
gdcmscu [OPTION]...[OPERATION]...HOSTNAME...[PORT]...
```

Execute a DICOM Q/R operation to HOSTNAME, using port PORT (104 when not specified)

15.2 DESCRIPTION

The **gdcmscu** command line program is the tool to execute DICOM Query/Retrieve operation. It supports:

- C-ECHO (SCU)
- C-FIND (SCU)
- C-STORE (SCU)
- C-MOVE (SCU/SCP) C-MOVE operation are executed using two different ports (one for the SCU and one for the SCP).

15.3 PARAMETERS

15.4 OPTIONS

15.4.1 OPTIONS

```
  -H --hostname    %s  Hostname.
  -p --port        %d  Port number.
     --aetitle     %s  Set calling AE Title.
     --call        %s  Set called AE Title.
```

15.4.2 mode options

```
     --echo           C-ECHO (default when none).
     --store          C-STORE.
```

```
    --find           C-FIND.
    --move           C-MOVE.
```

15.4.3 C-STORE options

```
-i --input       %s   DICOM filename
-r --recursive        recursively process (sub-)directories
   --store-query %s   Store constructed query in file
```

15.4.4 C-FIND/C-MOVE options

```
    --patientroot       C-FIND Patient Root Model.
    --studyroot         C-FIND Study Root Model.

    --patient           C-FIND Query on Patient Info (cannot be used with --studyroot).
    --study             C-FIND Query on Study Info.
    --series            C-FIND Query on Series Info.
    --image             C-FIND Query on Image Info.
    --key   %d,%d[=%s]  0123,4567=VALUE for specifying search criteria (wildcard allowed)
                        With --key, leave blank (ie, --key 10,20="" or --key 10,20) to retrieve values
```

15.4.5 C-MOVE options

```
-o --output    %s      DICOM filename / directory
   --port-scp  %d      Port for incoming associations
   --key   %d,%d[=%s   0123,4567=VALUE for specifying search criteria (wildcard not allowed)
                       Note that C-MOVE supports the same queries as C-FIND, but no wildcards are allowed
```

15.4.6 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information

-L   --log-file
       specify a filename where to write logs

--queryhelp
       print query help
```

15.4.7 environment variable

```
GDCM_ROOT_UID Root UID
```

15.5 C-ECHO usage

gdcmscu is a great tool to test if a DICOM server is up. For example to send a C-ECHO to server dicom.example.com using port 104, use:

```
$ gdcmscu dicom.example.com
```

or if you prefer being explicit:

```
$ gdcmscu --echo dicom.example.com 104
```

Using basic security your DICOM server might require that you set the appropriate called AE-TITLE

```
$ gdcmscu --echo dicom.example.com 11112 --call SERVSCP
```

If you want to specifiy your own AE-TITLE (default is GDCMSCU), simply use:

```
$ gdcmscu --echo dicom.example.com 11112 --call SERVSCP --aetitle MYSCU
```

For example you could test on the DICOM server provided by DICOMObject team:

```
$ gdcmscu www.dicomserver.co.uk 11112
```

15.6 C-STORE usage

C-STORE is the operation that allow sending a DICOM file to a remote DICOM server. For instance to send a file called myfile.dcm

```
$ gdcmscu --store dicom.example.com 104 myfile.dcm
```

or if you prefer being explicit:

```
$ gdcmscu --store dicom.example.com 104 -i myfile.dcm
```

You can even send multiple files using the same association:

```
$ gdcmscu --store dicom.example.com 104 myfile1.dcm myfile2.dcm myfile3.dcm ...
```

15.7 C-FIND usage

gdcmscu also allow querying a DICOM server. This is the C-FIND operation, for example to find all DICOM Instance where PatientsName match a particular pattern, usage is simply:

```
$ gdcmscu --find --patient dicom.example.com 11112 --patientroot --key 10,10,"A*"
```

We also support a DCMTK compatible convention:

```
$ gdcmscu --find --patient dicom.example.com 11112 --patientroot --key 10,10="A*"
```

When an attribute is set without a value it will be part of the output result:

```
$ gdcmscu --find --patient dicom.example.com 11112 --call MI2B2 --patientroot -k 10,10="A*" -k 10,20
```

15.8 C-MOVE usage

C-MOVE is the operation to retrieve a DICOM instance from a remote DICOM server. Most of the time, it is a subsequent operation after a C-FIND query. To retrieve a DICOM instance where PatientID is ABCD1234, simply execute:

```
$ gdcmscu --move --patient --aetitle ACME1 --call ACME_STORE dicom.example.com 5678 --patientroot -k 10,20="ABCD1
```

WARNING For this operation to work you need information from the DICOM server you are communicating with. Only the DICOM server you are sending a C-MOVE query will be responsible for sending back incoming associations (the actual C-STORE SCP). Therefore you need to make sure that you mapping of (AE-TITLE,PortNumber) is properly set on the DICOM server side as well as the port for incoming association (–port-scp).

gdcmscu does not currently support external C-STORE association (C-STORE request sent to an external SCP application).

15.9 patientroot notes

The flag –patientroot is just simply a wrapper around the syntax –key 8,52=PATIENT For instance one would write using DCMTK syntax:

```
$ findscu --patient dicom.example.com 11112 --key 8,52=PATIENT --key 10,10="F*"
```

This would become using GDCM syntax:

```
$ gdcmscu --find --patient dicom.example.com 11112 --patientroot --key 10,10="F*"
```

15.10 Debugging

This is sometime difficult to investigate why a connection to a remote DICOM server cannot be done. Some recommendations follows:

Always try to do a simple C-ECHO at first. If you cannot get the C-ECHO to work none of the other operations will work

Before trying to a C-MOVE operation, make sure you can execute the C-FIND equivalent query first.

When doing a C-MOVE operation you really need to communicate with the PACS admin as the C-MOVE operation is different from the other lower level operation such as HTTP/GET. When doing a C-MOVE, the server will communicate back using another channel (could be different port) using it's internal database to map an AE-TITLE back to the destination IP.

Indeed the C-MOVE operation by design does not always use your incoming IP address to send back the resulting dataset. Instead it uses a mapping of AE-TITLE to IP address to send back any results. So pay particular attention to the spelling of your AE-TITLE and your incoming port (which may be different from the port to connect to the server).

15.11 Port Warning

Watch out that port ranging [1-1024] are reserved for admin and not easily accessible unless granted special privileges. Therefore the default 104 DICOM port might not be accessible to all your users.

15.12 C-STORE Warnings

When constructing a C-STORE operation, gdcmscu will always use the Media Storage SOP Class UID as found in the file to be sent. For encapsulated DICOM file (eg. RLE Lossless) the receiving SCP server might not support this compression and will legitimately refuse the C-STORE operation. In this case users have to manually convert to a non-compressed form this particular file:

```
$ gdcmconv --raw compressed.dcm non_compressed.dcm
```

15.13 C-MOVE Warnings

At the moment **gdcmscu** only supports non-compressed transfer syntax. It will always request DataSet using Implicit VR Little Endian Transfer Syntax during a C-MOVE operation (both incoming and outgoing associations). This make gdcmscu –move equivalent to DCMTK movescu syntax:

```
$ movescu -xi +xi ...
```

15.14 C-FIND IMAGE level (Composite Object Instance)

One should pay attention that gdcmscu –find and findscu are not completely equivalent. Using gdcmscu –find, all Unique Keys will be added automatically. One can therefore execute something like this:

```
$ gdcmscu --find --patientroot --image --key 8,18=1.2.3.4.5.6 dicom.example.com 11112
```

instead of the more explicit form

```
$ gdcmscu --find --patientroot --image --key 8,18=1.2.3.4.5.6 dicom.example.com 11112 --key 10,20 --key 20,d --ke
```

This would also be equivalent to:

```
$ findscu --patient --key 8,52=IMAGE --key 8,18=1.2.3.4.5.6 dicom.example.com 11112 --key 10,20 --key 20,d --key
```

15.15 Storing the Query

It is also possible to store the query:

```
gdcmscu --find --patient --patientroot dicom.example.com 11112 --key 10,20="*" --key 10,10 --store-query query.dc
```

One can then check the DataSet values send for the query:

```
$ gdcmdump query.dcm
# Dicom-File-Format

# Dicom-Meta-Information-Header
# Used TransferSyntax:

# Dicom-Data-Set
# Used TransferSyntax: 1.2.840.10008.1.2
(0008,0005) ?? (CS) [ISO_IR 192]                                    # 10,1-n Specific Character Set
(0008,0052) ?? (CS) [PATIENT ]                                      # 8,1 Query/Retrieve Level
(0010,0010) ?? (PN) (no value)                                      # 0,1 Patient's Name
(0010,0020) ?? (LO) [* ]                                            # 2,1 Patient ID
```

The Specific Character Set was set to "ISO_IR 192" as the locale encoding of the system was found automatically by gdcmscu to be UTF-8.

This means that the following command line will properly setup the Query with the appropriate Charset to be executed correctly:

```
$ gdcmscu --find --patient --patientroot dicom.example.com 11112 --key 10,10="*Jérôme*"
```

The query is always executed on the server side (SCP), some implementations does not support string matching with different Character Set.

15.16 DICOM Public Servers

An up to date list of DICOM Public Servers can be found at:

`http://www.dclunie.com/medical-image-faq/html/part8.html#DICOMPublicServers`

15.17 SEE ALSO

gdcmconv(1)

15.18 COPYRIGHT

Copyright Insight Software Consortium

Chapter 16

Concatenate/Extract DICOM files.

16.1 SYNOPSIS

```
gdcmtar [options] file-in file-out
```

16.2 DESCRIPTION

The **gdcmtar** is a command line tool used to tar/untar multi-frames images (including SIEMENS MOSAIC file)

16.3 PARAMETERS

```
file-in   DICOM input filename

file-out  DICOM output filename
```

16.4 OPTIONS

16.4.1 OPTIONS

```
     --enhance    enhance (default)
  -U --unenhance  unenhance
  -M --mosaic     Split SIEMENS Mosaic image into multiple frames.
  -p --pattern    Specify trailing file pattern.
     --root-uid        Root UID.
```

16.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).
```

```
  -W   --warning
         warning mode, print warning information

  -E   --error
         error mode, print error information

  -D   --debug
         debug mode, print debug information
```

16.4.3 environment variable

```
  GDCM_ROOT_UID Root UID
```

16.5 Typical usage

16.5.1 SIEMENS Mosaic

```
$ gdcminfo MR-sonata-3D-as-Tile.dcm

MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
TransferSyntax is 1.2.840.10008.1.2.1 [Explicit VR Little Endian]
NumberOfDimensions: 2
Dimensions: (384,384,1)
\&...

$ gdcmtar --mosaic -i MR-sonata-3D-as-Tile.dcm -o mosaic --pattern %03d.dcm
```

Will output:

```
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic000.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic001.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic002.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic003.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic004.dcm
-rw-r--r--   1 mathieu mathieu 72886 2009-08-10 11:14 mosaic005.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic006.dcm
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic007.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic008.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic009.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic010.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic011.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic012.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic013.dcm
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic014.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic015.dcm
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic016.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic017.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic018.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic019.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic020.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic021.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic022.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic023.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic024.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic025.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic026.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic027.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic028.dcm
-rw-r--r--   1 mathieu mathieu 72884 2009-08-10 11:14 mosaic029.dcm
-rw-r--r--   1 mathieu mathieu 72882 2009-08-10 11:14 mosaic030.dcm
```

```
$ gdcminfo mosaic000.dcm


MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
TransferSyntax is 1.2.840.10008.1.2.1 [Explicit VR Little Endian]
NumberOfDimensions: 2
Dimensions: (64,64,1)
\&...
```

16.6 SEE ALSO

gdcmdump(1), **gdcmraw**(1), **gdcminfo**(1)

16.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 17

Simple DICOM viewer.

17.1 SYNOPSIS

```
gdcmviewer [options] file-in
```

17.2 DESCRIPTION

The **gdcmviewer** is a simple tool that show how to use vtkGDCMImageReader. The class that use gdcm to make a layer to VTK. **gdcmviewer** is basically only just a wrapper around VTK/GDCM.

This tool is meant for testing integration of GDCM in VTK. You should see it as a demo tool. It does compile with VTK ranging from 4.2 to 5.10, but only with VTK 5.2 (or above) can play with the widgets (as described below).

17.3 PARAMETERS

```
file-in   DICOM input filename
```

17.4 OPTIONS

17.4.1 OPTIONS

```
     --force-rescale    force rescale (advanced users)
     --force-spacing    force spacing (advanced users)
  -r --recursive        Recusively descend directory
```

17.4.2 general options

```
  -h   --help
         print this help text and exit

  -v   --version
         print version information and exit

  -V   --verbose
         verbose mode (warning+error).
```

```
-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

17.5 Typical usage

17.6 Simple usage

For now gdcmviewer should be started from a command line prompt. The next argument should be the name of the DICOM file you wish to read. For instance:

```
$ gdcmviewer -V 012345.002.050.dcm
```

gdcmviewer will try to read your file, and then print the vtk information associated with this file. Basically what kind of image you are looking at.

- ScalarType is the DICOM Real World Value type
- Dimensions is the dimension of the image
- Spacing is the spacing of the image
- NumberOfScalarComponents should be 1 for grayscale & PALETTE COLOR and 3 for RGB, YBR data.

17.7 Wiki Link

The wiki page, with color pictures can be found at: `http://gdcm.sourceforge.net/wiki/index.php/Gdcmviewer`

17.8 SEE ALSO

gdcmdump(1), **gdcm2vtk**(1)

17.9 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 18

provides a tool to convert a DICOM file into a XML infoset and vice-versa.

18.1 SYNOPSIS

```
gdcmxml [options] file-in[DICOM or XML] file-out[XML or DICOM]
```

18.2 DESCRIPTION

The **gdcmxml** command line program converts a DICOM file (DataSet) into an XML file (according to the Native DICOM Model) or vice-versa. For those familiar with DCMTK, this provides binary capabilities (i.e. functionality of both dcm2xml and xml2dcm).

The XML infoset which is from the DICOM file gdcmXMLPrintet Class. This is in strict compliance with the Native DICOM Model as given in Supp 118.

18.3 PARAMETERS

```
file-in   DICOM or XML input filename ( cannot be absent)

file-out  output filename (can be absent)
```

18.4 OPTIONS

18.4.1 PARAMETERS

```
  -i --input       DICOM filename
  -o --output      DICOM filename
```

18.4.2 Options for DICOM to XML:

```
  -B --loadBulkData   Loads bulk data into a binary file named "UUID" (by default UUID are written).
```

18.4.3 Options for XML to DICOM:

```
-B --loadBulkData   Loads bulk data from a binary file named as the "UUID" in XML file (by default UUID are wri
-T --TransferSyntax Loads transfer syntax from file (default is LittleEndianImplicit)
```

18.4.4 general options

```
-h   --help
       print this help text and exit

-v   --version
       print version information and exit

-V   --verbose
       verbose mode (warning+error).

-W   --warning
       warning mode, print warning information

-E   --error
       error mode, print error information

-D   --debug
       debug mode, print debug information
```

18.5 SEE ALSO

gdcmdump(1), **gdcmconv**(1)

18.6 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 19

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 differenciate 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 20

Deprecated List

Member gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, bool inMove=false)

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 21

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

Class gdcm::IPPSorter

There are currently a couple of bugs in this implementation:

- Gantry Tilt is not considered

Chapter 22

Namespace Index

22.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gdcm 109
gdcm::network 131
gdcm::SegmentHelper 136
gdcm::terminal
Class for Terminal Allow one to print in color in a shell 136

Chapter 23

Hierarchical Index

23.1 Class Hierarchy

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

gdcm::network::AbstractSyntax . . . 150
gdcm::network::ApplicationContext . . . 160
gdcm::ApplicationEntity . . . 161
gdcm::network::ARTIMTimer . . . 166
gdcm::ASN1 . . . 167
gdcm::network::AsynchronousOperationsWindowSub . . . 168
gdcm::Attribute< Group, Element, TVR, TVM > . . . 169
gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 176
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 183
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 181
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 188
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 187
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 191
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 190
gdcm::Base64 . . . 195
gdcm::network::BaseCompositeMessage . . . 196
gdcm::network::CEchoRQ . . . 233
gdcm::network::CEchoRSP . . . 234
gdcm::network::CFindCancelRQ . . . 236
gdcm::network::CFindRQ . . . 237
gdcm::network::CFindRSP . . . 238
gdcm::network::CMoveCancelRq . . . 239
gdcm::network::CMoveRQ . . . 241
gdcm::network::CMoveRSP . . . 242
gdcm::network::CStoreRQ . . . 276
gdcm::network::CStoreRSP . . . 277
gdcm::network::BasePDU . . . 198
gdcm::network::AAbortPDU . . . 139
gdcm::network::AAssociateACPDU . . . 141
gdcm::network::AAssociateRJPDU . . . 144
gdcm::network::AAssociateRQPDU . . . 145
gdcm::network::AReleaseRPPDU . . . 163

gdcm::network::AReleaseRQPDU . . . 164
gdcm::network::PDataTFPDU . . . 558
std::basic_string< Char >
std::string
gdcm::String< TDelimiter, TMaxLength, TPadChar > . . . 708
gdcm::SegmentHelper::BasicCodedEntry . . . 204
gdcm::BitmapToBitmapFilter . . . 217
gdcm::PixmapToPixmapFilter . . . 584
gdcm::ImageToImageFilter . . . 449
gdcm::ImageApplyLookupTable . . . 417
gdcm::ImageChangePhotometricInterpretation . . . 419
gdcm::ImageChangePlanarConfiguration . . . 422
gdcm::ImageChangeTransferSyntax . . . 425
gdcm::ImageFragmentSplitter . . . 436
gdcm::ByteBuffer . . . 222
gdcm::ByteSwap< T > . . . 223
gdcm::ByteSwapFilter . . . 224
gdcm::network::CFind . . . 235
gdcm::Coder . . . 244
gdcm::Codec . . . 243
gdcm::AudioCodec . . . 193
gdcm::ImageCodec . . . 429
gdcm::DeltaEncodingCodec . . . 308
gdcm::JPEG2000Codec . . . 476
gdcm::JPEGCodec . . . 482
gdcm::JPEG12Codec . . . 472
gdcm::JPEG16Codec . . . 474
gdcm::JPEG8Codec . . . 480
gdcm::JPEGLSCodec . . . 487
gdcm::KAKADUCodec . . . 492
gdcm::PGXCodec . . . 568
gdcm::PNMCodec . . . 589
gdcm::PVRGCodec . . . 611
gdcm::RAWCodec . . . 625
gdcm::RLECodec . . . 638
gdcm::PDFCodec . . . 564
gdcm::CodeString . . . 246
gdcm::network::CompositeMessageFactory . . . 252
gdcm::CompositeNetworkFunctions . . . 253
gdcm::ConstCharWrapper . . . 257
gdcm::CryptoFactory . . . 259
gdcm::CAPICryptoFactory . . . 229
gdcm::OpenSSLCryptoFactory . . . 538
gdcm::OpenSSLP7CryptoFactory . . . 542
gdcm::CryptographicMessageSyntax . . . 261
gdcm::CAPICryptographicMessageSyntax . . . 230
gdcm::OpenSSLCryptographicMessageSyntax . . . 539
gdcm::OpenSSLP7CryptographicMessageSyntax . . . 543
gdcm::CSAElement . . . 263
gdcm::CSAHeader . . . 267
gdcm::CSAHeaderDict . . . 271
gdcm::CSAHeaderDictEntry . . . 273

gdcm::DataElement . . . 282
gdcm::CP246ExplicitDataElement . . . 257
gdcm::ExplicitDataElement . . . 362
gdcm::ExplicitImplicitDataElement . . . 363
gdcm::Fragment . . . 401
gdcm::BasicOffsetTable . . . 207
gdcm::ImplicitDataElement . . . 456
gdcm::Item . . . 467
gdcm::UNExplicitDataElement . . . 831
gdcm::UNExplicitImplicitDataElement . . . 833
gdcm::VR16ExplicitDataElement . . . 857
gdcm::DataSet . . . 293
gdcm::CommandDataSet . . . 250
gdcm::FileMetaInformation . . . 379
gdcm::DataSetHelper . . . 303
gdcm::Decoder . . . 303
gdcm::Codec . . . 243
gdcm::DefinedTerms . . . 305
gdcm::Defs . . . 305
gdcm::DICOMDIR . . . 309
gdcm::DICOMDIRGenerator . . . 310
gdcm::Dict . . . 312
gdcm::DictConverter . . . 315
gdcm::DictEntry . . . 317
gdcm::Dicts . . . 321
gdcm::network::DIMSE . . . 324
gdcm::DirectionCosines . . . 325
gdcm::Directory . . . 327
gdcm::DirectoryHelper . . . 330
gdcm::DummyValueGenerator . . . 331
gdcm::Element< TVR, TVM > . . . 334
gdcm::Element< TVR, VM::VM1_n > . . . 338
gdcm::Element< TVR, VM::VM1_2 > . . . 337
gdcm::Element< TVR, VM::VM2_n > . . . 342
gdcm::Element< TVR, VM::VM2_2n > . . . 340
gdcm::Element< TVR, VM::VM3_n > . . . 345
gdcm::Element< TVR, VM::VM3_3n > . . . 343
gdcm::Element< VR::AS, VM::VM5 > . . . 346
gdcm::Element< VR::OB, VM::VM1_n > . . . 334
gdcm::Element< VR::OB, VM::VM1 > . . . 347
gdcm::Element< VR::OW, VM::VM1_n > . . . 334
gdcm::Element< VR::OW, VM::VM1 > . . . 348
gdcm::ElementDisableCombinations< TVR, TVM > . . . 350
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 351
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 351
gdcm::EncapsulatedDocument . . . 351
gdcm::EncodingImplementation< T > . . . 352
gdcm::EncodingImplementation< VR::VRASCII > . . . 352
gdcm::EncodingImplementation< VR::VRBINARY > . . . 353
gdcm::EnumeratedValues . . . 355
gdcm::Event . . . 356
gdcm::AnyEvent . . . 158

gdcm::AbortEvent . . . 149
gdcm::AnonymizeEvent . . . 151
gdcm::DataEvent . . . 291
gdcm::DataSetEvent . . . 300
gdcm::EndEvent . . . 354
gdcm::ExitEvent . . . 360
gdcm::FileNameEvent . . . 386
gdcm::InitializeEvent . . . 458
gdcm::IterationEvent . . . 470
gdcm::ModifiedEvent . . . 520
gdcm::ProgressEvent . . . 609
gdcm::StartEvent . . . 698
gdcm::UserEvent . . . 837
gdcm::NoEvent . . . 535
std::exception
gdcm::CSAHeaderDictException . . . 275
gdcm::DataElementException . . . 290
gdcm::Exception . . . 359
gdcm::ParseException . . . 553
gdcm::Fiducials . . . 365
gdcm::FileDerivation . . . 374
gdcm::FileExplicitFilter . . . 377
gdcm::Filename . . . 384
gdcm::FilenameGenerator . . . 388
gdcm::FileSet . . . 391
gdcm::Global . . . 404
gdcm::GroupDict . . . 406
gdcm::IconImageFilter . . . 408
gdcm::IconImageGenerator . . . 410
gdcm::ignore_char . . . 412
gdcm::ImageConverter . . . 435
gdcm::ImageHelper . . . 439
gdcm::network::ImplementationClassUIDSub . . . 454
gdcm::network::ImplementationUIDSub . . . 454
gdcm::network::ImplementationVersionNameSub . . . 455
gdcm::IOD . . . 459
gdcm::IODEntry . . . 460
gdcm::IODs . . . 462
gdcm::JSON . . . 490
gdcm::Scanner::Itstr . . . 500
gdcm::Macro . . . 501
gdcm::Macros . . . 502
gdcm::network::MaximumLengthSub . . . 504
gdcm::MD5 . . . 505
gdcm::MediaStorage . . . 506
gdcm::Module . . . 522
gdcm::ModuleEntry . . . 524
gdcm::NestedModuleEntries . . . 533
gdcm::Modules . . . 526
gdcm::Object . . . 536
gdcm::BaseRootQuery . . . 200
gdcm::FindPatientRootQuery . . . 397
gdcm::FindStudyRootQuery . . . 399

gdcm::MovePatientRootQuery . . . 528
gdcm::MoveStudyRootQuery . . . 530
gdcm::Bitmap . . . 209
gdcm::Pixmap . . . 578
gdcm::Image . . . 413
gdcm::Curve . . . 279
gdcm::File . . . 366
gdcm::FileWithName . . . 396
gdcm::LookupTable . . . 496
gdcm::SegmentedPaletteColorLookupTable . . . 653
gdcm::MeshPrimitive . . . 517
gdcm::Overlay . . . 547
gdcm::Segment . . . 649
gdcm::Subject . . . 715
gdcm::Anonymizer . . . 154
gdcm::Command . . . 248
gdcm::MemberCommand< T > . . . 513
gdcm::SimpleMemberCommand< T > . . . 681
gdcm::FileAnonymizer . . . 369
gdcm::FileChangeTransferSyntax . . . 372
gdcm::FileStreamer . . . 392
gdcm::network::ULConnectionManager . . . 825
gdcm::Scanner . . . 643
gdcm::ServiceClassUser . . . 675
gdcm::Surface . . . 717
gdcm::Value . . . 842
gdcm::ByteValue . . . 224
gdcm::SequenceOfFragments . . . 660
gdcm::SequenceOfItems . . . 665
gdcm::Orientation . . . 545
gdcm::Parser . . . 555
gdcm::Patient . . . 557
gdcm::PDBElement . . . 560
gdcm::PDBHeader . . . 562
gdcm::network::PDUFactory . . . 566
gdcm::PersonName . . . 567
gdcm::PhotometricInterpretation . . . 570
gdcm::PixelFormat . . . 573
gdcm::Preamble . . . 592
gdcm::PresentationContext . . . 593
gdcm::network::PresentationContextAC . . . 595
gdcm::PresentationContextGenerator . . . 596
gdcm::network::PresentationContextRQ . . . 598
gdcm::network::PresentationDataValue . . . 600
gdcm::Printer . . . 602
gdcm::DictPrinter . . . 319
gdcm::Dumper . . . 332
gdcm::PrivateDict . . . 605
gdcm::PythonFilter . . . 613
gdcm::QueryBase . . . 614
gdcm::QueryImage . . . 617
gdcm::QueryPatient . . . 619
gdcm::QuerySeries . . . 621

gdcm::QueryStudy . . . 623
gdcm::QueryFactory . . . 616
gdcm::Reader . . . 628
gdcm::PixmapReader . . . 581
gdcm::ImageReader . . . 443
gdcm::ImageRegionReader . . . 446
gdcm::SegmentReader . . . 655
gdcm::SurfaceReader . . . 726
gdcm::Region . . . 633
gdcm::BoxRegion . . . 219
gdcm::Rescaler . . . 635
gdcm::network::RoleSelectionSub . . . 641
gdcm::SerieHelper::Rule . . . 642
gdcm::SerieHelper . . . 671
gdcm::Series . . . 674
gdcm::network::ServiceClassApplicationInformation . . . 674
gdcm::SHA1 . . . 680
gdcm::SimpleSubjectWatcher . . . 685
gdcm::SmartPointer< ObjectType > . . . 686
gdcm::SmartPointer< gdcm::Bitmap > . . . 686
gdcm::SmartPointer< gdcm::File > . . . 686
gdcm::SmartPointer< gdcm::Image > . . . 686
gdcm::SmartPointer< gdcm::MemberCommand > . . . 686
gdcm::SmartPointer< gdcm::MeshPrimitive > . . . 686
gdcm::SmartPointer< gdcm::Pixmap > . . . 686
gdcm::SmartPointer< gdcm::SimpleMemberCommand > . . . 686
gdcm::SmartPointer< gdcm::Subject > . . . 686
gdcm::SmartPointer< LookupTable > . . . 686
gdcm::SmartPointer< Segment > . . . 686
gdcm::SmartPointer< Surface > . . . 686
gdcm::SmartPointer< Value > . . . 686
gdcm::network::SOPClassExtendedNegociationSub . . . 689
gdcm::SOPClassUIDToIOD . . . 690
gdcm::Sorter . . . 691
gdcm::IPPSorter . . . 464
gdcm::Spacing . . . 695
gdcm::Spectroscopy . . . 697
gdcm::SplitMosaicFilter . . . 697
gdcm::static_assert_test< x > . . . 700
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 700
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 700
gdcm::StreamImageReader . . . 700
gdcm::StreamImageWriter . . . 703
String<'\\', 64 >
gdcm::LO . . . 494
gdcm::StringFilter . . . 712
gdcm::Study . . . 714
gdcm::SurfaceHelper . . . 724
gdcm::SwapCode . . . 730
gdcm::SwapperDoOp . . . 732
gdcm::SwapperNoOp . . . 733
gdcm::System . . . 733
gdcm::Table . . . 737

gdcm::TableEntry . . . 738
gdcm::TableReader . . . 739
gdcm::XMLDictReader . . . 923
gdcm::XMLPrivateDictReader . . . 928
gdcm::network::TableRow . . . 741
gdcm::Tag . . . 742
gdcm::PrivateTag . . . 607
gdcm::TagPath . . . 749
gdcm::Testing . . . 750
gdcm::Trace . . . 755
gdcm::TransferSyntax . . . 758
gdcm::network::TransferSyntaxSub . . . 762
gdcm::network::Transition . . . 763
gdcm::Type . . . 765
gdcm::UI . . . 766
gdcm::UIDGenerator . . . 767
gdcm::UIDs . . . 769
gdcm::network::ULAction . . . 787
gdcm::network::ULActionAA1 . . . 790
gdcm::network::ULActionAA2 . . . 791
gdcm::network::ULActionAA3 . . . 792
gdcm::network::ULActionAA4 . . . 793
gdcm::network::ULActionAA5 . . . 794
gdcm::network::ULActionAA6 . . . 795
gdcm::network::ULActionAA7 . . . 796
gdcm::network::ULActionAA8 . . . 797
gdcm::network::ULActionAE1 . . . 798
gdcm::network::ULActionAE2 . . . 799
gdcm::network::ULActionAE3 . . . 800
gdcm::network::ULActionAE4 . . . 801
gdcm::network::ULActionAE5 . . . 802
gdcm::network::ULActionAE6 . . . 803
gdcm::network::ULActionAE7 . . . 804
gdcm::network::ULActionAE8 . . . 805
gdcm::network::ULActionAR1 . . . 806
gdcm::network::ULActionAR10 . . . 807
gdcm::network::ULActionAR2 . . . 808
gdcm::network::ULActionAR3 . . . 809
gdcm::network::ULActionAR4 . . . 810
gdcm::network::ULActionAR5 . . . 811
gdcm::network::ULActionAR6 . . . 812
gdcm::network::ULActionAR7 . . . 813
gdcm::network::ULActionAR8 . . . 814
gdcm::network::ULActionAR9 . . . 815
gdcm::network::ULActionDT1 . . . 816
gdcm::network::ULActionDT2 . . . 817
gdcm::network::ULConnection . . . 819
gdcm::network::ULConnectionCallback . . . 822
gdcm::network::ULBasicCallback . . . 818
gdcm::network::ULWritingCallback . . . 829
gdcm::network::ULConnectionInfo . . . 824
gdcm::network::ULEvent . . . 828
gdcm::network::ULTransitionTable . . . 829

gdcm::Unpacker12Bits . . . 835
gdcm::Usage . . . 836
gdcm::network::UserInformation . . . 839
gdcm::UUIDGenerator . . . 840
gdcm::Validate . . . 840
gdcm::ValueIO< TDE, TSwap, TType > . . . 844
gdcm::Version . . . 845
gdcm::VL . . . 846
gdcm::VM . . . 848
gdcm::VMToLength< T > . . . 852
gdcm::VR . . . 852
gdcm::VRToEncoding< T > . . . 859
gdcm::VRToType< T > . . . 859
gdcm::VRToType< TVR > . . . 859
gdcm::VRVLSize< T > . . . 860
gdcm::VRVLSize< 0 > . . . 860
gdcm::VRVLSize< 1 > . . . 860
vtkImageAlgorithm
  vtkImagePlanarComponentsToComponents . . . 906
vtkImageMapToColors
  vtkImageMapToWindowLevelColors2 . . . 904
vtkImageWriter
  vtkGDCMImageWriter . . . 873
vtkLookupTable
  vtkLookupTable16 . . . 911
vtkMedicalImageProperties
  vtkGDCMMedicalImageProperties . . . 877
vtkMedicalImageReader2
  vtkGDCMImageReader . . . 861
    vtkGDCMThreadedImageReader . . . 888
  vtkGDCMImageReader2 . . . 867
vtkObject
  vtkGDCMTesting . . . 885
  vtkImageColorViewer . . . 894
  vtkRTStructSetProperties . . . 913
vtkPolyDataAlgorithm
  vtkGDCMPolyDataReader . . . 879
vtkPolyDataWriter
  vtkGDCMPolyDataWriter . . . 882
vtkThreadedImageAlgorithm
  vtkGDCMThreadedImageReader2 . . . 890
  vtkImageMapToColors16 . . . 901
  vtkImageRGBToYBR . . . 908
  vtkImageYBRToRGB . . . 910
gdcm::Waveform . . . 918
gdcm::Writer . . . 918
  gdcm::PixmapWriter . . . 586
    gdcm::ImageWriter . . . 451
  gdcm::SegmentWriter . . . 658
    gdcm::SurfaceWriter . . . 728
gdcm::XMLPrinter . . . 925

Chapter 24

Class Index

24.1 Class List

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

gdcm::network::AAbortPDU
AAbortPDU Table 9-26 A-ABORT PDU FIELDS . . . 139
gdcm::network::AAssociateACPDU
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields . . . 141
gdcm::network::AAssociateRJPDU
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS . . . 144
gdcm::network::AAssociateRQPDU
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields . . . 145
gdcm::AbortEvent . . . 149
gdcm::network::AbstractSyntax
AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS . . . 150
gdcm::AnonymizeEvent
AnonymizeEvent Special type of event triggered during the Anonymization process . . . 151
gdcm::Anonymizer
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode: . . . 154
gdcm::AnyEvent . . . 158
gdcm::network::ApplicationContext
ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS . . . 160
gdcm::ApplicationEntity
ApplicationEntity . . . 161
gdcm::network::AReleaseRPPDU
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields . . . 163
gdcm::network::AReleaseRQPDU
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS . . . 164
gdcm::network::ARTIMTimer
ARTIMTimer This file contains the code for the ARTIM timer . . . 166
gdcm::ASN1
Class for ASN1 . . . 167
gdcm::network::AsynchronousOperationsWindowSub
AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WIN↩
DOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ) . . . 168
gdcm::Attribute< Group, Element, TVR, TVM >
Attribute class This class use template metaprograming tricks to let the user know when the template instanciation does not match the public dictionary . . . 169

gdcm::Attribute< Group, Element, TVR, VM::VM1 > . . . 176
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > . . . 180
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > . . . 181
gdcm::Attribute< Group, Element, TVR, VM::VM1_n > . . . 183
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > . . . 187
gdcm::Attribute< Group, Element, TVR, VM::VM2_n > . . . 188
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > . . . 190
gdcm::Attribute< Group, Element, TVR, VM::VM3_n > . . . 191
gdcm::AudioCodec
AudioCodec . . . 193
gdcm::Base64
Class for Base64 . . . 195
gdcm::network::BaseCompositeMessage
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 . . . 196
gdcm::network::BasePDU
BasePDU base class for PDUs . . . 198
gdcm::BaseRootQuery
BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root . . . 200
gdcm::SegmentHelper::BasicCodedEntry
This structure defines a basic coded entry with all of its attributes . . . 204
gdcm::BasicOffsetTable
Class to represent a BasicOffsetTable . . . 207
gdcm::Bitmap
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) . . . 209
gdcm::BitmapToBitmapFilter
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image . 217
gdcm::BoxRegion
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) . . . 219
gdcm::ByteBuffer
ByteBuffer . . . 222
gdcm::ByteSwap< T >
ByteSwap . . . 223
gdcm::ByteSwapFilter
ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ?? . . . 224
gdcm::ByteValue
Class to represent binary value (array of bytes) . . . 224
gdcm::CAPICryptoFactory . . . 229
gdcm::CAPICryptographicMessageSyntax . . . 230
gdcm::network::CEchoRQ
CEchoRQ this file defines the messages for the cecho action . . . 233
gdcm::network::CEchoRSP
CEchoRSP this file defines the messages for the cecho action . . . 234
gdcm::network::CFind . . . 235
gdcm::network::CFindCancelRQ
CFindCancelRQ this file defines the messages for the cfind action . . . 236
gdcm::network::CFindRQ
CFindRQ this file defines the messages for the cfind action . . . 237
gdcm::network::CFindRSP
CFindRSP this file defines the messages for the cfind action . . . 238

gdcm::network::CMoveCancelRq . . . 239
gdcm::network::CMoveRQ
CMoveRQ this file defines the messages for the cmove action . . . 241
gdcm::network::CMoveRSP
CMoveRSP this file defines the messages for the cmove action . . . 242
gdcm::Codec
Codec class . . . 243
gdcm::Coder
Coder . . . 244
gdcm::CodeString
CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct . . . 246
gdcm::Command
Command superclass for callback/observer methods . . . 248
gdcm::CommandDataSet
Class to represent a Command DataSet . . . 250
gdcm::network::CompositeMessageFactory
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) . . . 252
gdcm::CompositeNetworkFunctions
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: . . . 253
gdcm::ConstCharWrapper
Do not use me . . . 257
gdcm::CP246ExplicitDataElement
Class to read/write a DataElement as CP246Explicit Data Element . . . 257
gdcm::CryptoFactory
Class to do handle the crypto factory . . . 259
gdcm::CryptographicMessageSyntax . . . 261
gdcm::CSAElement
Class to represent a CSA Element . . . 263
gdcm::CSAHeader
Class for CSAHeader . . . 267
gdcm::CSAHeaderDict
Class to represent a map of CSAHeaderDictEntry . . . 271
gdcm::CSAHeaderDictEntry
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 . . . 273
gdcm::CSAHeaderDictException . . . 275
gdcm::network::CStoreRQ
CStoreRQ this file defines the messages for the cecho action . . . 276
gdcm::network::CStoreRSP
CStoreRSP this file defines the messages for the cecho action . . . 277

gdcm::Curve
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004 . . . 279
gdcm::DataElement
Class to represent a Data Element either Implicit or Explicit . . . 282
gdcm::DataElementException . . . 290
gdcm::DataEvent
DataEvent . . . 291
gdcm::DataSet
Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object . . . 293
gdcm::DataSetEvent
DataSetEvent Special type of event triggered during the DataSet store/move process . . . 300
gdcm::DataSetHelper
DataSetHelper (internal class, not intended for user level) . . . 303
gdcm::Decoder
Decoder . . . 303
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 . . . 305
gdcm::Defs
FIXME I do not like the name 'Defs' . . . 305
gdcm::DeltaEncodingCodec
DeltaEncodingCodec compression used by some private vendor . . . 308
gdcm::DICOMDIR
DICOMDIR class . . . 309
gdcm::DICOMDIRGenerator
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 . . . 310
gdcm::Dict
Class to represent a map of DictEntry . . . 312
gdcm::DictConverter
Class to convert a .dic file into something else: . . . 315
gdcm::DictEntry
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 . . . 317
gdcm::DictPrinter
DictPrinter class . . . 319
gdcm::Dicts
Class to manipulate the sum of knowledge (all the dict user load) . . . 321
gdcm::network::DIMSE
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) . . . 324
gdcm::DirectionCosines
Class to handle DirectionCosines . . . 325
gdcm::Directory
Class for manipulation directories . . . 327

gdcm::DirectoryHelper
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 . . . 330
gdcm::DummyValueGenerator
Class for generating dummy value . . . 331
gdcm::Dumper
Codec class . . . 332
gdcm::Element< TVR, TVM >
Element class . . . 334
gdcm::Element< TVR, VM::VM1_2 > . . . 337
gdcm::Element< TVR, VM::VM1_n > . . . 338
gdcm::Element< TVR, VM::VM2_2n > . . . 340
gdcm::Element< TVR, VM::VM2_n > . . . 342
gdcm::Element< TVR, VM::VM3_3n > . . . 343
gdcm::Element< TVR, VM::VM3_n > . . . 345
gdcm::Element< VR::AS, VM::VM5 > . . . 346
gdcm::Element< VR::OB, VM::VM1 > . . . 347
gdcm::Element< VR::OW, VM::VM1 > . . . 348
gdcm::ElementDisableCombinations< TVR, TVM >
A class which is used to produce compile errors for an invalid combination of template parameters . 350
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . 351
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . 351
gdcm::EncapsulatedDocument
EncapsulatedDocument . . . 351
gdcm::EncodingImplementation< T >
EncodingImplementation . . . 352
gdcm::EncodingImplementation< VR::VRASCII > . . . 352
gdcm::EncodingImplementation< VR::VRBINARY > . . . 353
gdcm::EndEvent . . . 354
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: . . . 355
gdcm::Event
Superclass for callback/observer methods . . . 356
gdcm::Exception
Exception . . . 359
gdcm::ExitEvent . . . 360
gdcm::ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 362
gdcm::ExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element . . . 363
gdcm::Fiducials
Fiducials . . . 365
gdcm::File
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 . . . 366
gdcm::FileAnonymizer
FileAnonymizer . . . 369

gdcm::FileChangeTransferSyntax
FileChangeTransferSyntax . . . 372
gdcm::FileDerivation
FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence . . . 374
gdcm::FileExplicitFilter
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 reprensentation, the VR is not stored directly in the file . . . 377
gdcm::FileMetaInformation
Class to represent a File Meta Information . . . 379
gdcm::Filename
Class to manipulate file name's . . . 384
gdcm::FileNameEvent
FileNameEvent Special type of event triggered during processing of FileSet . . . 386
gdcm::FilenameGenerator
FilenameGenerator . . . 388
gdcm::FileSet
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 . . . 391
gdcm::FileStreamer
FileStreamer This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data . . . 392
gdcm::FileWithName
FileWithName . . . 396
gdcm::FindPatientRootQuery
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root . . . 397
gdcm::FindStudyRootQuery
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root . . . 399
gdcm::Fragment
Class to represent a Fragment . . . 401
gdcm::Global
Global . . . 404
gdcm::GroupDict
Class to represent the mapping from group number to its abbreviation and name . . . 406
gdcm::IconImageFilter
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 408
gdcm::IconImageGenerator
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: . . . 410
gdcm::ignore_char . . . 412
gdcm::Image
Image This is the container for an Image in the general sense. From this container you should be able to request information like: . . . 413
gdcm::ImageApplyLookupTable
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image . . . 417
gdcm::ImageChangePhotometricInterpretation
ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM . . . 419

gdcm::ImageChangePlanarConfiguration
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0 422
gdcm::ImageChangeTransferSyntax
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM 425
gdcm::ImageCodec
ImageCodec 429
gdcm::ImageConverter
Image Converter 435
gdcm::ImageFragmentSplitter
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments 436
gdcm::ImageHelper
ImageHelper (internal class, not intended for user level) 439
gdcm::ImageReader
ImageReader 443
gdcm::ImageRegionReader
ImageRegionReader 446
gdcm::ImageToImageFilter
ImageToImageFilter class Super class for all filter taking an image and producing an output image . 449
gdcm::ImageWriter
ImageWriter 451
gdcm::network::ImplementationClassUIDSub
ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIE↩ LDS (A-ASSOCIATE-RQ) 454
gdcm::network::ImplementationUIDSub
ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-↩ AC) 454
gdcm::network::ImplementationVersionNameSub
ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FI↩ ELDS (A-ASSOCIATE-RQ) 455
gdcm::ImplicitDataElement
Class to represent an *Implicit VR* Data Element 456
gdcm::InitializeEvent 458
gdcm::IOD
Class for representing a IOD 459
gdcm::IODEntry
Class for representing a IODEntry 460
gdcm::IODs
Class for representing a IODs 462
gdcm::IPPSorter
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 464
gdcm::Item
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 Standart 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 467
gdcm::IterationEvent 470
gdcm::JPEG12Codec
Class to do JPEG 12bits (lossy & lossless) 472
gdcm::JPEG16Codec
Class to do JPEG 16bits (lossless) 474

gdcm::JPEG2000Codec
Class to do JPEG 2000 . . . 476
gdcm::JPEG8Codec
Class to do JPEG 8bits (lossy & lossless) . . . 480
gdcm::JPEGCodec
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 . . . 482
gdcm::JPEGLSCodec
JPEG-LS . . . 487
gdcm::JSON . . . 490
gdcm::KAKADUCodec
KAKADUCodec . . . 492
gdcm::LO
LO . . . 494
gdcm::LookupTable
LookupTable class . . . 496
gdcm::Scanner::ltstr . . . 500
gdcm::Macro
Class for representing a Macro . . . 501
gdcm::Macros
Class for representing a Modules . . . 502
gdcm::network::MaximumLengthSub
MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ) . . . 504
gdcm::MD5
Class for MD5 . . . 505
gdcm::MediaStorage
MediaStorage . . . 506
gdcm::MemberCommand< T >
Command subclass that calls a pointer to a member function . . . 513
gdcm::MeshPrimitive
This class defines surface mesh primitives. It is designed from surface mesh primitives macro . . . 517
gdcm::ModifiedEvent . . . 520
gdcm::Module
Class for representing a Module . . . 522
gdcm::ModuleEntry
Class for representing a ModuleEntry . . . 524
gdcm::Modules
Class for representing a Modules . . . 526
gdcm::MovePatientRootQuery
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root 528
gdcm::MoveStudyRootQuery
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root . 530
gdcm::NestedModuleEntries
Class for representing a NestedModuleEntries . . . 533
gdcm::NoEvent . . . 535
gdcm::Object
Object . . . 536
gdcm::OpenSSLCryptoFactory . . . 538
gdcm::OpenSSLCryptographicMessageSyntax . . . 539
gdcm::OpenSSLP7CryptoFactory . . . 542

gdcm::OpenSSLP7CryptographicMessageSyntax
Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities . . . 543
gdcm::Orientation
Class to handle Orientation . . . 545
gdcm::Overlay
Overlay class . . . 547
gdcm::ParseException
ParseException Standard exception handling object . . . 553
gdcm::Parser
Parser ala XML_Parser from expat (SAX) . . . 555
gdcm::Patient
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54 . . . 557
gdcm::network::PDataTFPDU
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS . . . 558
gdcm::PDBElement
Class to represent a PDB Element . . . 560
gdcm::PDBHeader
Class for PDBHeader . . . 562
gdcm::PDFCodec
PDFCodec class . . . 564
gdcm::network::PDUFactory
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 . . . 566
gdcm::PersonName
PersonName class . . . 567
gdcm::PGXCodec
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images . . . 568
gdcm::PhotometricInterpretation
Class to represent an PhotometricInterpretation . . . 570
gdcm::PixelFormat
PixelFormat . . . 573
gdcm::Pixmap
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) . . . 578
gdcm::PixmapReader
PixmapReader . . . 581
gdcm::PixmapToPixmapFilter
PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image 584
gdcm::PixmapWriter
PixmapWriter This class will takes two inputs: . . . 586
gdcm::PNMCodec
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/` . . . 589
gdcm::Preamble
DICOM Preamble (Part 10) . . . 592
gdcm::PresentationContext
PresentationContext . . . 593
gdcm::network::PresentationContextAC
PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS . . . 595

gdcm::PresentationContextGenerator
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 . . . 596
gdcm::network::PresentationContextRQ
PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS . . . 598
gdcm::network::PresentationDataValue
PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS . . . 600
gdcm::Printer
Printer class . . . 602
gdcm::PrivateDict
Private Dict . . . 605
gdcm::PrivateTag
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner) . . . 607
gdcm::ProgressEvent
ProgressEvent Special type of event triggered during . . . 609
gdcm::PVRGCodec
PVRGCodec . . . 611
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 . . . 613
gdcm::QueryBase
QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE . 614
gdcm::QueryFactory
QueryFactory.h . . . 616
gdcm::QueryImage
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE . . . 617
gdcm::QueryPatient
QueryPatient contains: class to construct a patient-based query for c-find and c-move . . . 619
gdcm::QuerySeries
QuerySeries contains: class to construct a series-based query for c-find and c-move . . . 621
gdcm::QueryStudy
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE . . . 623
gdcm::RAWCodec
RAWCodec class . . . 625
gdcm::Reader
Reader ala DOM (Document Object Model) . . . 628
gdcm::Region
Class for manipulation region . . . 633
gdcm::Rescaler
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 . . . 635
gdcm::RLECodec
Class to do RLE . . . 638

gdcm::network::RoleSelectionSub
RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSO↩ CIATE-RQ) . . . 641
gdcm::SerieHelper::Rule . . . 642
gdcm::Scanner
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 mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute . . . 643
gdcm::Segment
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface . . . 649
gdcm::SegmentedPaletteColorLookupTable
SegmentedPaletteColorLookupTable class . . . 653
gdcm::SegmentReader
This class defines a segment reader. It reads attributes of group 0x0062 . . . 655
gdcm::SegmentWriter
This class defines a segment writer. It writes attributes of group 0x0062 . . . 658
gdcm::SequenceOfFragments
Class to represent a Sequence Of Fragments . . . 660
gdcm::SequenceOfItems
Class to represent a Sequence Of Items (value representation : SQ) . . . 665
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 disapear soon, you've been warned . . . 671
gdcm::Series
Series . . . 674
gdcm::network::ServiceClassApplicationInformation . . . 674
gdcm::ServiceClassUser
ServiceClassUser . . . 675
gdcm::SHA1
Class for SHA1 . . . 680
gdcm::SimpleMemberCommand< T >
Command subclass that calls a pointer to a member function . . . 681
gdcm::SimpleSubjectWatcher
SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events . . . 685
gdcm::SmartPointer< ObjectType >
Class for Smart Pointer . . . 686
gdcm::network::SOPClassExtendedNegociationSub
SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC) . . . 689
gdcm::SOPClassUIDToIOD
Class convert a class SOP Class UID into IOD . . . 690
gdcm::Sorter
Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction . . . 691
gdcm::Spacing
Class for Spacing . . . 695
gdcm::Spectroscopy
Spectroscopy class . . . 697
gdcm::SplitMosaicFilter
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 . . . 697
gdcm::StartEvent . . . 698

gdcm::static_assert_test< x > . . . 700
gdcm::STATIC_ASSERTION_FAILURE< x > . . . 700
gdcm::STATIC_ASSERTION_FAILURE< true > . . . 700
gdcm::StreamImageReader
StreamImageReader . . . 700
gdcm::StreamImageWriter
StreamImageReader . . . 703
gdcm::String< TDelimiter, TMaxLength, TPadChar >
String . . . 708
gdcm::StringFilter
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 . . . 712
gdcm::Study
Study . . . 714
gdcm::Subject
Subject . . . 715
gdcm::Surface
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes . . . 717
gdcm::SurfaceHelper
SurfaceHelper Helper class for Surface object . . . 724
gdcm::SurfaceReader
This class defines a SURFACE IE reader. It reads surface mesh module attributes . . . 726
gdcm::SurfaceWriter
This class defines a SURFACE IE writer. It writes surface mesh module attributes . . . 728
gdcm::SwapCode
SwapCode representation . . . 730
gdcm::SwapperDoOp . . . 732
gdcm::SwapperNoOp . . . 733
gdcm::System
Class to do system operation . . . 733
gdcm::Table
Table . . . 737
gdcm::TableEntry
TableEntry . . . 738
gdcm::TableReader
Class for representing a TableReader . . . 739
gdcm::network::TableRow . . . 741
gdcm::Tag
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) . . . 742
gdcm::TagPath
Class to handle a path of tag . . . 749
gdcm::Testing
Class for testing . . . 750
gdcm::Trace
Trace . . . 755
gdcm::TransferSyntax
Class to manipulate Transfer Syntax . . . 758
gdcm::network::TransferSyntaxSub
TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS . . . 762
gdcm::network::Transition . . . 763

gdcm::Type
Type . . . 765
gdcm::UI . . . 766
gdcm::UIDGenerator
Class for generating unique UID . . . 767
gdcm::UIDs
All known uids . . . 769
gdcm::network::ULAction
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 . . . 787
gdcm::network::ULActionAA1 . . . 790
gdcm::network::ULActionAA2 . . . 791
gdcm::network::ULActionAA3 . . . 792
gdcm::network::ULActionAA4 . . . 793
gdcm::network::ULActionAA5 . . . 794
gdcm::network::ULActionAA6 . . . 795
gdcm::network::ULActionAA7 . . . 796
gdcm::network::ULActionAA8 . . . 797
gdcm::network::ULActionAE1 . . . 798
gdcm::network::ULActionAE2 . . . 799
gdcm::network::ULActionAE3 . . . 800
gdcm::network::ULActionAE4 . . . 801
gdcm::network::ULActionAE5 . . . 802
gdcm::network::ULActionAE6 . . . 803
gdcm::network::ULActionAE7 . . . 804
gdcm::network::ULActionAE8 . . . 805
gdcm::network::ULActionAR1 . . . 806
gdcm::network::ULActionAR10 . . . 807
gdcm::network::ULActionAR2 . . . 808
gdcm::network::ULActionAR3 . . . 809
gdcm::network::ULActionAR4 . . . 810
gdcm::network::ULActionAR5 . . . 811
gdcm::network::ULActionAR6 . . . 812
gdcm::network::ULActionAR7 . . . 813
gdcm::network::ULActionAR8 . . . 814
gdcm::network::ULActionAR9 . . . 815
gdcm::network::ULActionDT1 . . . 816
gdcm::network::ULActionDT2 . . . 817
gdcm::network::ULBasicCallback
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 . . . 818
gdcm::network::ULConnection
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 . . . 819
gdcm::network::ULConnectionCallback . . . 822
gdcm::network::ULConnectionInfo
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 . . . 824

gdcm::network::ULConnectionManager
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) . . . 825
gdcm::network::ULEvent
ULEvent base class for network events . . . 828
gdcm::network::ULTransitionTable
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates . . . 829
gdcm::network::ULWritingCallback . . . 829
gdcm::UNExplicitDataElement
Class to read/write a DataElement as UNExplicit Data Element . . . 831
gdcm::UNExplicitImplicitDataElement
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs: . . . 833
gdcm::Unpacker12Bits
Pack/Unpack 12 bits pixel into 16bits . . . 835
gdcm::Usage
Usage . . . 836
gdcm::UserEvent . . . 837
gdcm::network::UserInformation
UserInformation Table 9-16 USER INFORMATION ITEM FIELDS . . . 839
gdcm::UUIDGenerator
Class for generating unique UUID generate DCE 1.1 uid . . . 840
gdcm::Validate
Validate class . . . 840
gdcm::Value
Class to represent the value of a Data Element . . . 842
gdcm::ValueIO< TDE, TSwap, TType >
Class to dispatch template calls . . . 844
gdcm::Version
Major/minor and build version . . . 845
gdcm::VL
Value Length . . . 846
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 . . . 848
gdcm::VMToLength< T > . . . 852
gdcm::VR
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 . . . 852
gdcm::VR16ExplicitDataElement
Class to read/write a DataElement as Explicit Data Element . . . 857
gdcm::VRToEncoding< T > . . . 859
gdcm::VRToType< T > . . . 859
gdcm::VRVLSize< T > . . . 860
gdcm::VRVLSize< 0 > . . . 860
gdcm::VRVLSize< 1 > . . . 860
vtkGDCMImageReader . . . 861
vtkGDCMImageReader2 . . . 867
vtkGDCMImageWriter . . . 873
vtkGDCMMedicalImageProperties . . . 877
vtkGDCMPolyDataReader . . . 879
vtkGDCMPolyDataWriter . . . 882

vtkGDCMTesting . . . 885
vtkGDCMThreadedImageReader . . . 888
vtkGDCMThreadedImageReader2 . . . 890
vtkImageColorViewer . . . 894
vtkImageMapToColors16 . . . 901
vtkImageMapToWindowLevelColors2 . . . 904
vtkImagePlanarComponentsToComponents . . . 906
vtkImageRGBToYBR . . . 908
vtkImageYBRToRGB . . . 910
vtkLookupTable16 . . . 911
vtkRTStructSetProperties . . . 913
gdcm::Waveform
Waveform class . . . 918
gdcm::Writer
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only . . . 918
gdcm::XMLDictReader
Class for representing a XMLDictReader . . . 923
gdcm::XMLPrinter . . . 925
gdcm::XMLPrivateDictReader
Class for representing a XMLPrivateDictReader . . . 928

Chapter 25

File Index

25.1 File List

Here is a list of all files with brief descriptions:

gdcmAAbortPDU.h . . . 931
gdcmAAssociateACPDU.h . . . 932
gdcmAAssociateRJPDU.h . . . 932
gdcmAAssociateRQPDU.h . . . 933
gdcmAbstractSyntax.h . . . 934
gdcmAnonymizeEvent.h . . . 935
gdcmAnonymizer.h . . . 936
gdcmApplicationContext.h . . . 937
gdcmApplicationEntity.h . . . 938
gdcmAReleaseRPPDU.h . . . 939
gdcmAReleaseRQPDU.h . . . 940
gdcmARTIMTimer.h . . . 941
gdcmASN1.h . . . 942
gdcmAsynchronousOperationsWindowSub.h . . . 943
gdcmAttribute.h . . . 943
gdcmAudioCodec.h . . . 945
gdcmBase64.h . . . 945
gdcmBaseCompositeMessage.h . . . 946
gdcmBasePDU.h . . . 947
gdcmBaseRootQuery.h . . . 948
gdcmBasicOffsetTable.h . . . 949
gdcmBitmap.h . . . 951
gdcmBitmapToBitmapFilter.h . . . 952
gdcmBoxRegion.h . . . 953
gdcmByteBuffer.h . . . 953
gdcmByteSwap.h . . . 954
gdcmByteSwapFilter.h . . . 955
gdcmByteValue.h . . . 956
gdcmCAPICryptoFactory.h . . . 957
gdcmCAPICryptographicMessageSyntax.h . . . 957
gdcmCEchoMessages.h . . . 958
gdcmCFindMessages.h . . . 959
gdcmCMoveMessages.h . . . 960
gdcmCodec.h . . . 961

gdcmCoder.h . . . 962
gdcmCodeString.h . . . 963
gdcmCommand.h . . . 964
gdcmCommandDataSet.h . . . 966
gdcmCompositeMessageFactory.h . . . 966
gdcmCompositeNetworkFunctions.h . . . 967
gdcmConstCharWrapper.h . . . 968
gdcmCP246ExplicitDataElement.h . . . 969
gdcmCryptoFactory.h . . . 969
gdcmCryptographicMessageSyntax.h . . . 970
gdcmCSAElement.h . . . 971
gdcmCSAHeader.h . . . 972
gdcmCSAHeaderDict.h . . . 973
gdcmCSAHeaderDictEntry.h . . . 975
gdcmCStoreMessages.h . . . 976
gdcmCurve.h . . . 977
gdcmDataElement.h . . . 978
gdcmDataEvent.h . . . 980
gdcmDataSet.h . . . 981
gdcmDataSetEvent.h . . . 982
gdcmDataSetHelper.h . . . 982
gdcmDecoder.h . . . 983
gdcmDefinedTerms.h . . . 984
gdcmDeflateStream.h . . . 985
gdcmDefs.h . . . 985
gdcmDeltaEncodingCodec.h . . . 987
gdcmDICOMDIR.h . . . 987
gdcmDICOMDIRGenerator.h . . . 988
gdcmDict.h . . . 989
gdcmDictConverter.h . . . 991
gdcmDictEntry.h . . . 991
gdcmDictPrinter.h . . . 993
gdcmDicts.h . . . 993
gdcmDIMSE.h . . . 995
gdcmDirectionCosines.h . . . 995
gdcmDirectory.h . . . 996
gdcmDirectoryHelper.h . . . 997
gdcmDummyValueGenerator.h . . . 998
gdcmDumper.h . . . 999
gdcmElement.h . . . 999
gdcmEncapsulatedDocument.h . . . 1001
gdcmEnumeratedValues.h . . . 1001
gdcmEvent.h . . . 1002
gdcmException.h . . . 1004
gdcmExplicitDataElement.h . . . 1004
gdcmExplicitImplicitDataElement.h . . . 1005
gdcmFiducials.h . . . 1006
gdcmFile.h . . . 1007
gdcmFileAnonymizer.h . . . 1008
gdcmFileChangeTransferSyntax.h . . . 1008
gdcmFileDerivation.h . . . 1009
gdcmFileExplicitFilter.h . . . 1010
gdcmFileMetaInformation.h . . . 1011
gdcmFilename.h . . . 1012

gdcmFileNameEvent.h . . . 1013
gdcmFilenameGenerator.h . . . 1014
gdcmFileSet.h . . . 1015
gdcmFileStreamer.h . . . 1016
gdcmFindPatientRootQuery.h . . . 1017
gdcmFindStudyRootQuery.h . . . 1018
gdcmFragment.h . . . 1019
gdcmGlobal.h . . . 1021
gdcmGroupDict.h . . . 1022
gdcmIconImage.h . . . 1022
gdcmIconImageFilter.h . . . 1023
gdcmIconImageGenerator.h . . . 1024
gdcmImage.h . . . 1025
gdcmImageApplyLookupTable.h . . . 1026
gdcmImageChangePhotometricInterpretation.h . . . 1027
gdcmImageChangePlanarConfiguration.h . . . 1028
gdcmImageChangeTransferSyntax.h . . . 1029
gdcmImageCodec.h . . . 1030
gdcmImageConverter.h . . . 1031
gdcmImageFragmentSplitter.h . . . 1032
gdcmImageHelper.h . . . 1033
gdcmImageReader.h . . . 1034
gdcmImageRegionReader.h . . . 1034
gdcmImageToImageFilter.h . . . 1035
gdcmImageWriter.h . . . 1036
gdcmImplementationClassUIDSub.h . . . 1037
gdcmImplementationUIDSub.h . . . 1038
gdcmImplementationVersionNameSub.h . . . 1039
gdcmImplicitDataElement.h . . . 1040
gdcmIOD.h . . . 1041
gdcmIODEntry.h . . . 1043
gdcmIODs.h . . . 1045
gdcmIPPSorter.h . . . 1046
gdcmItem.h . . . 1047
gdcmJPEG12Codec.h . . . 1048
gdcmJPEG16Codec.h . . . 1049
gdcmJPEG2000Codec.h . . . 1049
gdcmJPEG8Codec.h . . . 1050
gdcmJPEGCodec.h . . . 1051
gdcmJPEGLSCodec.h . . . 1052
gdcmJSON.h . . . 1053
gdcmKAKADUCodec.h . . . 1054
gdcmLegacyMacro.h . . . 1055
gdcmLO.h . . . 1056
gdcmLookupTable.h . . . 1056
gdcmMacro.h . . . 1057
gdcmMacroEntry.h . . . 1059
gdcmMacros.h . . . 1061
gdcmMaximumLengthSub.h . . . 1063
gdcmMD5.h . . . 1064
gdcmMediaStorage.h . . . 1065
gdcmMeshPrimitive.h . . . 1066
gdcmModule.h . . . 1067
gdcmModuleEntry.h . . . 1069

gdcmModules.h . . . 1071
gdcmMovePatientRootQuery.h . . . 1072
gdcmMoveStudyRootQuery.h . . . 1073
gdcmNestedModuleEntries.h . . . 1074
gdcmNetworkEvents.h . . . 1076
gdcmNetworkStateID.h . . . 1077
gdcmObject.h . . . 1078
gdcmOpenSSLCryptoFactory.h . . . 1079
gdcmOpenSSLCryptographicMessageSyntax.h . . . 1079
gdcmOpenSSLP7CryptoFactory.h . . . 1080
gdcmOpenSSLP7CryptographicMessageSyntax.h . . . 1081
gdcmOrientation.h . . . 1083
gdcmOverlay.h . . . 1083
gdcmParseException.h . . . 1084
gdcmParser.h . . . 1086
gdcmPatient.h . . . 1086
gdcmPDataTFPDU.h . . . 1087
gdcmPDBElement.h . . . 1088
gdcmPDBHeader.h . . . 1090
gdcmPDFCodec.h . . . 1090
gdcmPDUFactory.h . . . 1091
gdcmPersonName.h . . . 1092
gdcmPGXCodec.h . . . 1093
gdcmPhotometricInterpretation.h . . . 1093
gdcmPixelFormat.h . . . 1094
gdcmPixmap.h . . . 1095
gdcmPixmapReader.h . . . 1096
gdcmPixmapToPixmapFilter.h . . . 1097
gdcmPixmapWriter.h . . . 1098
gdcmPNMCodec.h . . . 1099
gdcmPreamble.h . . . 1100
gdcmPresentationContext.h . . . 1101
gdcmPresentationContextAC.h . . . 1102
gdcmPresentationContextGenerator.h . . . 1104
gdcmPresentationContextRQ.h . . . 1104
gdcmPresentationDataValue.h . . . 1105
gdcmPrinter.h . . . 1106
gdcmPrivateTag.h . . . 1107
gdcmProgressEvent.h . . . 1109
gdcmPVRGCodec.h . . . 1109
gdcmPythonFilter.h . . . 1110
gdcmQueryBase.h . . . 1111
gdcmQueryFactory.h . . . 1112
gdcmQueryImage.h . . . 1113
gdcmQueryPatient.h . . . 1114
gdcmQuerySeries.h . . . 1115
gdcmQueryStudy.h . . . 1116
gdcmRAWCodec.h . . . 1117
gdcmReader.h . . . 1118
gdcmRegion.h . . . 1119
gdcmRescaler.h . . . 1120
gdcmRLECodec.h . . . 1121
gdcmRoleSelectionSub.h . . . 1121
gdcmScanner.h . . . 1122

gdcmSegment.h . . . 1123
gdcmSegmentedPaletteColorLookupTable.h . . . 1125
gdcmSegmentHelper.h . . . 1125
gdcmSegmentReader.h . . . 1127
gdcmSegmentWriter.h . . . 1128
gdcmSequenceOfFragments.h . . . 1129
gdcmSequenceOfItems.h . . . 1129
gdcmSerieHelper.h . . . 1130
gdcmSeries.h . . . 1132
gdcmServiceClassApplicationInformation.h . . . 1133
gdcmServiceClassUser.h . . . 1134
gdcmSHA1.h . . . 1134
gdcmSimpleSubjectWatcher.h . . . 1135
gdcmSmartPointer.h . . . 1136
gdcmSOPClassExtendedNegociationSub.h . . . 1137
gdcmSOPClassUIDToIOD.h . . . 1138
gdcmSorter.h . . . 1139
gdcmSpacing.h . . . 1141
gdcmSpectroscopy.h . . . 1141
gdcmSplitMosaicFilter.h . . . 1142
gdcmStaticAssert.h . . . 1143
gdcmStreamImageReader.h . . . 1144
gdcmStreamImageWriter.h . . . 1144
gdcmString.h . . . 1145
gdcmStringFilter.h . . . 1146
gdcmStudy.h . . . 1147
gdcmSubject.h . . . 1148
gdcmSurface.h . . . 1149
gdcmSurfaceHelper.h . . . 1150
gdcmSurfaceReader.h . . . 1151
gdcmSurfaceWriter.h . . . 1152
gdcmSwapCode.h . . . 1153
gdcmSwapper.h . . . 1154
gdcmSystem.h . . . 1155
gdcmTable.h . . . 1156
gdcmTableEntry.h . . . 1157
gdcmTableReader.h . . . 1158
gdcmTag.h . . . 1160
gdcmTagPath.h . . . 1161
gdcmTagToVR.h . . . 1161
gdcmTerminal.h . . . 1162
gdcmTestDriver.h . . . 1164
gdcmTesting.h . . . 1164
gdcmTrace.h . . . 1165
gdcmTransferSyntax.h . . . 1168
gdcmTransferSyntaxSub.h . . . 1169
gdcmType.h . . . 1170
gdcmTypes.h . . . 1171
gdcmUIDGenerator.h . . . 1172
gdcmUIDs.h . . . 1173
gdcmULAction.h . . . 1173
gdcmULActionAA.h . . . 1174
gdcmULActionAE.h . . . 1175
gdcmULActionAR.h . . . 1176

gdcmULActionDT.h . . . 1177
gdcmULBasicCallback.h . . . 1177
gdcmULConnection.h . . . 1178
gdcmULConnectionCallback.h . . . 1179
gdcmULConnectionInfo.h . . . 1180
gdcmULConnectionManager.h . . . 1182
gdcmULEvent.h . . . 1182
gdcmULTransitionTable.h . . . 1183
gdcmULWritingCallback.h . . . 1185
gdcmUNExplicitDataElement.h . . . 1185
gdcmUNExplicitImplicitDataElement.h . . . 1186
gdcmUnpacker12Bits.h . . . 1187
gdcmUsage.h . . . 1187
gdcmUserInformation.h . . . 1190
gdcmUUIDGenerator.h . . . 1191
gdcmValidate.h . . . 1191
gdcmValue.h . . . 1192
gdcmValueIO.h . . . 1193
gdcmVersion.h . . . 1194
gdcmVL.h . . . 1195
gdcmVM.h . . . 1196
gdcmVR.h . . . 1197
gdcmVR16ExplicitDataElement.h . . . 1200
gdcmWaveform.h . . . 1200
gdcmWin32.h . . . 1201
gdcmWriter.h . . . 1202
gdcmXMLDictReader.h . . . 1203
gdcmXMLPrinter.h . . . 1203
gdcmXMLPrivateDictReader.h . . . 1204
vtkGDCMImageReader.h . . . 1205
vtkGDCMImageReader2.h . . . 1206
vtkGDCMImageWriter.h . . . 1207
vtkGDCMMedicalImageProperties.h . . . 1207
vtkGDCMPolyDataReader.h . . . 1208
vtkGDCMPolyDataWriter.h . . . 1209
vtkGDCMTesting.h . . . 1209
vtkGDCMThreadedImageReader.h . . . 1210
vtkGDCMThreadedImageReader2.h . . . 1211
vtkImageColorViewer.h . . . 1211
vtkImageMapToColors16.h . . . 1212
vtkImageMapToWindowLevelColors2.h . . . 1212
vtkImagePlanarComponentsToComponents.h . . . 1213
vtkImageRGBToYBR.h . . . 1213
vtkImageYBRToRGB.h . . . 1214
vtkLookupTable16.h . . . 1214
vtkRTStructSetProperties.h . . . 1215

Chapter 26

Namespace Documentation

26.1 gdcm Namespace Reference

Namespaces

- network
- SegmentHelper
- terminal

  Class for Terminal Allow one to print in color in a shell.

Classes

- class AbortEvent
- class AnonymizeEvent

  AnonymizeEvent Special type of event triggered during the Anonymization process.
- class Anonymizer

  Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:
- 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 instanciation 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 BaseRootQuery

  BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.

- class BasicOffsetTable

  Class to represent a BasicOffsetTable.

- class Bitmap

  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)

- class BitmapToBitmapFilter

  BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

- class BoxRegion

  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)

- class ByteBuffer

  ByteBuffer.

- class ByteSwap

  ByteSwap.

- class ByteSwapFilter

  ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

- class ByteValue

  Class to represent binary value (array of bytes)

- class CAPICryptoFactory
- class CAPICryptographicMessageSyntax
- class Codec

  Codec class.

- class Coder

  Coder.

- class CodeString

  CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

- class Command

  Command superclass for callback/observer methods.

- class CommandDataSet

  Class to represent a Command DataSet.

- class CompositeNetworkFunctions

  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:

- 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 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.
- class CSAHeaderDictException
- class Curve

  Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.
- 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) A Data Set represents an instance of a real world Information Object.
- class DataSetEvent

  DataSetEvent Special type of event triggered during the DataSet store/move process.
- 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 This is a STD-GEN-CD DICOMDIR generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.
- 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 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.

- 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 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.

- 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_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 EncapsulatedDocument

  EncapsulatedDocument.

- singleton 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 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 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.
- class FileAnonymizer

  FileAnonymizer.
- class FileChangeTransferSyntax

  FileChangeTransferSyntax.
- class FileDerivation

  FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.
- class FileExplicitFilter

  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 reprensentation, the VR is not stored directly in the file.
- class FileMetaInformation

  Class to represent a File Meta Information.
- class Filename

  Class to manipulate file name's.
- class FileNameEvent

  FileNameEvent Special type of event triggered during processing of FileSet.
- class FilenameGenerator

  FilenameGenerator.
- class FileSet

  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.
- class FileStreamer

  FileStreamer This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data.
- class FileWithName

  FileWithName.
- class FindPatientRootQuery

  PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.
- class FindStudyRootQuery

  FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.
- 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 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.

- class IconImageGenerator

  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:
- struct ignore_char
- class Image

  Image This is the container for an Image in the general sense. From this container you should be able to request information like:
- class ImageApplyLookupTable

  ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.
- class ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.
- class ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.
- class ImageChangeTransferSyntax

  ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.
- class ImageCodec

  ImageCodec.
- class ImageConverter

  Image Converter.
- class ImageFragmentSplitter

  ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.
- class ImageHelper

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

  ImageReader.
- class ImageRegionReader

  ImageRegionReader.
- class ImageToImageFilter

  ImageToImageFilter class Super class for all filter taking an image and producing an output image.
- 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 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.

- class Item

  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 Standart 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.
- 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 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.
- 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 MediaStorage

  MediaStorage.
- class MemberCommand

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

  This class defines surface mesh primitives. It is designed from surface mesh primitives macro.
- 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 contains: the class which will produce a dataset for c-move with patient root.

- class MoveStudyRootQuery

  MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.
- class NestedModuleEntries

  Class for representing a NestedModuleEntries.
- class NoEvent
- class Object

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

  Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.
- 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 See PGX as used in JPEG 2000 implementation and reference images.
- class PhotometricInterpretation

  Class to represent an PhotometricInterpretation.
- class PixelFormat

  PixelFormat.
- class Pixmap

  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)
- class PixmapReader

  PixmapReader.
- class PixmapToPixmapFilter

  PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.
- class PixmapWriter

  PixmapWriter This class will takes two inputs:
- class PNMCodec

Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.sourceforge.net/`.

- class Preamble

  DICOM Preamble (Part 10)
- class PresentationContext

  PresentationContext.
- class PresentationContextGenerator

  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.
- 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 Special type of event triggered during.
- 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 contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.
- class QueryFactory

  QueryFactory.h.
- class QueryImage

  QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.
- class QueryPatient

  QueryPatient contains: class to construct a patient-based query for c-find and c-move.
- class QuerySeries

  QuerySeries contains: class to construct a series-based query for c-find and c-move.
- class QueryStudy

  QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.
- class RAWCodec

  RAWCodec class.
- class Reader

  Reader ala DOM (Document Object Model)
- class Region

  Class for manipulation region.
- class Rescaler

  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.

- class RLECodec

  Class to do RLE.
- class Scanner

  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 mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.
- class Segment

  This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.
- class SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.
- class SegmentReader

  This class defines a segment reader. It reads attributes of group 0x0062.
- class SegmentWriter

  This class defines a segment writer. It writes attributes of group 0x0062.
- class SequenceOfFragments

  Class to represent a Sequence Of Fragments.
- class SequenceOfItems

  Class to represent a Sequence Of Items (value representation : SQ)
- 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 disapear 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 This is a typical Subject Watcher class. It will observe all events.
- class SmartPointer

  Class for Smart Pointer.
- class SOPClassUIDToIOD

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

  Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction.
- class Spacing

  Class for Spacing.
- class Spectroscopy

  Spectroscopy class.
- class SplitMosaicFilter

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

- class StreamImageReader

  StreamImageReader.
- class StreamImageWriter

  StreamImageReader.
- class String

  String.
- class StringFilter

  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.
- class Study

  Study.
- class Subject

  Subject.
- class Surface

  This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.
- class SurfaceHelper

  SurfaceHelper Helper class for Surface object.
- class SurfaceReader

  This class defines a SURFACE IE reader. It reads surface mesh module attributes.
- class SurfaceWriter

  This class defines a SURFACE IE writer. It writes surface mesh module attributes.
- 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). Basically an uint32_t which can also be expressed as two uint16_t (group and 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 This class gather two known bugs:

- class Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.

- class Usage

  Usage.

- class UserEvent
- class UUIDGenerator

  Class for generating unique UUID generate DCE 1.1 uid.

- 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 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.

- class VR16ExplicitDataElement

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

- struct VRToEncoding
- struct VRToType
- singleton VRVLSize
- class VRVLSize< 0 >
- class VRVLSize< 1 >
- class Waveform

  Waveform class.

- class Writer

  Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.

- 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<'\\', 64, 0 > UIComp
- 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 EQueryLevel {
  ePatient = 0,
  eStudy = 1,
  eSeries = 2,
  eImage = 3 }

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

Functions

- ignore_char const backslash ('\\')
- 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 Version &v)
- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)
- std::ostream & operator<< (std::ostream &os, const SwapCode &sc)
- std::ostream & operator<< (std::ostream &os, const FileSet &f)
- std::ostream & operator<< (std::ostream &os, const Region &r)
- std::ostream & operator<< (std::ostream &os, Event &e)

  Generic inserter operator for Event and its subclasses.
- std::ostream & operator<< (std::ostream &os, const PDBElement &val)
- std::ostream & operator<< (std::ostream &os, const CommandDataSet &val)
- std::ostream & operator<< (std::ostream &os, const Orientation &o)
- std::ostream & operator<< (std::ostream &_os, const IODs &_val)
- std::ostream & operator<< (std::ostream &_os, const Macros &_val)
- std::ostream & operator<< (std::ostream &_os, const Modules &_val)
- std::ostream & operator<< (std::ostream &_os, const Type &val)
- std::ostream & operator<< (std::ostream &_os, const ModuleEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const GroupDict &_val)
- std::ostream & operator<< (std::ostream &os, const PrivateTag &val)
- std::ostream & operator<< (std::ostream &_os, const IOD &_val)
- std::ostream & operator<< (std::ostream &os, const File &val)
- std::ostream & operator<< (std::ostream &_os, const Usage &val)
- std::ostream & operator<< (std::ostream &os, const Sorter &s)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDictEntry &val)
- std::ostream & operator<< (std::ostream &os, const Preamble &val)
- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)
- std::ostream & operator<< (std::ostream &_os, const Macro &_val)
- std::ostream & operator<< (std::ostream &os, const CSAHeaderDict &val)
- std::ostream & operator<< (std::ostream &os, const Dicts &d)
- std::ostream & operator<< (std::ostream &os, const PDBHeader &d)
- std::ostream & operator<< (std::ostream &os, const CodeString &str)
- std::ostream & operator<< (std::ostream &_os, const Module &_val)
- std::ostream & operator<< (std::ostream &os, const PhotometricInterpretation &val)
- std::ostream & operator<< (std::ostream &os, const Directory &d)

- std::ostream & operator<< (std::ostream &os, const Global &g)
- std::ostream & operator<< (std::ostream &os, const Object &obj)
- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)
- std::ostream & operator<< (std::ostream &os, const DictEntry &val)
- std::ostream & operator<< (std::ostream &os, const VL &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 TransferSyntax &ts)
- std::ostream & operator<< (std::ostream &os, const FileMetaInformation &val)
- std::ostream & operator<< (std::ostream &_os, const VM &_val)
- std::ostream & operator<< (std::ostream &os, const Scanner &s)
- std::ostream & operator<< (std::ostream &os, const Dict &val)
- std::ostream & operator<< (std::ostream &_os, const MediaStorage &ms)
- std::ostream & operator<< (std::ostream &_os, const VR &val)
- std::ostream & operator<< (std::ostream &os, const Fragment &val)
- std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)
- std::ostream & operator<< (std::ostream &_os, const UI &_val)
- std::ostream & operator<< (std::ostream &os, const DataElement &val)
- std::ostream & operator<< (std::ostream &_os, const Tag &_val)
- std::ostream & operator<< (std::ostream &os, const DataSet &val)
- std::ostream & operator<< (std::ostream &os, const Item &val)
- std::ostream & operator<< (std::ostream &os, const PrivateDict &val)
- std::ostream & operator<< (std::ostream &_os, const UIDs &uid)
- bool operator== (const CodeString &ref, const CodeString &cs)
- template<char TDelimiter, unsigned int TMaxLength, char TPadChar>
  std::istream & operator>> (std::istream &is, String< TDelimiter, TMaxLength, TPadChar > &ms)
- std::istream & operator>> (std::istream &in, ignore_char const &ic)
- std::istream & operator>> (std::istream &_is, Tag &_val)
- template<typename Float >
  std::string to_string (Float data)
- TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

Variables

- static Global GlobalInstance
- VRBINARY

26.1.1 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.

26.1.2 Typedef Documentation

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

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

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

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

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

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

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

26.1.2.8 typedef Bitmap gdcm::IconImage

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

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

26.1.2.11 typedef ModuleEntry gdcm::MacroEntry

26.1.2.12 typedef NestedModuleEntries gdcm::NestedMacroEntries

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

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

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

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

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

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

26.1.3 Enumeration Type Documentation

26.1.3.1 enum gdcm::CompOperators

Enumerator

GDCM_EQUAL

GDCM_DIFFERENT
GDCM_GREATER
GDCM_GREATEROREQUAL
GDCM_LESS
GDCM_LESSOREQUAL

26.1.3.2 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

26.1.3.3 enum gdcm::EQueryLevel

Enumerator

ePatient
eStudy
eSeries
eImage

26.1.3.4 enum gdcm::EQueryType

Enumerator

eFind
eMove

26.1.3.5 enum gdcm::ERootType

Enumerator

ePatientRootType

eStudyRootType

26.1.3.6 enum gdcm::LodModeType

Enumerator

LD_ALL

LD_NOSEQ

LD_NOSHADOW

LD_NOSHADOWSEQ

26.1.4 Function Documentation

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

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

26.1.4.2 VR::VRType gdcm::GetVRFromTag (Tag const & *tag*)

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

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

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

References gdcm::Version::Print().

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

References gdcm::ModuleEntry::DataElementType, gdcm::ModuleEntry::DescriptionField, and gdcm::ModuleEntry::Name.

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

References gdcm::SwapCode::GetSwapCodeString().

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

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

References gdcm::Region::Print().

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

Generic inserter operator for Event and its subclasses.

References gdcm::Event::Print().

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

References gdcm::PDBElement::NameField, and gdcm::PDBElement::ValueField.

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

References gdcm::DataSet::Print().

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

References gdcm::Orientation::Print().

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

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

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

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

References gdcm::Type::GetTypeString().

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

References gdcm::ModuleEntry::DataElementType, gdcm::ModuleEntry::DescriptionField, and gdcm::ModuleEntry::↩ Name.

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

References gdcm::GroupDict::GetAbbreviation(), gdcm::GroupDict::GetName(), and gdcm::GroupDict::Size().

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

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

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

References gdcm::File::GetHeader().

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

References gdcm::Usage::GetUsageString().

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

References gdcm::Sorter::Print().

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

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

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

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

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

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

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

References gdcm::PDBHeader::Print().

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

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

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

References gdcm::PhotometricInterpretation::GetPIString().

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

References gdcm::Directory::Print().

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

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

References gdcm::Object::Print().

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

References gdcm::DataElement::GetByteValue(), gdcm::DataElement::ValueField, and gdcm::DataElement::Value↩LengthField.

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

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

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

References gdcm::CSAElement::DataField, gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), gdcm::CSAElement::KeyField, gdcm::CSAElement::NameField, gdcm::CSAElement::NoOfItemsField, gdcm::CSAElement::SyngoDTField, gdcm::CSAElement::ValueMultiplicityField, gdcm::VM::VM1, and gdcm::CSAElement::VRField.

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

References gdcm::CSAHeader::Print().

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

References gdcm::TransferSyntax::GetTSString().

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

References gdcm::FileMetaInformation::GetPreamble(), and gdcm::DataSet::Print().

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

References gdcm::VM::GetVMString().

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

References gdcm::Scanner::Print().

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

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

References gdcm::MediaStorage::GetMSString().

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

References gdcm::VR::GetVRString().

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

References gdcm::DataElement::TagField, gdcm::DataElement::ValueField, and gdcm::DataElement::ValueLengthField.

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

References gdcm::PixelFormat::Print().

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

References gdcm::UI::Internal.

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

References gdcm::Object::Print(), gdcm::DataElement::TagField, gdcm::DataElement::ValueField, gdcm::Data↩Element::ValueLengthField, and gdcm::DataElement::VRField.

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

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

References gdcm::DataSet::Print().

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

References gdcm::DataSet::Print(), gdcm::DataElement::TagField, and gdcm::DataElement::ValueLengthField.

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

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

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

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

Examples:

DumpPhilipsECHO.cxx.

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

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

References gdcm::ignore_char::m_char.

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

References gdcm::Tag::SetElement(), and gdcm::Tag::SetGroup().

26.1.4.63 template<typename Float > std::string gdcm::to_string (Float data)

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

26.1.4.64 gdcm::TYPETOENCODING (SQ , VRBINARY , unsigned *char*)

26.1.5 Variable Documentation

26.1.5.1 Global gdcm::GlobalInstance [static]

26.1.5.2 gdcm::VRBINARY

26.2 gdcm::network Namespace Reference

Classes

- class AAbortPDU

  AAbortPDU Table 9-26 A-ABORT PDU FIELDS.
- class AAssociateACPDU

  AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.
- class AAssociateRJPDU

  AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.
- class AAssociateRQPDU

  AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.
- class AbstractSyntax

  AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.
- class ApplicationContext

  ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS.
- class AReleaseRPPDU

  AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.
- class AReleaseRQPDU

  AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.
- class ARTIMTimer

  ARTIMTimer This file contains the code for the ARTIM timer.
- class AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIE↩ LDS (A-ASSOCIATE-RQ)
- class BaseCompositeMessage

  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.
- class BasePDU

  BasePDU base class for PDUs.
- class CEchoRQ

  CEchoRQ this file defines the messages for the cecho action.
- 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 this file defines the messages for the cfind action.

- class CFindRSP

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

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

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

  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).
- class CStoreRQ

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

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

  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)
- class ImplementationClassUIDSub

  ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class ImplementationUIDSub

  ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)
- class ImplementationVersionNameSub

  ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class MaximumLengthSub

  MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class PDataTFPDU

  PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.
- class PDUFactory

  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.
- class PresentationContextAC

  PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.
- class PresentationContextRQ

  PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.
- class PresentationDataValue

  PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.
- class RoleSelectionSub

  RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)
- class ServiceClassApplicationInformation
- class SOPClassExtendedNegociationSub

  SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)
- class TableRow
- class TransferSyntaxSub

  TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

- struct Transition
- class ULAction

  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.
- 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 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.
- class ULConnection

  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.
- class ULConnectionCallback
- class ULConnectionInfo

  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.
- class ULConnectionManager

  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).
- class ULEvent

  ULEvent base class for network events.

- class ULTransitionTable

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

  UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

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

26.2.1 Enumeration Type Documentation

26.2.1.1 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

26.2.1.2 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

26.2.2 Function Documentation

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

References eSta10ReleaseCollisionAc, eSta11ReleaseCollisionRq, eSta12ReleaseCollisionAcLocal, eSta13Awaiting↩Close, eSta1Idle, eSta2Open, eSta3WaitLocalAssoc, eSta4LocalAssocDone, eSta5WaitRemoteAssoc, eSta6Transfer↩Ready, eSta7WaitRelease, eSta8WaitLocalRelease, eSta9ReleaseCollisionRqLocal, and eStaDoesNotExist.

26.2.3 Variable Documentation

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

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

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

26.3 gdcm::SegmentHelper Namespace Reference

Classes

- struct BasicCodedEntry

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

26.4 gdcm::terminal Namespace Reference

Class for Terminal Allow one to print in color in a shell.

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)

26.4.1 Detailed Description

Class for Terminal Allow one to print in color in a shell.

- support VT100 compatible shell
- win32 console

26.4.2 Enumeration Type Documentation

26.4.2.1 enum gdcm::terminal::Attribute

Enumerator

reset

bright

dim

underline

blink

reverse

hidden

26.4.2.2 enum gdcm::terminal::Color

Enumerator

black

red

green

yellow

blue

magenta

cyan

white

26.4.2.3 enum gdcm::terminal::Mode

Enumerator

CONSOLE

VT100

26.4.3 Function Documentation

26.4.3.1 GDCM_EXPORT std::string gdcm::terminal::setattribute (Attribute *att*)

26.4.3.2 GDCM_EXPORT std::string gdcm::terminal::setbgcolor (Color *c*)

26.4.3.3 GDCM_EXPORT std::string gdcm::terminal::setfgcolor (Color *c*)

26.4.3.4 GDCM_EXPORT void gdcm::terminal::setmode (Mode *m*)

Chapter 27

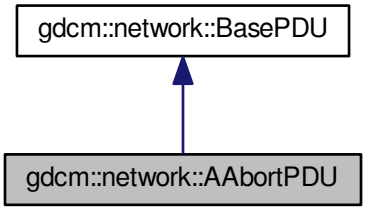
Class Documentation

27.1 gdcm::network::AAbortPDU Class Reference

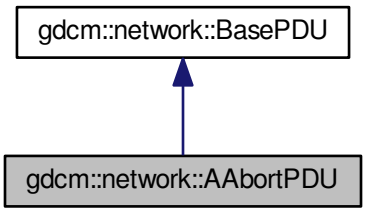
AAbortPDU Table 9-26 A-ABORT PDU FIELDS.

`#include <gdcmAAbortPDU.h>`

Inheritance diagram for gdcm::network::AAbortPDU:



Collaboration diagram for gdcm::network::AAbortPDU:



Public Member Functions

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

27.1.1 Detailed Description

AAbortPDU Table 9-26 A-ABORT PDU FIELDS.

27.1.2 Constructor & Destructor Documentation

27.1.2.1 gdcm::network::AAbortPDU::AAbortPDU ( )

27.1.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

27.1.3.2 void gdcm::network::AAbortPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

27.1.3.3 std::istream& gdcm::network::AAbortPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

27.1.3.4 void gdcm::network::AAbortPDU::SetReason (const uint8_t *r*)

27.1.3.5 void gdcm::network::AAbortPDU::SetSource (const uint8_t *s*)

27.1.3.6 size_t gdcm::network::AAbortPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

27.1.3.7 const std::ostream& gdcm::network::AAbortPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

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

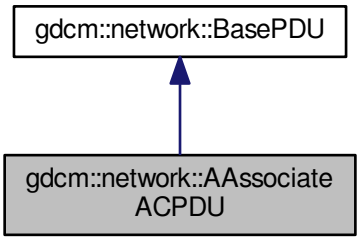
- gdcmAAbortPDU.h

27.2 gdcm::network::AAssociateACPDU Class Reference

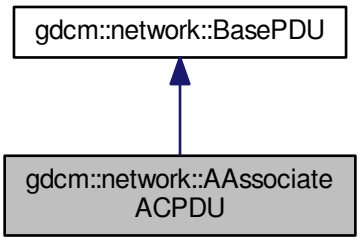
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

`#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
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- SizeType Size () const
- const std::ostream & Write (std::ostream &os) const

Protected Member Functions

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

Friends

- class AAssociateRQPDU

27.2.1 Detailed Description

AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields.

27.2.2 Member Typedef Documentation

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

27.2.3 Constructor & Destructor Documentation

27.2.3.1 gdcm::network::AAssociateACPDU::AAssociateACPDU ( )

27.2.4 Member Function Documentation

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

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

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

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

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

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

Implements gdcm::network::BasePDU.

27.2.4.7 void gdcm::network::AAssociateACPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

27.2.4.8 std::istream& gdcm::network::AAssociateACPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

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

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

27.2.4.11 SizeType gdcm::network::AAssociateACPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

27.2.4.12 const std::ostream& gdcm::network::AAssociateACPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

27.2.5 Friends And Related Function Documentation

27.2.5.1 friend class AAssociateRQPDU [friend]

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

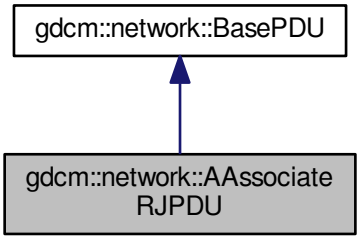
- gdcmAAssociateACPDU.h

27.3 gdcm::network::AAssociateRJPDU Class Reference

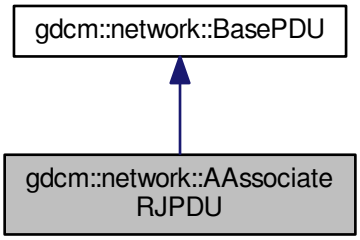
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

`#include <gdcmAAssociateRJPDU.h>`

Inheritance diagram for gdcm::network::AAssociateRJPDU:



Collaboration diagram for gdcm::network::AAssociateRJPDU:



Public Member Functions

- AAssociateRJPDU ()
- bool IsLastFragment () const

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

27.3.1 Detailed Description

AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

27.3.2 Constructor & Destructor Documentation

27.3.2.1 gdcm::network::AAssociateRJPDU::AAssociateRJPDU ( )

27.3.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

27.3.3.2 void gdcm::network::AAssociateRJPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

27.3.3.3 std::istream& gdcm::network::AAssociateRJPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

27.3.3.4 size_t gdcm::network::AAssociateRJPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

27.3.3.5 const std::ostream& gdcm::network::AAssociateRJPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

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

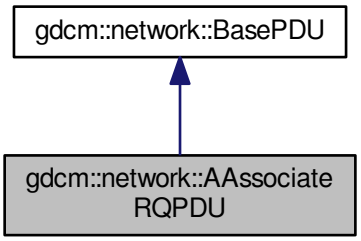
- gdcmAAssociateRJPDU.h

27.4 gdcm::network::AAssociateRQPDU Class Reference

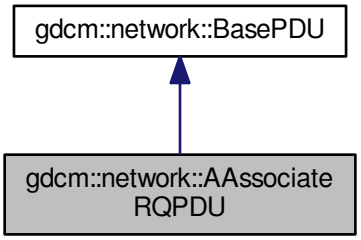
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

`#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 &as) const
- const PresentationContextRQ * GetPresentationContextByID (uint8_t i) const
- PresentationContextArrayType const & GetPresentationContexts ()
- const UserInformation & GetUserInformation () const
- bool IsLastFragment () const
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- 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
- const std::ostream & Write (std::ostream &os) const

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

27.4.1 Detailed Description

AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields.

27.4.2 Member Typedef Documentation

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

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

27.4.3 Constructor & Destructor Documentation

27.4.3.1 gdcm::network::AAssociateRQPDU::AAssociateRQPDU ( )

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

27.4.4 Member Function Documentation

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

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

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

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

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

27.4.4.6 const PresentationContextRQ* gdcm::network::AAssociateRQPDU::GetPresentationContextByAbstractSyntax (AbstractSyntax const & *as*) const

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

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

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

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

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

Check whether or not the.

Parameters

title	is a valid AE title

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

Implements gdcm::network::BasePDU.

27.4.4.13 void gdcm::network::AAssociateRQPDU::Print (std::ostream & *os*) const [virtual]

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

Implements gdcm::network::BasePDU.

27.4.4.14 std::istream& gdcm::network::AAssociateRQPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

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

Set the Called AE Title.

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

Set the Calling AE Title.

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

27.4.4.18 size_t gdcm::network::AAssociateRQPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

27.4.4.19 const std::ostream& gdcm::network::AAssociateRQPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

27.4.5 Friends And Related Function Documentation

27.4.5.1 friend class AAssociateACPDU [friend]

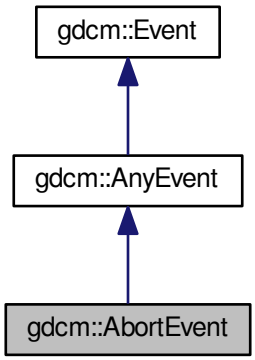
The documentation for this class was generated from the following file:

- gdcmAAssociateRQPDU.h

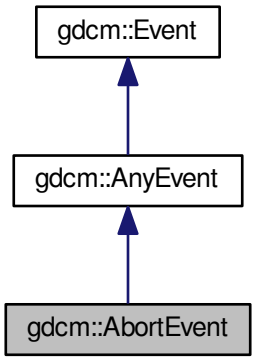
27.5 gdcm::AbortEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::AbortEvent:



Collaboration diagram for gdcm::AbortEvent:



Additional Inherited Members

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

- gdcmEvent.h

27.6 gdcm::network::AbstractSyntax Class Reference

AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

`#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

27.6.1 Detailed Description

AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

27.6.2 Constructor & Destructor Documentation

27.6.2.1 gdcm::network::AbstractSyntax::AbstractSyntax ( )

27.6.3 Member Function Documentation

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

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

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

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

27.6.3.5 std::istream& gdcm::network::AbstractSyntax::Read (std::istream & *is*)

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

27.6.3.7 void gdcm::network::AbstractSyntax::SetNameFromUID (UIDs::TSName *tsname*)

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

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

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

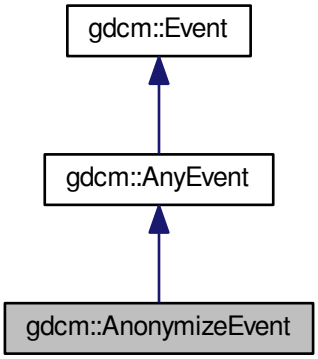
- gdcmAbstractSyntax.h

27.7 gdcm::AnonymizeEvent Class Reference

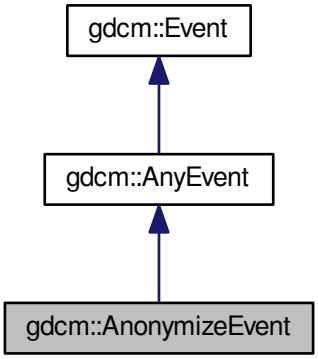
AnonymizeEvent Special type of event triggered during the Anonymization process.

`#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 (Tag const &tag=0)
- AnonymizeEvent (const Self &s)

- virtual ~AnonymizeEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- virtual const char * GetEventName () const
- Tag const & GetTag () const
- virtual ::gdcm::Event * MakeObject () const
- void SetTag (const Tag &t)

27.7.1 Detailed Description

AnonymizeEvent Special type of event triggered during the Anonymization process.

See also

Anonymizer

27.7.2 Member Typedef Documentation

27.7.2.1 typedef AnonymizeEvent gdcm::AnonymizeEvent::Self

27.7.2.2 typedef AnyEvent gdcm::AnonymizeEvent::Superclass

27.7.3 Constructor & Destructor Documentation

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

27.7.3.2 virtual gdcm::AnonymizeEvent::~AnonymizeEvent ( ) [inline], [virtual]

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

27.7.4 Member Function Documentation

27.7.4.1 virtual bool gdcm::AnonymizeEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

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

Return the StringName associated with the event.

Implements gdcm::Event.

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

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

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

Implements gdcm::Event.

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

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

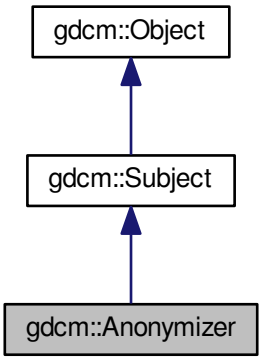
- gdcmAnonymizeEvent.h

27.8 gdcm::Anonymizer Class Reference

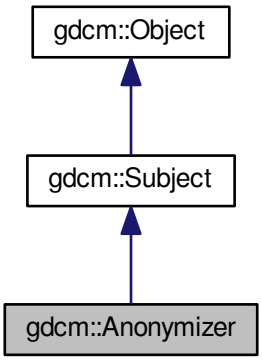
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:

`#include <gdcmAnonymizer.h>`

Inheritance diagram for gdcm::Anonymizer:



Collaboration diagram for gdcm::Anonymizer:



Public Member Functions

- Anonymizer ()
- ~Anonymizer ()

- bool BasicApplicationLevelConfidentialityProfile (bool deidentify=true)
- bool Empty (Tag const &t)
- const CryptographicMessageSyntax * GetCryptographicMessageSyntax () const
- File & GetFile ()
- bool Remove (Tag const &t)
- 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 (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.

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)

27.8.1 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, and EncapsulateFileInRawData.cxx.

27.8.2 Constructor & Destructor Documentation

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

27.8.2.2 gdcm::Anonymizer::∼Anonymizer ( )

27.8.3 Member Function Documentation

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

27.8.3.2 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

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

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

Clear the internal mapping of real UIDs to generated UIDs

Warning

the mapping is definitely lost

27.8.3.5 bool gdcm::Anonymizer::Empty (Tag const & *t*)

Make Tag t empty (if not found tag will be created) Warning: does not handle SQ element

Examples:

CreateJPIPDataSet.cxx.

27.8.3.6 static 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.

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

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

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

for wrapped language: instantiate a reference counted object

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

27.8.3.11 bool gdcm::Anonymizer::Remove (Tag const & *t*)

remove a tag (even a SQ can be removed) Return code is false when tag t cannot be found

27.8.3.12 bool gdcm::Anonymizer::RemoveGroupLength ( )

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

Examples:

ClinicalTrialAnnotate.cxx.

27.8.3.13 bool gdcm::Anonymizer::RemovePrivateTags ( )

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

Examples:

ClinicalTrialAnnotate.cxx.

27.8.3.14 bool gdcm::Anonymizer::RemoveRetired ( )

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

27.8.3.15 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, and EncapsulateFileInRawData.cxx.

27.8.3.16 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

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

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

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

Set/Get File.

Examples:

ClinicalTrialAnnotate.cxx, CreateJPIPDataSet.cxx, and EncapsulateFileInRawData.cxx.

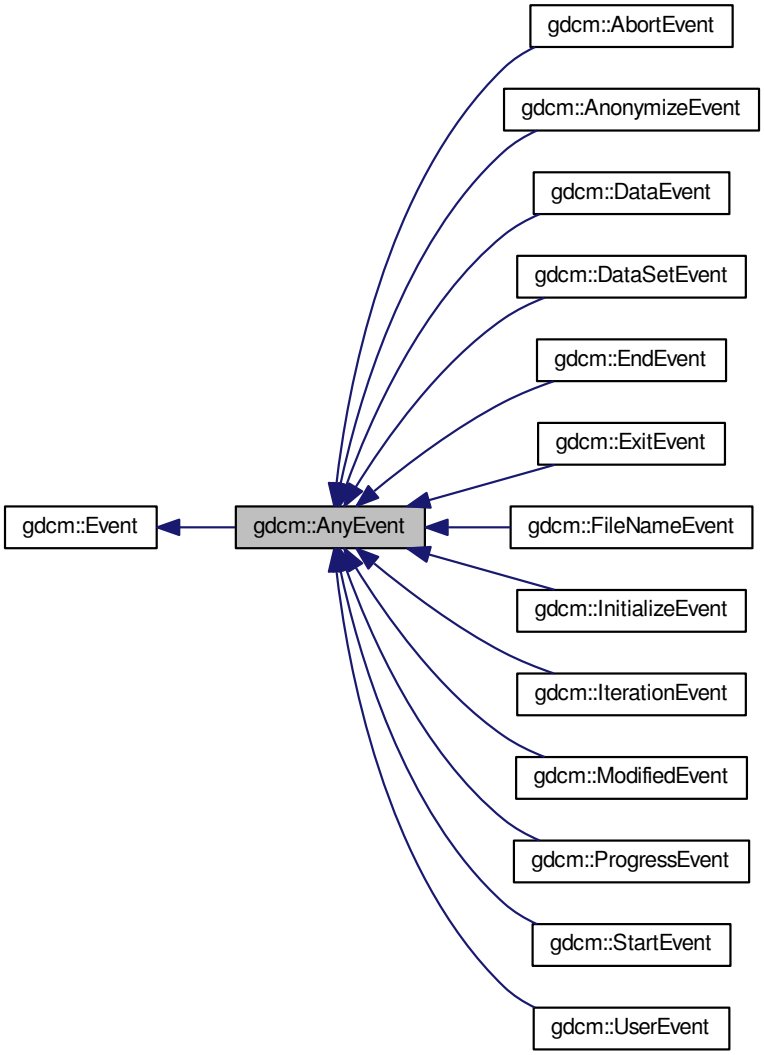
The documentation for this class was generated from the following file:

- gdcmAnonymizer.h

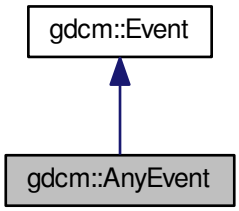
27.9 gdcm::AnyEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::AnyEvent:



Collaboration diagram for gdcm::AnyEvent:



Additional Inherited Members

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

- gdcmEvent.h

27.10 gdcm::network::ApplicationContext Class Reference

ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS.

`#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

27.10.1 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)

27.10.2 Constructor & Destructor Documentation

27.10.2.1 gdcm::network::ApplicationContext::ApplicationContext ( )

27.10.3 Member Function Documentation

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

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

27.10.3.3 std::istream& gdcm::network::ApplicationContext::Read (std::istream & *is*)

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

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

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

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

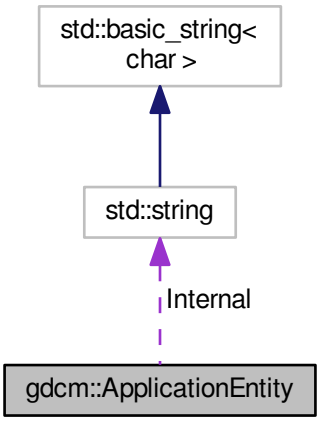
- gdcmApplicationContext.h

27.11 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 = ' '

27.11.1 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

27.11.2 Member Function Documentation

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

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

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

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

27.11.3 Member Data Documentation

27.11.3.1 std::string gdcm::ApplicationEntity::Internal

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

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

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

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

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

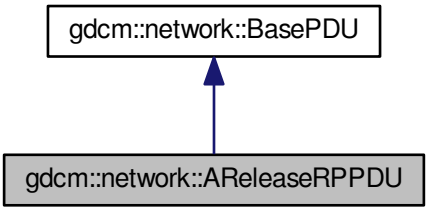
- gdcmApplicationEntity.h

27.12 gdcm::network::AReleaseRPPDU Class Reference

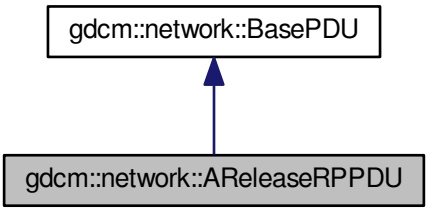
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

`#include <gdcmAReleaseRPPDU.h>`

Inheritance diagram for gdcm::network::AReleaseRPPDU:



Collaboration diagram for gdcm::network::AReleaseRPPDU:



Public Member Functions

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

27.12.1 Detailed Description

AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

27.12.2 Constructor & Destructor Documentation

27.12.2.1 gdcm::network::AReleaseRPPDU::AReleaseRPPDU ( )

27.12.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

27.12.3.2 void gdcm::network::AReleaseRPPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

27.12.3.3 std::istream& gdcm::network::AReleaseRPPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

27.12.3.4 size_t gdcm::network::AReleaseRPPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

27.12.3.5 const std::ostream& gdcm::network::AReleaseRPPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

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

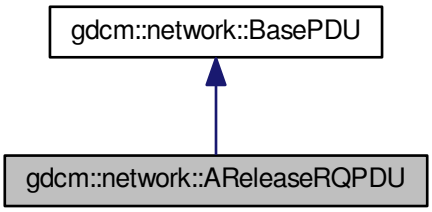
- gdcmAReleaseRPPDU.h

27.13 gdcm::network::AReleaseRQPDU Class Reference

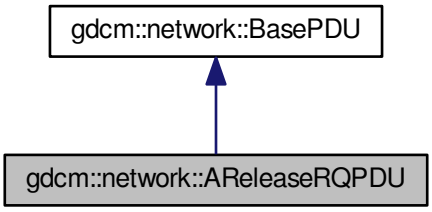
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

`#include <gdcmAReleaseRQPDU.h>`

Inheritance diagram for gdcm::network::AReleaseRQPDU:



Collaboration diagram for gdcm::network::AReleaseRQPDU:



Public Member Functions

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

27.13.1 Detailed Description

AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

27.13.2 Constructor & Destructor Documentation

27.13.2.1 gdcm::network::AReleaseRQPDU::AReleaseRQPDU ( )

27.13.3 Member Function Documentation

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

Implements gdcm::network::BasePDU.

27.13.3.2 void gdcm::network::AReleaseRQPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

27.13.3.3 std::istream& gdcm::network::AReleaseRQPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

27.13.3.4 size_t gdcm::network::AReleaseRQPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

27.13.3.5 const std::ostream& gdcm::network::AReleaseRQPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

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

- gdcmAReleaseRQPDU.h

27.14 gdcm::network::ARTIMTimer Class Reference

ARTIMTimer This file contains the code for the ARTIM timer.

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

Public Member Functions

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

27.14.1 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.

27.14.2 Constructor & Destructor Documentation

27.14.2.1 gdcm::network::ARTIMTimer::ARTIMTimer ( )

27.14.3 Member Function Documentation

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

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

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

27.14.3.4 void gdcm::network::ARTIMTimer::SetTimeout (double *inTimeout*)

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

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

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

- gdcmARTIMTimer.h

27.15 gdcm::ASN1 Class Reference

Class for ASN1.

`#include <gdcmASN1.h>`

Public Member Functions

- ASN1 ()
- ~ASN1 ()

Static Public Member Functions

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

Protected Member Functions

- int TestPBKDF2 ()

27.15.1 Detailed Description

Class for ASN1.

27.15.2 Constructor & Destructor Documentation

27.15.2.1 gdcm::ASN1::ASN1 ( )

27.15.2.2 gdcm::ASN1::~ASN1 ( )

27.15.3 Member Function Documentation

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

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

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

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

- gdcmASN1.h

27.16 gdcm::network::AsynchronousOperationsWindowSub Class Reference

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

`#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

27.16.1 Detailed Description

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

27.16.2 Constructor & Destructor Documentation

27.16.2.1 gdcm::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ( )

27.16.3 Member Function Documentation

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

27.16.3.2 std::istream& gdcm::network::AsynchronousOperationsWindowSub::Read (std::istream & *is*)

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

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

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

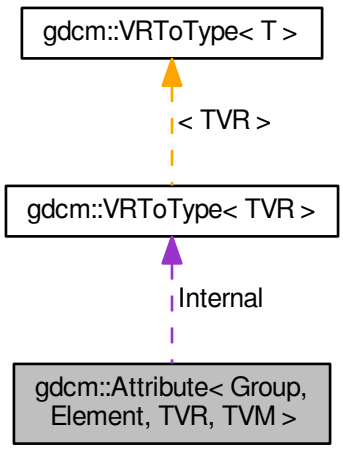
- gdcmAsynchronousOperationsWindowSub.h

27.17 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 instanciation does not match the public dictionary.

`#include <gdcmAttribute.h>`

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::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)))
- 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)

27.17.1 Detailed Description

template<uint16_t Group, uint16_t Element, int 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 instanciation 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, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenFakeIdentifyFile.cxx, GetSequenceUltrasound.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndPrintAttributes.cxx, rle2img.cxx, SortImage.cxx, StreamImageReaderTest.cxx, and VolumeSorter.cxx.

27.17.2 Member Typedef Documentation

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

27.17.3 Member Enumeration Documentation

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

Enumerator

VMType

27.17.4 Member Function Documentation

27.17.4.1 template<uint16_t Group, uint16_t Element, int 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)))

27.17.4.2 template<uint16_t Group, uint16_t Element, int 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)))

27.17.4.3 template<uint16_t Group, uint16_t Element, int 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)))

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

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VR_ASCII.

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

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

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

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue(), gdcm::Attribute< Group, Element, TVR, TVM >::operator!=(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=(), gdcm::Attribute< Group, Element, TVR, TVM >::operator<(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator<(), gdcm::Attribute< Group, Element, TVR, TVM >::operator==(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::operator==(), gdcm::Attribute< Group, Element, TVR, TVM >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, TVM >::SetValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetValues(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues().

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

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), 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 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

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

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues().

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::operator[](), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[]().

27.17.4.10 template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues().

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

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

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

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

Referenced by gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print().

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

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

27.17.4.14 template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

27.17.4.15 template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

27.17.4.16 template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

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

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValue().

27.17.4.18 template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetValue().

27.17.4.19 template<uint16_t Group, uint16_t Element, int 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 gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetTag().

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

References gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

27.17.4.21 template<uint16_t Group, uint16_t Element, int 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(), gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::ByteValue::GetPointer().

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

27.17.4.22 template<uint16_t Group, uint16_t Element, int 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(), gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::ByteValue::GetPointer().

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement().

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

References gdcm::DataElement::GetByteValue(), gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::VR::INVALID, gdcm::DataElement::IsEmpty(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), and gdcm::VR::UN.

Referenced by 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 >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

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

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::IsEmpty(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

27.17.4.25 template<uint16_t Group, uint16_t Element, int 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]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues().

27.17.4.26 template<uint16_t Group, uint16_t Element, int 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:

LargeVRDSExplicit.cxx.

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues().

Referenced by gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues().

27.17.5 Member Data Documentation

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

Referenced by gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::~Attribute().

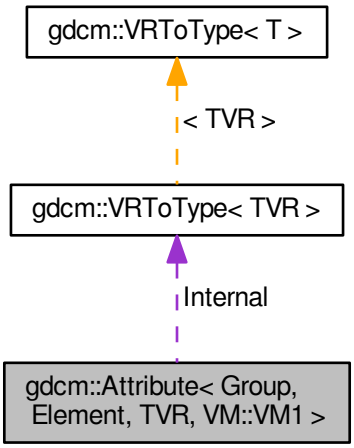
The documentation for this class was generated from the following file:

- gdcmAttribute.h

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

`#include <gdcmAttribute.h>`

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



Public Types

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

Public Member Functions

- GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length==1)
- GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))
- GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VM↩ Type)))
- GDCM_STATIC_ASSERT (((((VR::VRType) TVR &VR::VR_VM1)&&((VM::VMType) VM::VM1==VM::V↩ M1))||!((VR::VRType) TVR &VR::VR_VM1)))
- 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
- 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)

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)

27.18.1 Member Typedef Documentation

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

27.18.2 Member Enumeration Documentation

27.18.2.1 template<uint16_t Group, uint16_t Element, int TVR> anonymous enum

Enumerator

VMType

27.18.3 Member Function Documentation

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

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

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

27.18.3.4 template<uint16_t Group, uint16_t Element, int 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)))

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

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VR↩ ASCII.

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

27.18.3.7 template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR ( ) [inline], [static]

27.18.3.8 template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues ( ) const [inline]

27.18.3.9 template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag ( ) [inline], [static]

27.18.3.10 template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

27.18.3.11 template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

27.18.3.12 template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

27.18.3.13 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVM ( ) [inline], [static]

References gdcm::VM::VM1.

27.18.3.14 template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR ( ) [inline], [static]

27.18.3.15 template<uint16_t Group, uint16_t Element, int 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(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

27.18.3.16 template<uint16_t Group, uint16_t Element, int 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(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

27.18.3.17 template<uint16_t Group, uint16_t Element, int 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(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetValues().

27.18.3.18 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Print (std::ostream & *os*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

27.18.3.19 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Set (DataSet const & *ds*) [inline]

References gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

27.18.3.20 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue (const ByteValue * *bv*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::ByteValue::GetPointer().

27.18.3.21 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap (const ByteValue * *bv*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::ByteValue::GetPointer().

27.18.3.22 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement (DataElement const & *de*) [inline]

References gdcm::DataElement::GetByteValue(), gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::VR::INVALID, gdcm::DataElement::IsEmpty(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), and gdcm::VR::UN.

27.18.3.23 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet (DataSet const & *ds*) [inline]

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::IsEmpty(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

27.18.3.24 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (ArrayType *v*) [inline]

27.18.4 Member Data Documentation

27.18.4.1 template<uint16_t Group, uint16_t Element, int TVR> ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal

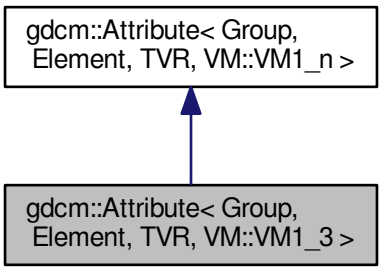
The documentation for this class was generated from the following file:

- gdcmAttribute.h

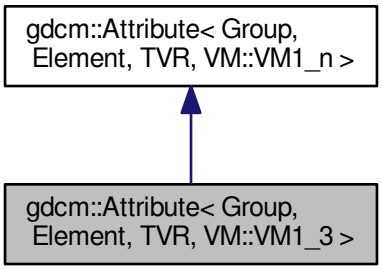
27.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

`#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 Member Functions

- VM GetVM () const

Additional Inherited Members

27.19.1 Member Function Documentation

27.19.1.1 template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >::GetVM ( ) const [inline]

References gdcm::VM::VM1_3.

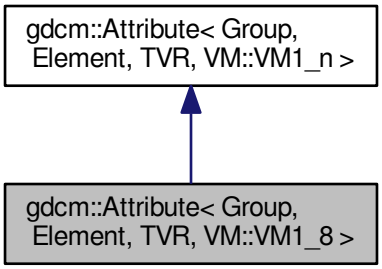
The documentation for this class was generated from the following file:

- gdcmAttribute.h

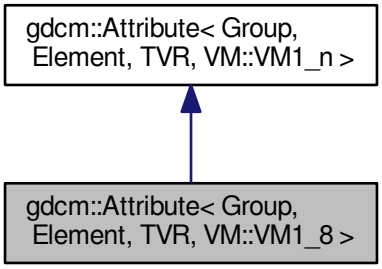
27.20 gdcm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference

```
#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 Member Functions

- VM GetVM () const

Additional Inherited Members

27.20.1 Member Function Documentation

27.20.1.1 template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM ( ) const [inline]

References gdcm::VM::VM1_8.

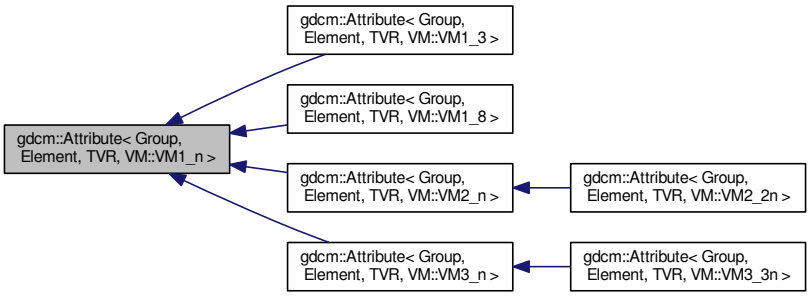
The documentation for this class was generated from the following file:

- gdcmAttribute.h

27.21 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 >:



Public Types

- typedef VRToType< TVR >::Type ArrayType

Public Member Functions

- Attribute ()
- ~Attribute ()
- 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)))
- 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
- 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 (unsigned int idx, ArrayType v)
- void SetValue (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)

27.21.1 Member Typedef Documentation

27.21.1.1 template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::ArrayType

27.21.2 Constructor & Destructor Documentation

27.21.2.1 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute ( ) [inline], [explicit]

27.21.2.2 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::~Attribute ( ) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

27.21.3 Member Function Documentation

27.21.3.1 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT (((VR::VRType) TVR &(VR::VRType)(TagToType< Group, Element >::VRType)))

27.21.3.2 template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GDCM_STATIC_ASSERT ((VM::VM1_n &(VM::VMType)(TagToType< Group, Element >::VMType)))

27.21.3.3 template<uint16_t Group, uint16_t Element, int 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))||!((VR::VRType) TVR &VR::VR_VM1)))

27.21.3.4 template<uint16_t Group, uint16_t Element, int TVR> DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VR↩ ASCII.

27.21.3.5 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM ( ) [inline], [static]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetVM().

27.21.3.6 template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR ( ) [inline], [static]

27.21.3.7 template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues ( ) const [inline]

27.21.3.8 template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag ( ) [inline], [static]

27.21.3.9 template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues().

27.21.3.10 template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues().

27.21.3.11 template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues ( ) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

27.21.3.12 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM ( ) [inline], [static]

References gdcm::VM::VM1_n.

27.21.3.13 template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR ( ) [inline], [static]

27.21.3.14 template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (unsigned int *idx*) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValue().

27.21.3.15 template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (unsigned int *idx*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetValue().

27.21.3.16 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Print (std::ostream & *os*) const [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVM(), and gdcm::Attribute< Group, Element, TVR, TVM >::GetVR().

27.21.3.17 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Set (DataSet const & *ds*) [inline]

References gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

27.21.3.18 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue (const ByteValue * *bv*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetValues().

27.21.3.19 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement (DataElement const & *de*) [inline]

References gdcm::DataElement::GetByteValue(), gdcm::Tag::GetGroup(), gdcm::DataElement::GetTag(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::GetVR(), gdcm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcm::DataElement::IsEmpty(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue().

27.21.3.20 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet (DataSet const & *ds*) [inline]

References gdcm::DataSet::FindDataElement(), gdcm::DataSet::GetDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcm::DataElement::IsEmpty(), and gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

27.21.3.21 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues (unsigned int *numel*) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::SetValues().

27.21.3.22 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (unsigned int *idx,* ArrayType *v*) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues().

27.21.3.23 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue (ArrayType *v*) [inline]

References SetValue().

Referenced by SetValue().

27.21.3.24 template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues (const ArrayType * *array,* unsigned int *numel,* bool *own* = false) [inline]

References gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

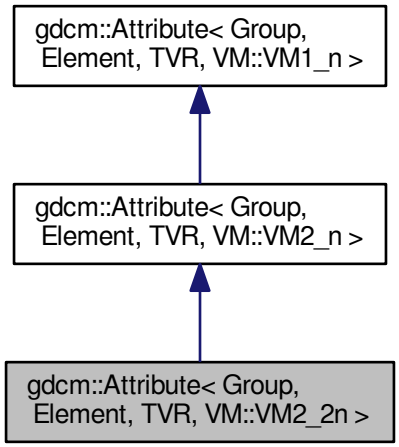
The documentation for this class was generated from the following file:

- gdcmAttribute.h

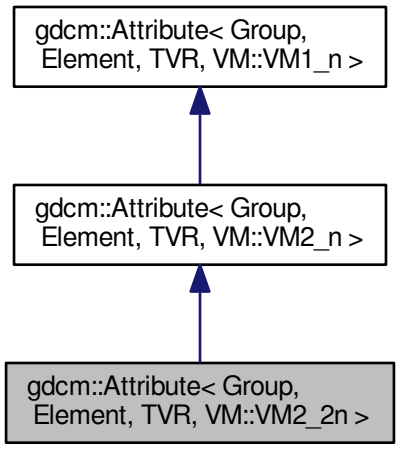
27.22 gdcm::Attribute< Group, Element, TVR, VM::VM2_2n > Class Template Reference

`#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 >:



Static Public Member Functions

- static VM GetVM ()

Additional Inherited Members

27.22.1 Member Function Documentation

27.22.1.1 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM ( ) [inline], [static]

References gdcm::VM::VM2_2n.

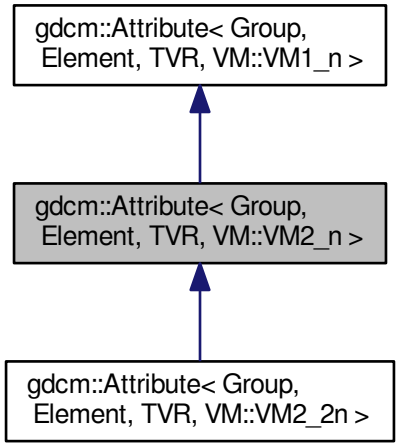
The documentation for this class was generated from the following file:

- gdcmAttribute.h

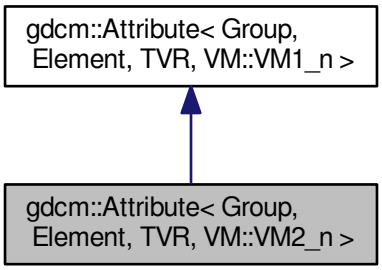
27.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

`#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 Member Functions

- VM GetVM () const

Additional Inherited Members

27.23.1 Member Function Documentation

27.23.1.1 template<uint16_t Group, uint16_t Element, int TVR> VM gdcm::Attribute< Group, Element, TVR, VM::VM2_n >::GetVM ( ) const [inline]

References gdcm::VM::VM2_n.

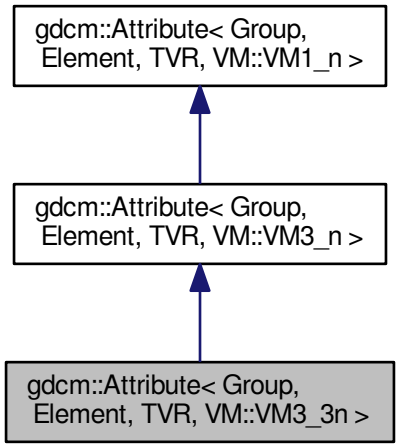
The documentation for this class was generated from the following file:

- gdcmAttribute.h

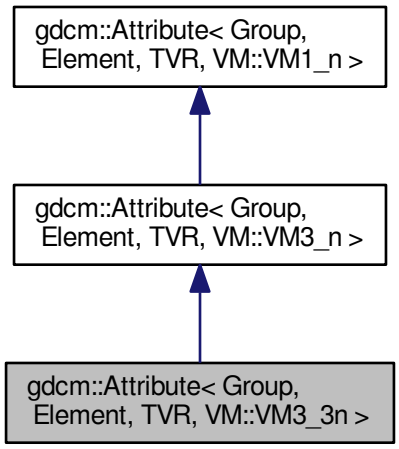
27.24 gdcm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

`#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 >:



Static Public Member Functions

- static VM GetVM ()

Additional Inherited Members

27.24.1 Member Function Documentation

27.24.1.1 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM ( ) [inline], [static]

References gdcm::VM::VM3_3n.

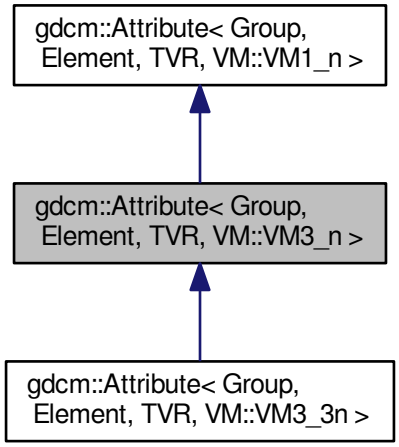
The documentation for this class was generated from the following file:

- gdcmAttribute.h

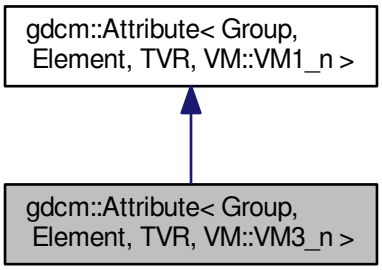
27.25 gdcm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference

```
#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 >:



Static Public Member Functions

- static VM GetVM ()

Additional Inherited Members

27.25.1 Member Function Documentation

27.25.1.1 template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM3_n >::GetVM ( ) [inline], [static]

References gdcm::VM::VM3_n.

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

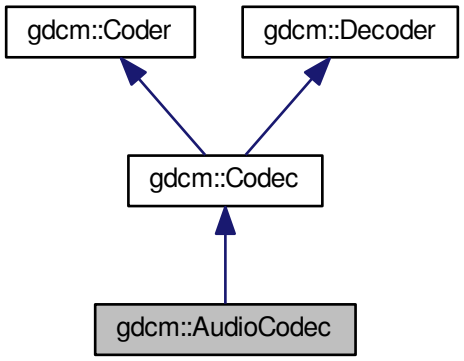
- gdcmAttribute.h

27.26 gdcm::AudioCodec Class Reference

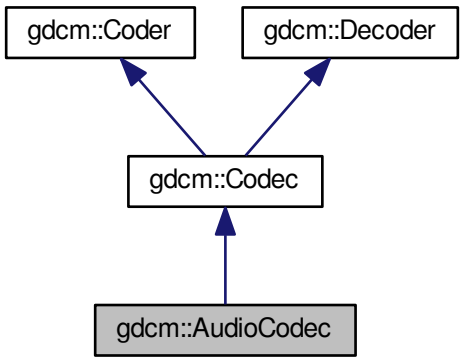
AudioCodec.

`#include <gdcmAudioCodec.h>`

Inheritance diagram for gdcm::AudioCodec:



Collaboration diagram for gdcm::AudioCodec:



Public Member Functions

- AudioCodec ()
- ~AudioCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Additional Inherited Members

27.26.1 Detailed Description

AudioCodec.

27.26.2 Constructor & Destructor Documentation

27.26.2.1 gdcm::AudioCodec::AudioCodec ( )

27.26.2.2 gdcm::AudioCodec::~AudioCodec ( )

27.26.3 Member Function Documentation

27.26.3.1 bool gdcm::AudioCodec::CanCode (TransferSyntax const &) const [inline], [virtual]

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

27.26.3.2 bool gdcm::AudioCodec::CanDecode (TransferSyntax const &) const [inline], [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

27.26.3.3 bool gdcm::AudioCodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

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

- gdcmAudioCodec.h

27.27 gdcm::Base64 Class Reference

Class for Base64.

`#include <gdcmBase64.h>`

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)

27.27.1 Detailed Description

Class for Base64.

27.27.2 Member Function Documentation

27.27.2.1 static 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.

27.27.2.2 static 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

27.27.2.3 static size_t gdcm::Base64::GetDecodeLength (const char * *src*, size_t *len*) [static]

Call this function to obtain the required buffer size

Examples:

DumpExamCard.cxx.

27.27.2.4 static size_t gdcm::Base64::GetEncodeLength (const char * *src*, size_t *srclen*) [static]

Call this function to obtain the required buffer size

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

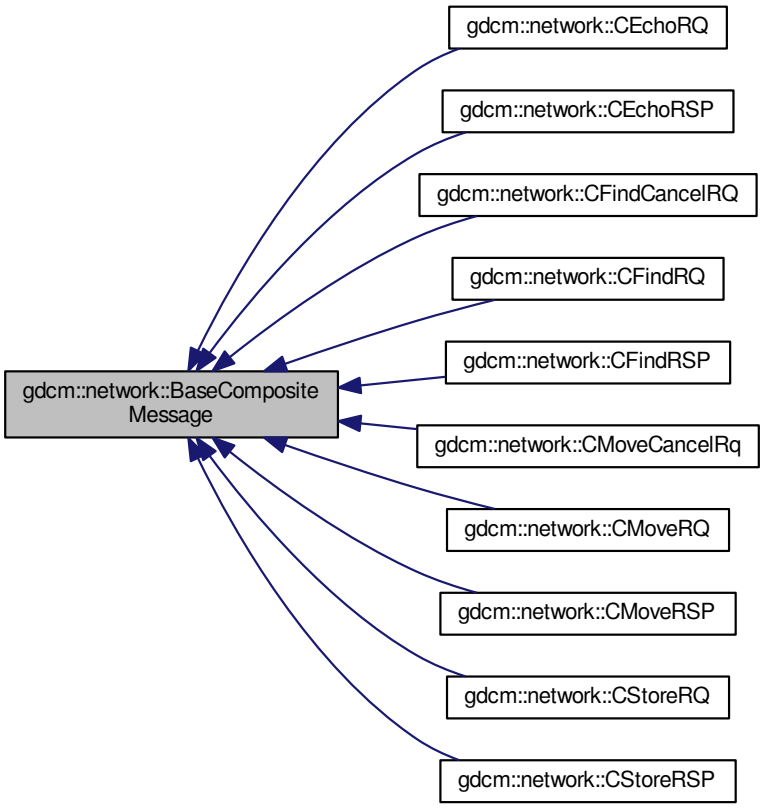
- gdcmBase64.h

27.28 gdcm::network::BaseCompositeMessage Class Reference

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.

```
#include <gdcmBaseCompositeMessage.h>
```

Inheritance diagram for gdcm::network::BaseCompositeMessage:



Public Member Functions

- virtual ~BaseCompositeMessage ()
- virtual std::vector < PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0

27.28.1 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.

27.28.2 Constructor & Destructor Documentation

27.28.2.1 virtual gdcm::network::BaseCompositeMessage::~BaseCompositeMessage ( ) [inline], [virtual]

27.28.3 Member Function Documentation

27.28.3.1 virtual std::vector<PresentationDataValue> gdcm::network::BaseCompositeMessage::ConstructPDV (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [pure virtual]

Implemented in gdcm::network::CMoveRQ, gdcm::network::CFindRQ, and gdcm::network::CEchoRQ.

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

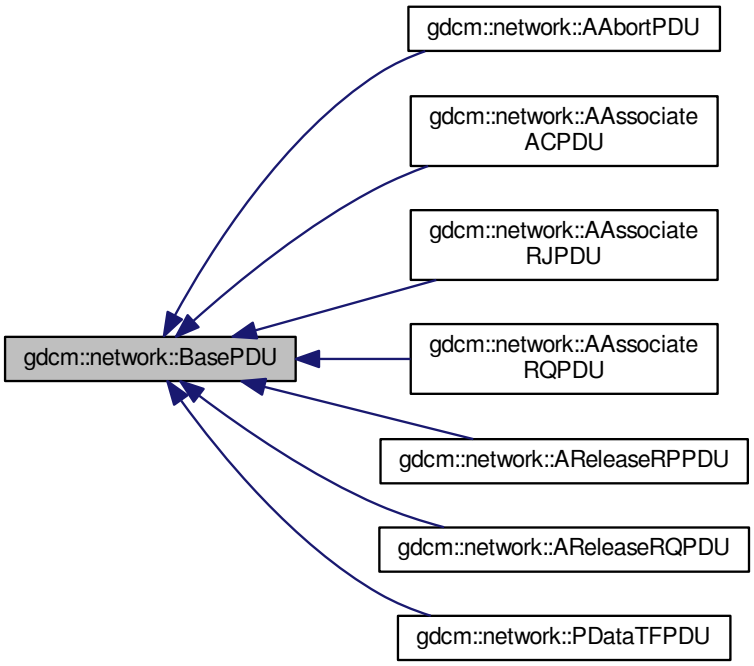
- gdcmBaseCompositeMessage.h

27.29 gdcm::network::BasePDU Class Reference

BasePDU base class for PDUs.

`#include <gdcmBasePDU.h>`

Inheritance diagram for gdcm::network::BasePDU:



Public Member Functions

- virtual ~BasePDU ()
- 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

27.29.1 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

27.29.2 Constructor & Destructor Documentation

27.29.2.1 virtual gdcm::network::BasePDU::~BasePDU ( ) [inline], [virtual]

27.29.3 Member Function Documentation

27.29.3.1 virtual bool gdcm::network::BasePDU::IsLastFragment ( ) const [pure virtual]

Implemented in gdcm::network::AAssociateRQPDU, gdcm::network::AAssociateACPDU, gdcm::network::PDataTFPDU, gdcm::network::AAbortPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, and gdcm::network::AReleaseRQPDU.

27.29.3.2 virtual void gdcm::network::BasePDU::Print (std::ostream & *os*) const [pure virtual]

Implemented in gdcm::network::AAssociateRQPDU, gdcm::network::AAssociateACPDU, gdcm::network::PDataTFPDU, gdcm::network::AAbortPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAssociateRJPDU.

27.29.3.3 virtual std::istream& gdcm::network::BasePDU::Read (std::istream & *is*) [pure virtual]

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTFPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAbortPDU.

27.29.3.4 virtual size_t gdcm::network::BasePDU::Size ( ) const [pure virtual]

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTFPDU, gdcm::network::AAbortPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, and gdcm::network::AReleaseRQPDU.

27.29.3.5 virtual const std::ostream& gdcm::network::BasePDU::Write (std::ostream & *os*) const [pure virtual]

Implemented in gdcm::network::AAssociateACPDU, gdcm::network::AAssociateRQPDU, gdcm::network::PDataTFPDU, gdcm::network::AAssociateRJPDU, gdcm::network::AReleaseRPPDU, gdcm::network::AReleaseRQPDU, and gdcm::network::AAbortPDU.

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

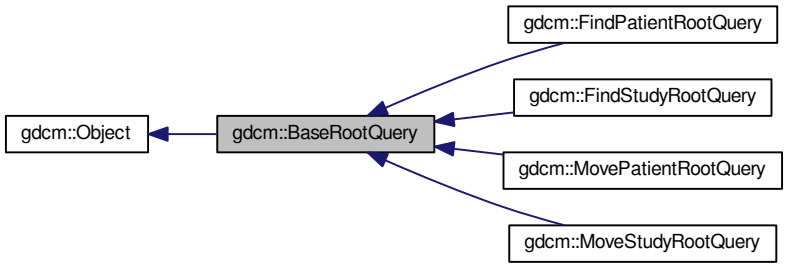
- gdcmBasePDU.h

27.30 gdcm::BaseRootQuery Class Reference

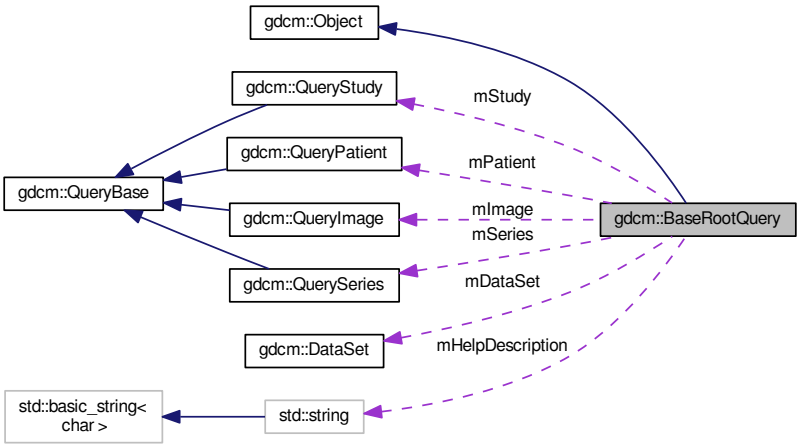
BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.

`#include <gdcmBaseRootQuery.h>`

Inheritance diagram for gdcm::BaseRootQuery:



Collaboration diagram for gdcm::BaseRootQuery:



Public Member Functions

- virtual ~BaseRootQuery ()
- void AddQueryDataSet (const DataSet &ds)
- virtual UIDs::TSName GetAbstractSyntaxUID () const =0
- DataSet const & GetQueryDataSet () const

  Set/Get the internal representation of the query as a DataSet.
- DataSet & GetQueryDataSet ()
- EQueryLevel GetQueryLevelFromQueryRoot (ERootType roottype)
- virtual std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)=0
- virtual void InitializeDataSet (const EQueryLevel &inQueryLevel)=0
- void Print (std::ostream &os) const
- void SetSearchParameter (const Tag &inTag, const std::string &inValue)

- void SetSearchParameter (const std::string &inKeyword, const std::string &inValue)
- virtual bool ValidateQuery (bool inStrict=true) const =0
- const std::ostream & WriteHelpFile (std::ostream &os)
- bool WriteQuery (const std::string &inFileName)

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 ()
- void SetSearchParameter (const Tag &inTag, const DictEntry &inDictEntry, const std::string &inValue)

Protected Attributes

- DataSet mDataSet
- std::string mHelpDescription
- QueryImage mImage
- QueryPatient mPatient
- ERootType mRootType
- QuerySeries mSeries
- QueryStudy mStudy

Friends

- class QueryFactory

27.30.1 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.

27.30.2 Constructor & Destructor Documentation

27.30.2.1 gdcm::BaseRootQuery::BaseRootQuery ( ) [protected]

27.30.2.2 virtual gdcm::BaseRootQuery::~BaseRootQuery ( ) [virtual]

27.30.3 Member Function Documentation

27.30.3.1 void gdcm::BaseRootQuery::AddQueryDataSet (const DataSet & *ds*)

27.30.3.2 static QueryBase* gdcm::BaseRootQuery::Construct (ERootType *inRootType,* EQueryLevel *qlevel*) [static]

27.30.3.3 virtual UIDs::TSName gdcm::BaseRootQuery::GetAbstractSyntaxUID ( ) const [pure virtual]

Implemented in gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::FindPatientRootQuery.

27.30.3.4 DataSet const& gdcm::BaseRootQuery::GetQueryDataSet ( ) const

Set/Get the internal representation of the query as a DataSet.

27.30.3.5 DataSet& gdcm::BaseRootQuery::GetQueryDataSet ( )

27.30.3.6 EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (ERootType *roottype*)

27.30.3.7 static int gdcm::BaseRootQuery::GetQueryLevelFromString (const char * *str*) [static]

27.30.3.8 static const char* gdcm::BaseRootQuery::GetQueryLevelString (EQueryLevel *ql*) [static]

27.30.3.9 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, and gdcm::MoveStudyRootQuery.

27.30.3.10 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, and gdcm::MoveStudyRootQuery.

27.30.3.11 void gdcm::BaseRootQuery::Print (std::ostream & *os*) const [virtual]

Reimplemented from gdcm::Object.

27.30.3.12 void gdcm::BaseRootQuery::SetSearchParameter (const Tag & *inTag,* const DictEntry & *inDictEntry,* const std::string & *inValue*) [protected]

27.30.3.13 void gdcm::BaseRootQuery::SetSearchParameter (const Tag & *inTag,* const std::string & *inValue*)

27.30.3.14 void gdcm::BaseRootQuery::SetSearchParameter (const std::string & *inKeyword,* const std::string & *inValue*)

27.30.3.15 virtual bool gdcm::BaseRootQuery::ValidateQuery (bool *inStrict* = `true`) const [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.

Implemented in gdcm::FindStudyRootQuery, gdcm::MovePatientRootQuery, gdcm::MoveStudyRootQuery, and gdcm::FindPatientRootQuery.

27.30.3.16 const std::ostream& gdcm::BaseRootQuery::WriteHelpFile (std::ostream & *os*)

27.30.3.17 bool gdcm::BaseRootQuery::WriteQuery (const std::string & *inFileName*)

27.30.4 Friends And Related Function Documentation

27.30.4.1 friend class QueryFactory [friend]

27.30.5 Member Data Documentation

27.30.5.1 DataSet gdcm::BaseRootQuery::mDataSet [protected]

27.30.5.2 std::string gdcm::BaseRootQuery::mHelpDescription [protected]

27.30.5.3 QueryImage gdcm::BaseRootQuery::mImage [protected]

27.30.5.4 QueryPatient gdcm::BaseRootQuery::mPatient [protected]

27.30.5.5 ERootType gdcm::BaseRootQuery::mRootType [protected]

27.30.5.6 QuerySeries gdcm::BaseRootQuery::mSeries [protected]

27.30.5.7 QueryStudy gdcm::BaseRootQuery::mStudy [protected]

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

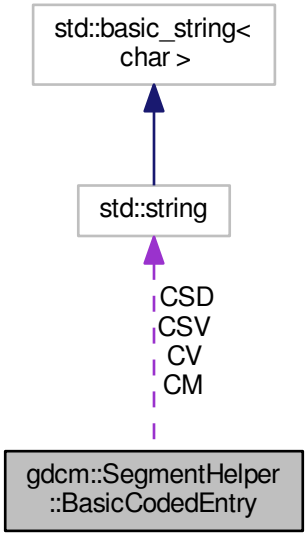
- gdcmBaseRootQuery.h

27.31 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 attibutes 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

27.31.1 Detailed Description

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

See also

PS 3.3 section 8.8.

27.31.2 Constructor & Destructor Documentation

27.31.2.1 gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ( ) [inline]

Constructor.

27.31.2.2 gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (const char * *a_CV*, const char * *a_CSD*, const char * *a_CM*) [inline]

constructor which defines type 1 attributes.

27.31.2.3 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.

27.31.3 Member Function Documentation

27.31.3.1 bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (const bool *checkOptionalAttributes* = false) const

Check if each attibutes of the basic coded entry is defined.

Parameters

checkOptional↩ Attributes	Check also type 1C attributes.

27.31.4 Member Data Documentation

27.31.4.1 std::string gdcm::SegmentHelper::BasicCodedEntry::CM

Coding Scheme Version attribute.

27.31.4.2 std::string gdcm::SegmentHelper::BasicCodedEntry::CSD

Code Value attribute.

27.31.4.3 std::string gdcm::SegmentHelper::BasicCodedEntry::CSV

Coding Scheme Designator attribute.

27.31.4.4 std::string gdcm::SegmentHelper::BasicCodedEntry::CV

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

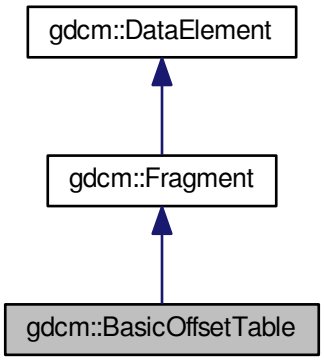
- gdcmSegmentHelper.h

27.32 gdcm::BasicOffsetTable Class Reference

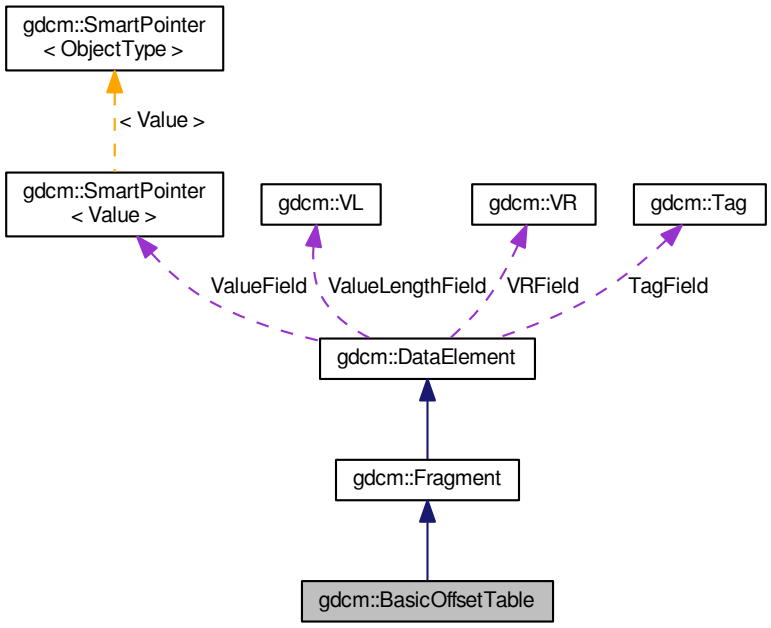
Class to represent a BasicOffsetTable.

`#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)

Friends

- std::ostream & operator<< (std::ostream &os, const BasicOffsetTable &val)

Additional Inherited Members

27.32.1 Detailed Description

Class to represent a BasicOffsetTable.

27.32.2 Constructor & Destructor Documentation

27.32.2.1 gdcm::BasicOffsetTable::BasicOffsetTable ( ) [inline]

27.32.3 Member Function Documentation

27.32.3.1 template<typename TSwap > std::istream& gdcm::BasicOffsetTable::Read (std::istream & *is*) [inline]

References gdcmDebugMacro.

27.32.4 Friends And Related Function Documentation

27.32.4.1 std::ostream& operator<< (std::ostream & *os,* const BasicOffsetTable & *val*) [friend]

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

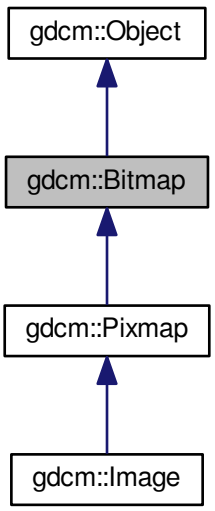
- gdcmBasicOffsetTable.h

27.33 gdcm::Bitmap Class Reference

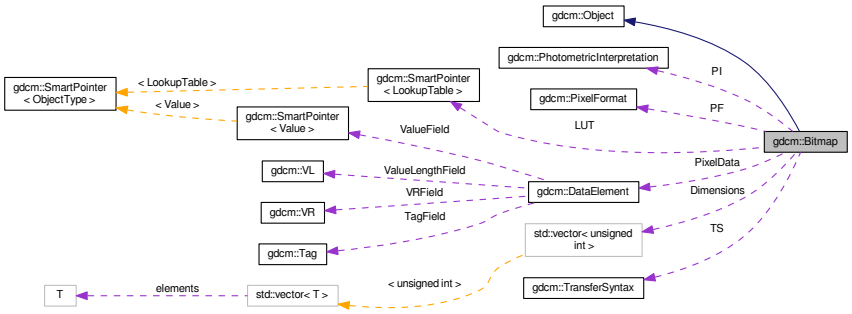
Bitmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

`#include <gdcmBitmap.h>`

Inheritance diagram for gdcm::Bitmap:



Collaboration diagram for gdcm::Bitmap:



Public Member Functions

- Bitmap ()
- ~Bitmap ()
- virtual bool AreOverlaysInPixelData () const
- void Clear ()
- bool GetBuffer (char *buffer) const

  Acces the raw data.
- unsigned long GetBufferLength () const
- unsigned int GetColumns () const
- const DataElement & GetDataElement () const
- DataElement & GetDataElement ()
- 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)...
- const LookupTable & GetLUT () const
- LookupTable & GetLUT ()
- bool GetNeedByteSwap () const
- 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
- const PixelFormat & GetPixelFormat () const

  Get/Set PixelFormat.
- PixelFormat & GetPixelFormat ()
- 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

- 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.

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 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

27.33.1 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)

Examples:

ExtractIconFromFile.cxx.

27.33.2 Member Typedef Documentation

27.33.2.1 typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr [protected]

27.33.3 Constructor & Destructor Documentation

27.33.3.1 gdcm::Bitmap::Bitmap ( )

27.33.3.2 gdcm::Bitmap::~Bitmap ( )

27.33.4 Member Function Documentation

27.33.4.1 virtual bool gdcm::Bitmap::AreOverlaysInPixelData ( ) const [inline], [virtual]

Reimplemented in gdcm::Pixmap.

27.33.4.2 void gdcm::Bitmap::Clear ( )

27.33.4.3 bool gdcm::Bitmap::ComputeLossyFlag ( ) [protected]

27.33.4.4 bool gdcm::Bitmap::GetBuffer (char * *buffer*) const

Acces the raw data.

Examples:

ConvertToQImage.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

27.33.4.5 bool gdcm::Bitmap::GetBuffer2 (std::ostream & *os*) const [protected]

27.33.4.6 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:

ConvertToQImage.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

27.33.4.7 unsigned int gdcm::Bitmap::GetColumns ( ) const [inline]

27.33.4.8 const DataElement& gdcm::Bitmap::GetDataElement ( ) const [inline]

Examples:

ExtractIconFromFile.cxx.

27.33.4.9 DataElement& gdcm::Bitmap::GetDataElement ( ) [inline]

27.33.4.10 unsigned int gdcm::Bitmap::GetDimension (unsigned int *idx*) const

27.33.4.11 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.

27.33.4.12 const LookupTable& gdcm::Bitmap::GetLUT ( ) const [inline]

Examples:

ExtractIconFromFile.cxx.

27.33.4.13 LookupTable& gdcm::Bitmap::GetLUT ( ) [inline]

27.33.4.14 bool gdcm::Bitmap::GetNeedByteSwap ( ) const [inline]

27.33.4.15 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:

HelloVizWorld.cxx, and threadgdcm.cxx.

27.33.4.16 const PhotometricInterpretation& gdcm::Bitmap::GetPhotometricInterpretation ( ) const

return the photometric interpretation

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, and HelloVizWorld.cxx.

27.33.4.17 const PixelFormat& gdcm::Bitmap::GetPixelFormat ( ) const [inline]

Get/Set PixelFormat.

Examples:

ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetJPEGSample↩Precision.cxx, and threadgdcm.cxx.

27.33.4.18 PixelFormat& gdcm::Bitmap::GetPixelFormat ( ) [inline]

27.33.4.19 unsigned int gdcm::Bitmap::GetPlanarConfiguration ( ) const

return the planar configuration

27.33.4.20 unsigned int gdcm::Bitmap::GetRows ( ) const [inline]

27.33.4.21 const TransferSyntax& gdcm::Bitmap::GetTransferSyntax ( ) const [inline]

Examples:

ExtractIconFromFile.cxx.

27.33.4.22 bool gdcm::Bitmap::IsEmpty ( ) const [inline]

27.33.4.23 bool gdcm::Bitmap::IsLossy ( ) const

Return whether or not the image was compressed using a lossy compressor or not.

27.33.4.24 bool gdcm::Bitmap::IsTransferSyntaxCompatible (TransferSyntax const & *ts*) const

27.33.4.25 void gdcm::Bitmap::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Object.

Reimplemented in gdcm::Image, and gdcm::Pixmap.

Examples:

ExtractIconFromFile.cxx.

27.33.4.26 void gdcm::Bitmap::SetColumns (unsigned int *col*) [inline]

27.33.4.27 void gdcm::Bitmap::SetDataElement (DataElement const & *de*) [inline]

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

27.33.4.28 void gdcm::Bitmap::SetDimension (unsigned int *idx,* unsigned int *dim*)

Examples:

csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

27.33.4.29 void gdcm::Bitmap::SetDimensions (const unsigned int *dims[3]*)

Examples:

CreateARGBImage.cxx, and CreateCMYKImage.cxx.

27.33.4.30 void gdcm::Bitmap::SetLossyFlag (bool *f*) [inline]

Specifically set that the image was compressed using a lossy compression mechanism.

27.33.4.31 void gdcm::Bitmap::SetLUT (LookupTable const & *lut*) [inline]

Set/Get LUT.

27.33.4.32 void gdcm::Bitmap::SetNeedByteSwap (bool *b*) [inline]

27.33.4.33 void gdcm::Bitmap::SetNumberOfDimensions (unsigned int *dim*)

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, FileChangeTS.cs, FileChangeTSLossy.cs, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

27.33.4.34 void gdcm::Bitmap::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*)

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

27.33.4.35 void gdcm::Bitmap::SetPixelFormat (PixelFormat const & *pf*) [inline]

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, and iU22tomultisc.cxx.

References gdcm::PixelFormat::Validate().

27.33.4.36 void gdcm::Bitmap::SetPlanarConfiguration (unsigned int *pc*)

Warning

you need to call SetPixelFormat first (before SetPlanarConfiguration) for consistency checking

27.33.4.37 void gdcm::Bitmap::SetRows (unsigned int *rows*) [inline]

27.33.4.38 void gdcm::Bitmap::SetTransferSyntax (TransferSyntax const & *ts*) [inline]

Transfer syntax.

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, and MergeTwoFiles.cxx.

27.33.4.39 bool gdcm::Bitmap::TryJPEG2000Codec (char * *buffer,* bool & *lossyflag*) const [protected]

27.33.4.40 bool gdcm::Bitmap::TryJPEG2000Codec2 (std::ostream & *os*) const [protected]

27.33.4.41 bool gdcm::Bitmap::TryJPEGCodec (char * *buffer,* bool & *lossyflag*) const [protected]

27.33.4.42 bool gdcm::Bitmap::TryJPEGCodec2 (std::ostream & *os*) const [protected]

27.33.4.43 bool gdcm::Bitmap::TryJPEGLSCodec (char * *buffer,* bool & *lossyflag*) const [protected]

27.33.4.44 bool gdcm::Bitmap::TryKAKADUCodec (char * *buffer,* bool & *lossyflag*) const [protected]

27.33.4.45 bool gdcm::Bitmap::TryPVRGCodec (char * *buffer,* bool & *lossyflag*) const [protected]

27.33.4.46 bool gdcm::Bitmap::TryRAWCodec (char * *buffer,* bool & *lossyflag*) const [protected]

27.33.4.47 bool gdcm::Bitmap::TryRLECodec (char * *buffer,* bool & *lossyflag*) const [protected]

27.33.5 Friends And Related Function Documentation

27.33.5.1 friend class ImageChangeTransferSyntax [friend]

27.33.5.2 friend class PixmapReader [friend]

27.33.6 Member Data Documentation

27.33.6.1 std::vector<unsigned int> gdcm::Bitmap::Dimensions [protected]

27.33.6.2 bool gdcm::Bitmap::LossyFlag [protected]

27.33.6.3 LUTPtr gdcm::Bitmap::LUT [protected]

27.33.6.4 bool gdcm::Bitmap::NeedByteSwap [protected]

27.33.6.5 unsigned int gdcm::Bitmap::NumberOfDimensions [protected]

27.33.6.6 PixelFormat gdcm::Bitmap::PF [protected]

27.33.6.7 PhotometricInterpretation gdcm::Bitmap::PI [protected]

27.33.6.8 DataElement gdcm::Bitmap::PixelData [protected]

27.33.6.9 unsigned int gdcm::Bitmap::PlanarConfiguration [protected]

27.33.6.10 TransferSyntax gdcm::Bitmap::TS [protected]

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

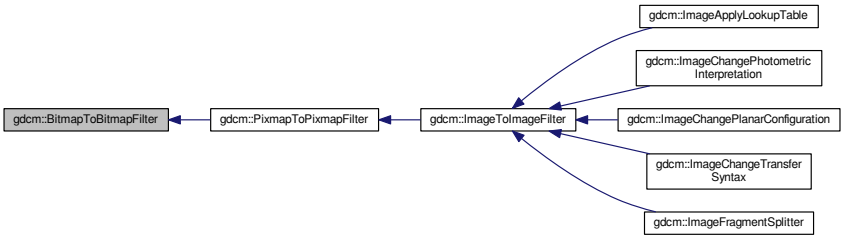
- gdcmBitmap.h

27.34 gdcm::BitmapToBitmapFilter Class Reference

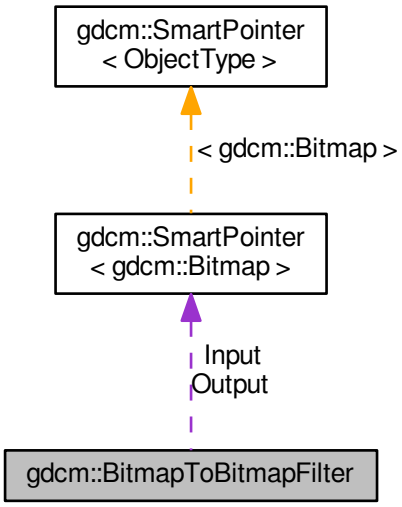
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

`#include <gdcmBitmapToBitmapFilter.h>`

Inheritance diagram for gdcm::BitmapToBitmapFilter:



Collaboration diagram for gdcm::BitmapToBitmapFilter:



Public Member Functions

- BitmapToBitmapFilter ()
- ~BitmapToBitmapFilter ()
- 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

27.34.1 Detailed Description

BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image.

27.34.2 Constructor & Destructor Documentation

27.34.2.1 gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ( )

27.34.2.2 gdcm::BitmapToBitmapFilter::~BitmapToBitmapFilter ( ) [inline]

27.34.3 Member Function Documentation

27.34.3.1 const Bitmap& gdcm::BitmapToBitmapFilter::GetOutput ( ) const [inline]

Get Output image.

27.34.3.2 const Bitmap& gdcm::BitmapToBitmapFilter::GetOutputAsBitmap ( ) const

27.34.3.3 void gdcm::BitmapToBitmapFilter::SetInput (const Bitmap & *image*)

Set input image.

Examples:

CompressImage.cxx.

27.34.4 Member Data Documentation

27.34.4.1 SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input [protected]

27.34.4.2 SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output [protected]

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

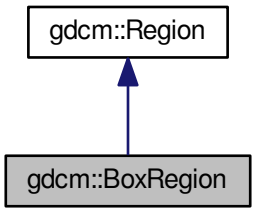
- gdcmBitmapToBitmapFilter.h

27.35 gdcm::BoxRegion Class Reference

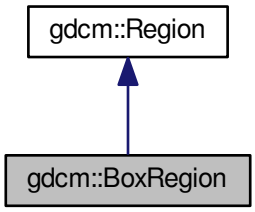
Class for manipulation box region This is a very simple implementation of the Region class. It only support 3D box type region. It assumes the 3D Box does not have a tilt Origin is as (0,0,0)

`#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 ()
- size_t Area () const

  compute the area
- Region * Clone () const
- BoxRegion ComputeBoundingBox ()

  Return the Axis-Aligned minimum bounding box for all regions.

- bool Empty () const

  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

  return whether this is valid domain
- void operator= (const BoxRegion &)
- void Print (std::ostream &os=std::cout) const

  Print.
- void SetDomain (unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax)

  Set domain.

Static Public Member Functions

- static BoxRegion BoundingBox (BoxRegion const &b1, BoxRegion const &b2)

  Helper class to compute the bounding box of two BoxRegion.

27.35.1 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)

27.35.2 Constructor & Destructor Documentation

27.35.2.1 gdcm::BoxRegion::BoxRegion ( )

27.35.2.2 gdcm::BoxRegion::~BoxRegion ( )

27.35.2.3 gdcm::BoxRegion::BoxRegion (const BoxRegion &)

copy/cstor and al.

27.35.3 Member Function Documentation

27.35.3.1 size_t gdcm::BoxRegion::Area ( ) const [virtual]

compute the area

Implements gdcm::Region.

27.35.3.2 static BoxRegion gdcm::BoxRegion::BoundingBox (BoxRegion const & *b1*, BoxRegion const & *b2*) [static]

Helper class to compute the bounding box of two BoxRegion.

27.35.3.3 Region* gdcm::BoxRegion::Clone ( ) const [virtual]

Implements gdcm::Region.

27.35.3.4 BoxRegion gdcm::BoxRegion::ComputeBoundingBox ( ) [virtual]

Return the Axis-Aligned minimum bounding box for all regions.

Implements gdcm::Region.

27.35.3.5 bool gdcm::BoxRegion::Empty ( ) const [virtual]

return whether this domain is empty:

Implements gdcm::Region.

27.35.3.6 unsigned int gdcm::BoxRegion::GetXMax ( ) const

27.35.3.7 unsigned int gdcm::BoxRegion::GetXMin ( ) const

Get domain.

27.35.3.8 unsigned int gdcm::BoxRegion::GetYMax ( ) const

27.35.3.9 unsigned int gdcm::BoxRegion::GetYMin ( ) const

27.35.3.10 unsigned int gdcm::BoxRegion::GetZMax ( ) const

27.35.3.11 unsigned int gdcm::BoxRegion::GetZMin ( ) const

27.35.3.12 bool gdcm::BoxRegion::IsValid ( ) const [virtual]

return whether this is valid domain

Implements gdcm::Region.

27.35.3.13 void gdcm::BoxRegion::operator= (const BoxRegion &)

27.35.3.14 void gdcm::BoxRegion::Print (std::ostream & *os* = std::cout) const [virtual]

Print.

Reimplemented from gdcm::Region.

27.35.3.15 void gdcm::BoxRegion::SetDomain (unsigned int *xmin,* unsigned int *xmax,* unsigned int *ymin,* unsigned int *ymax,* unsigned int *zmin,* unsigned int *zmax*)

Set domain.

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

- gdcmBoxRegion.h

27.36 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 ()

27.36.1 Detailed Description

ByteBuffer.

Detailled description here

Note

looks like a std::streambuf or std::filebuf class with the get and peek pointer

27.36.2 Constructor & Destructor Documentation

27.36.2.1 gdcm::ByteBuffer::ByteBuffer ( ) [inline]

27.36.3 Member Function Documentation

27.36.3.1 char* gdcm::ByteBuffer::Get (int *len*) [inline]

27.36.3.2 const char* gdcm::ByteBuffer::GetStart ( ) const [inline]

27.36.3.3 void gdcm::ByteBuffer::ShiftEnd (int *len*) [inline]

27.36.3.4 void gdcm::ByteBuffer::UpdatePosition ( ) [inline]

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

- gdcmByteBuffer.h

27.37 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 ()

27.37.1 Detailed Description

template<class T>class gdcm::ByteSwap< T >

ByteSwap.

Perform machine dependent byte swaping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian). TODO: bswap_32 / bswap_64 ...

Examples:

TestByteSwap.cxx.

27.37.2 Member Function Documentation

27.37.2.1 template<class T > static void gdcm::ByteSwap< T >::Swap (T & *p*) [static]

27.37.2.2 template<class T > static void gdcm::ByteSwap< T >::SwapFromSwapCodeIntoSystem (T & *p,* SwapCode const & *sc*) [static]

Examples:

TestByteSwap.cxx.

27.37.2.3 template<class T > static void gdcm::ByteSwap< T >::SwapRange (T * *p,* unsigned int *num*) [static]

27.37.2.4 template<class T > static void gdcm::ByteSwap< T >::SwapRangeFromSwapCodeIntoSystem (T * *p,* SwapCode const & *sc,* std::streamoff *num*) [static]

Examples:

TestByteSwap.cxx.

27.37.2.5 template<class T > static bool gdcm::ByteSwap< T >::SystemIsBigEndian ( ) [static]

Query the machine Endian-ness.

27.37.2.6 template<class T > static bool gdcm::ByteSwap< T >::SystemIsLittleEndian ( ) [static]

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

- gdcmByteSwap.h

27.38 gdcm::ByteSwapFilter Class Reference

ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

`#include <gdcmByteSwapFilter.h>`

Public Member Functions

- ByteSwapFilter (DataSet &ds)
- ~ByteSwapFilter ()
- bool ByteSwap ()
- void SetByteSwapTag (bool b)

27.38.1 Detailed Description

ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

27.38.2 Constructor & Destructor Documentation

27.38.2.1 gdcm::ByteSwapFilter::ByteSwapFilter (DataSet & *ds*) [inline]

27.38.2.2 gdcm::ByteSwapFilter::~ByteSwapFilter ( )

27.38.3 Member Function Documentation

27.38.3.1 bool gdcm::ByteSwapFilter::ByteSwap ( )

27.38.3.2 void gdcm::ByteSwapFilter::SetByteSwapTag (bool *b*) [inline]

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

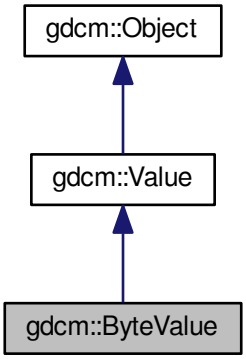
- gdcmByteSwapFilter.h

27.39 gdcm::ByteValue Class Reference

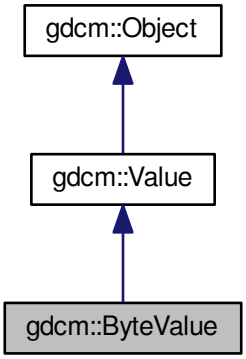
Class to represent binary value (array of bytes)

`#include <gdcmByteValue.h>`

Inheritance diagram for gdcm::ByteValue:



Collaboration diagram for gdcm::ByteValue:



Public Member Functions

- ByteValue (const char *array=0, VL const &vl=0)
- ByteValue (std::vector< char > &v)
- ~ByteValue ()
- void Append (ByteValue const &bv)
- void Clear ()
- VL ComputeLength () const
- void Fill (char c)

- bool GetBuffer (char *buffer, unsigned long length) const
- VL GetLength () const
- const char * GetPointer () 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
- 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 , typename TType >
  std::istream & Read (std::istream &is, bool readvalues=true)
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- void SetLength (VL vl)
- template<typename TSwap , typename TType >
  std::ostream const & Write (std::ostream &os) const
- template<typename TSwap >
  std::ostream const & Write (std::ostream &os) const
- bool WriteBuffer (std::ostream &os) const

Protected Member Functions

- void Print (std::ostream &os) const
- void SetLengthOnly (VL vl)

27.39.1 Detailed Description

Class to represent binary value (array of bytes)

Note

Examples:

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, and rle2img.cxx.

27.39.2 Constructor & Destructor Documentation

27.39.2.1 gdcm::ByteValue::ByteValue (const char * *array* = 0, VL const & *vl* = 0) [inline]

References gdcmDebugMacro.

27.39.2.2 gdcm::ByteValue::ByteValue (std::vector< char > & *v*) [inline]

Warning

casting to uint32_t

27.39.2.3 gdcm::ByteValue::~ByteValue ( ) [inline]

27.39.3 Member Function Documentation

27.39.3.1 void gdcm::ByteValue::Append (ByteValue const & *bv*)

27.39.3.2 void gdcm::ByteValue::Clear ( ) [inline], [virtual]

Implements gdcm::Value.

27.39.3.3 VL gdcm::ByteValue::ComputeLength ( ) const [inline]

Referenced by gdcm::Fragment::Write().

27.39.3.4 void gdcm::ByteValue::Fill (char *c*) [inline]

Examples:

DuplicatePCDE.cxx.

27.39.3.5 bool gdcm::ByteValue::GetBuffer (char * *buffer,* unsigned long *length*) const

Examples:

FixJAIBugJPEGLS.cxx.

27.39.3.6 VL gdcm::ByteValue::GetLength ( ) const [inline], [virtual]

Implements gdcm::Value.

Examples:

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilips↩ ECHO.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicit↩ LengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::SequenceOfFragments::ReadValue(), gdcm::Element< VR::OB, VM::V↩ M1_n >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::Set↩ ByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcm↩ ::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< VR::OB, VM::VM1_n >::↩ SetNoSwap(), gdcm::Element< TVR, VM::VM1_n >::SetNoSwap(), and gdcm::Fragment::Write().

27.39.3.7 const char* gdcm::ByteValue::GetPointer ( ) const [inline]

Examples:

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, GetSubSequenceData.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::SequenceOfFragments::ReadValue(), gdcm::Element< VR::OB, VM::VM1_n >::Set(), gdcm::Element< TVR, VM::VM1_n >::Set(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcm::Element< VR::OB, VM::VM1_n >::SetNoSwap(), and gdcm::Element< TVR, VM::VM1_n >::SetNoSwap().

27.39.3.8 bool gdcm::ByteValue::IsEmpty ( ) const [inline]

27.39.3.9 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...

27.39.3.10 gdcm::ByteValue::operator const std::vector< char > & ( ) const [inline]

27.39.3.11 ByteValue& gdcm::ByteValue::operator= (const ByteValue & *val*) [inline]

27.39.3.12 bool gdcm::ByteValue::operator== (const ByteValue & *val*) const [inline]

27.39.3.13 bool gdcm::ByteValue::operator== (const Value & *val*) const [inline], [virtual]

Implements gdcm::Value.

27.39.3.14 void gdcm::ByteValue::Print (std::ostream & *os*) const [inline], [protected], [virtual]

Reimplemented from gdcm::Object.

27.39.3.15 void gdcm::ByteValue::PrintASCII (std::ostream & *os,* VL *maxlength*) const

27.39.3.16 void gdcm::ByteValue::PrintASCIIXML (std::ostream & *os*) const

27.39.3.17 void gdcm::ByteValue::PrintGroupLength (std::ostream & *os*) [inline]

27.39.3.18 void gdcm::ByteValue::PrintHex (std::ostream & *os,* VL *maxlength*) const

27.39.3.19 void gdcm::ByteValue::PrintHexXML (std::ostream & *os*) const

27.39.3.20 void gdcm::ByteValue::PrintPNXML (std::ostream & *os*) const

To Print Values in Native DICOM format

27.39.3.21 template<typename TSwap , typename TType > std::istream& gdcm::ByteValue::Read (std::istream & *is*, bool *readvalues* = true) [inline]

27.39.3.22 template<typename TSwap > std::istream& gdcm::ByteValue::Read (std::istream & *is*) [inline]

27.39.3.23 void gdcm::ByteValue::SetLength (VL *vl*) [inline], [virtual]

Implements gdcm::Value.

References gdcmDebugMacro, gdcm::VL::IsOdd(), and gdcm::VL::IsUndefined().

27.39.3.24 void gdcm::ByteValue::SetLengthOnly (VL *vl*) [inline], [protected], [virtual]

Reimplemented from gdcm::Value.

27.39.3.25 template<typename TSwap , typename TType > std::ostream const& gdcm::ByteValue::Write (std::ostream & *os*) const [inline]

Referenced by gdcm::Fragment::Write().

27.39.3.26 template<typename TSwap > std::ostream const& gdcm::ByteValue::Write (std::ostream & *os*) const [inline]

27.39.3.27 bool gdcm::ByteValue::WriteBuffer (std::ostream & *os*) const [inline]

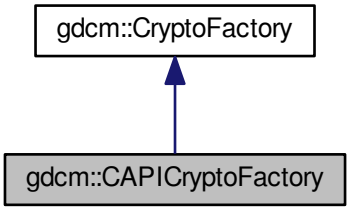
The documentation for this class was generated from the following file:

- gdcmByteValue.h

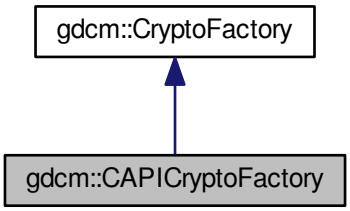
27.40 gdcm::CAPICryptoFactory Class Reference

`#include <gdcmCAPICryptoFactory.h>`

Inheritance diagram for gdcm::CAPICryptoFactory:



Collaboration diagram for gdcm::CAPICryptoFactory:



Public Member Functions

- CAPICryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Additional Inherited Members

27.40.1 Constructor & Destructor Documentation

27.40.1.1 gdcm::CAPICryptoFactory::CAPICryptoFactory (CryptoLib *id*)

27.40.2 Member Function Documentation

27.40.2.1 CryptographicMessageSyntax* gdcm::CAPICryptoFactory::CreateCMSProvider ( ) [virtual]

Implements gdcm::CryptoFactory.

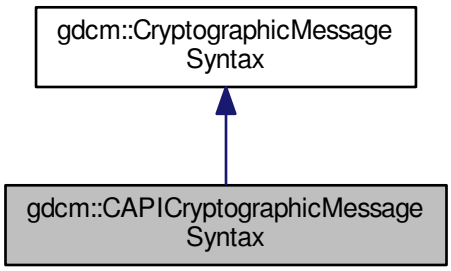
The documentation for this class was generated from the following file:

- gdcmCAPICryptoFactory.h

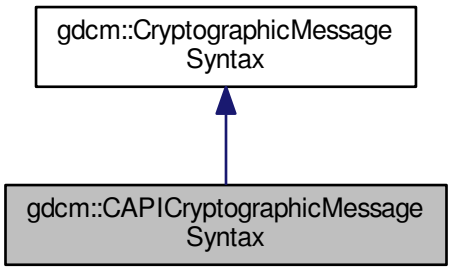
27.41 gdcm::CAPICryptographicMessageSyntax Class Reference

`#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)

Additional Inherited Members

27.41.1 Constructor & Destructor Documentation

27.41.1.1 gdcm::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax ( )

27.41.1.2 gdcm::CAPICryptographicMessageSyntax::~CAPICryptographicMessageSyntax ( )

27.41.2 Member Function Documentation

27.41.2.1 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.

27.41.2.2 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.

27.41.2.3 CipherTypes gdcm::CAPICryptographicMessageSyntax::GetCipherType ( ) const [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.41.2.4 bool gdcm::CAPICryptographicMessageSyntax::GetInitialized ( ) const [inline]

27.41.2.5 bool gdcm::CAPICryptographicMessageSyntax::ParseCertificateFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.41.2.6 bool gdcm::CAPICryptographicMessageSyntax::ParseKeyFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.41.2.7 void gdcm::CAPICryptographicMessageSyntax::SetCipherType (CipherTypes *type*)

27.41.2.8 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:

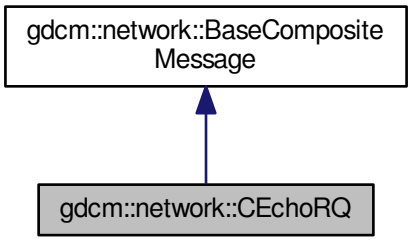
- gdcmCAPICryptographicMessageSyntax.h

27.42 gdcm::network::CEchoRQ Class Reference

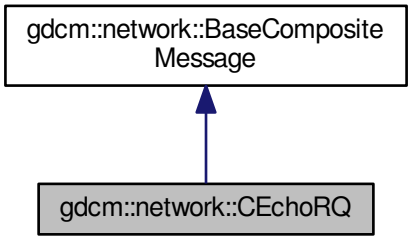
CEchoRQ this file defines the messages for the cecho action.

`#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 *in↩ RootQuery)

Public Attributes

- UIComp AffectedSOPClassUID
- uint16_t MessageID

27.42.1 Detailed Description

CEchoRQ this file defines the messages for the cecho action.

27.42.2 Member Function Documentation

27.42.2.1 std::vector<PresentationDataValue> gdcm::network::CEchoRQ::ConstructPDV (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [virtual]

Implements gdcm::network::BaseCompositeMessage.

27.42.3 Member Data Documentation

27.42.3.1 UIComp gdcm::network::CEchoRQ::AffectedSOPClassUID

27.42.3.2 uint16_t gdcm::network::CEchoRQ::MessageID

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

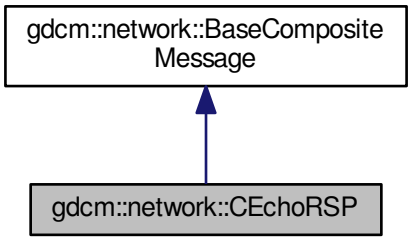
- gdcmCEchoMessages.h
- gdcmDIMSE.h

27.43 gdcm::network::CEchoRSP Class Reference

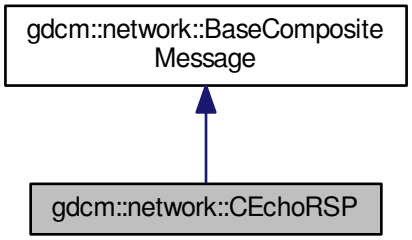
CEchoRSP this file defines the messages for the cecho action.

`#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)

27.43.1 Detailed Description

CEchoRSP this file defines the messages for the cecho action.

27.43.2 Member Function Documentation

27.43.2.1 std::vector<PresentationDataValue> gdcm::network::CEchoRSP::ConstructPDVByDataSet (const DataSet * *inDataSet*)

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

- gdcmCEchoMessages.h

27.44 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

27.44.1 Detailed Description

PS 3.4 - 2009 Table B.2-1 C-STORE STATUS

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

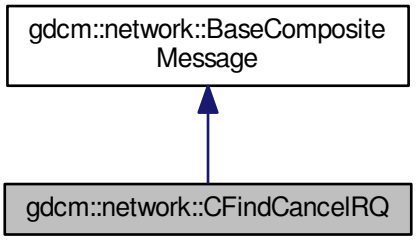
- gdcmDIMSE.h

27.45 gdcm::network::CFindCancelRQ Class Reference

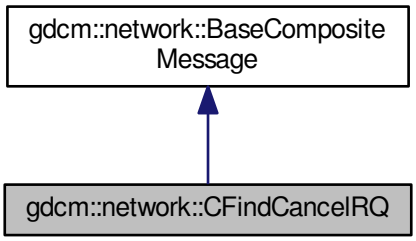
CFindCancelRQ this file defines the messages for the cfind action.

`#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)

27.45.1 Detailed Description

CFindCancelRQ this file defines the messages for the cfind action.

27.45.2 Member Function Documentation

27.45.2.1 std::vector<PresentationDataValue> gdcm::network::CFindCancelRQ::ConstructPDVByDataSet (const DataSet * *inDataSet*)

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

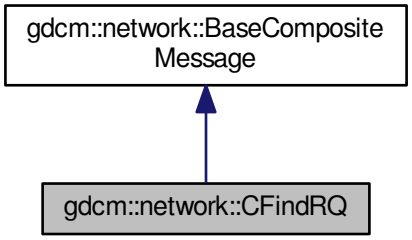
- gdcmCFindMessages.h

27.46 gdcm::network::CFindRQ Class Reference

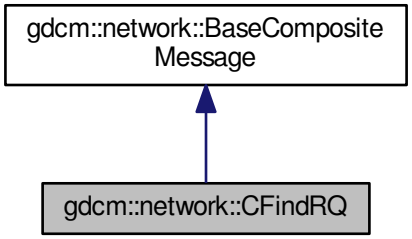
CFindRQ this file defines the messages for the cfind action.

`#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 *in↩

RootQuery)

27.46.1 Detailed Description

CFindRQ this file defines the messages for the cfind action.

27.46.2 Member Function Documentation

27.46.2.1 std::vector<PresentationDataValue> gdcm::network::CFindRQ::ConstructPDV (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [virtual]

Implements gdcm::network::BaseCompositeMessage.

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

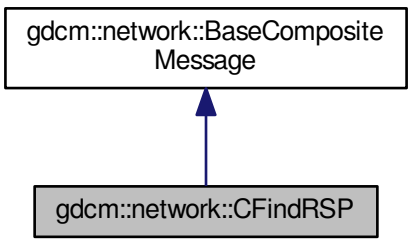
- gdcmCFindMessages.h

27.47 gdcm::network::CFindRSP Class Reference

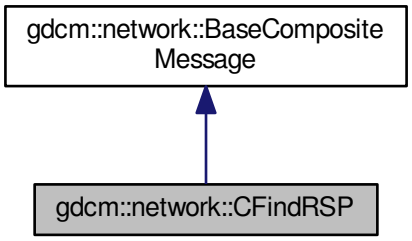
CFindRSP this file defines the messages for the cfind action.

`#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)

27.47.1 Detailed Description

CFindRSP this file defines the messages for the cfind action.

27.47.2 Member Function Documentation

27.47.2.1 std::vector<PresentationDataValue> gdcm::network::CFindRSP::ConstructPDVByDataSet (const DataSet * *inDataSet*)

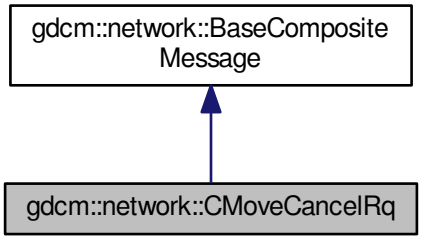
The documentation for this class was generated from the following file:

- gdcmCFindMessages.h

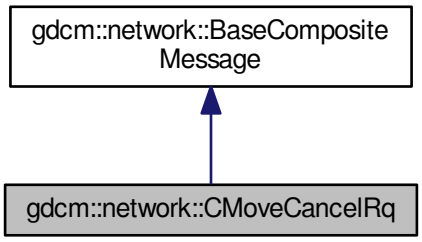
27.48 gdcm::network::CMoveCancelRq Class Reference

```
#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)

27.48.1 Member Function Documentation

27.48.1.1 std::vector<PresentationDataValue> gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (const DataSet * *inDataSet*)

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

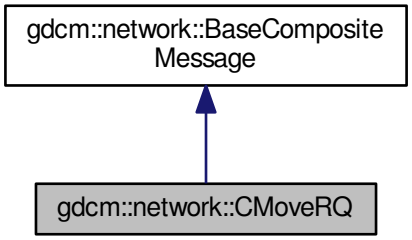
- gdcmCMoveMessages.h

27.49 gdcm::network::CMoveRQ Class Reference

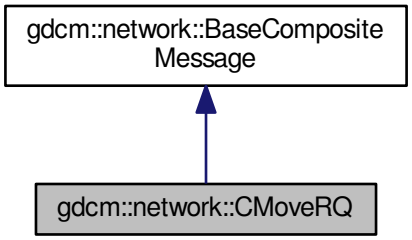
CMoveRQ this file defines the messages for the cmove action.

`#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)

27.49.1 Detailed Description

CMoveRQ this file defines the messages for the cmove action.

27.49.2 Member Function Documentation

27.49.2.1 std::vector<PresentationDataValue> gdcm::network::CMoveRQ::ConstructPDV (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [virtual]

Implements gdcm::network::BaseCompositeMessage.

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

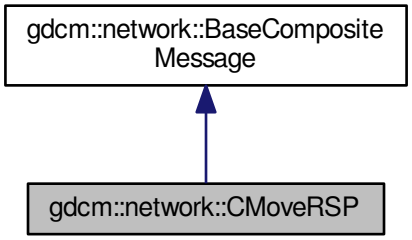
- gdcmCMoveMessages.h

27.50 gdcm::network::CMoveRSP Class Reference

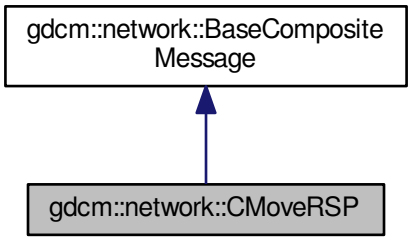
CMoveRSP this file defines the messages for the cmove action.

`#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)

27.50.1 Detailed Description

CMoveRSP this file defines the messages for the cmove action.

27.50.2 Member Function Documentation

27.50.2.1 std::vector<PresentationDataValue> gdcm::network::CMoveRSP::ConstructPDVByDataSet (const DataSet * *inDataSet*)

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

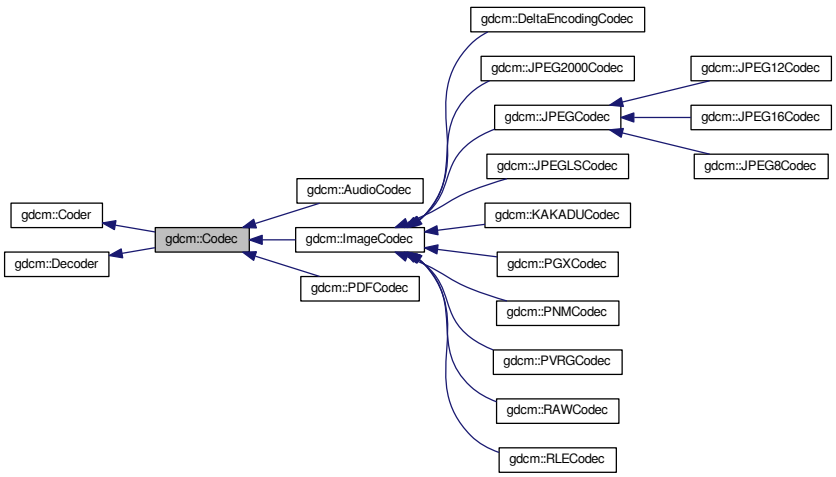
- gdcmCMoveMessages.h

27.51 gdcm::Codec Class Reference

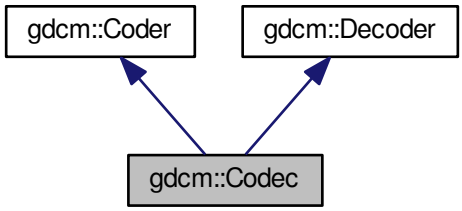
Codec class.

`#include <gdcmCodec.h>`

Inheritance diagram for gdcm::Codec:



Collaboration diagram for gdcm::Codec:



Additional Inherited Members

27.51.1 Detailed Description

Codec class.

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

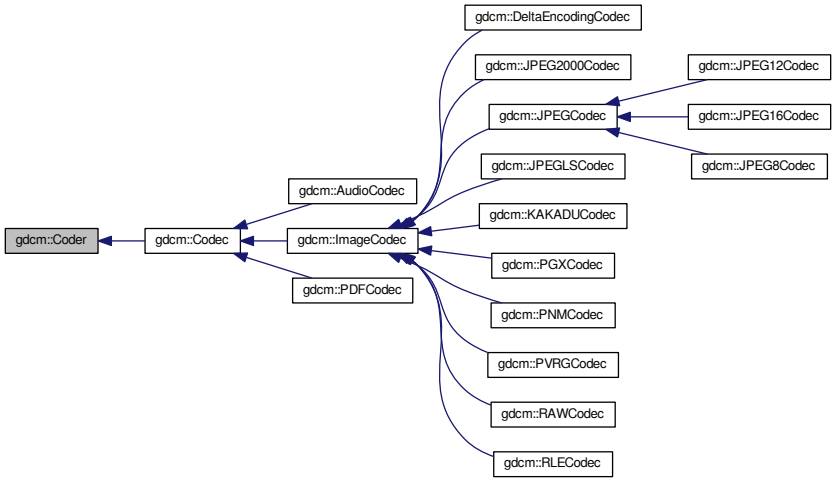
- gdcmCodec.h

27.52 gdcm::Coder Class Reference

Coder.

```
#include <gdcmCoder.h>
```

Inheritance diagram for gdcm::Coder:



Public Member Functions

- virtual ∼Coder ()
- 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)

27.52.1 Detailed Description

Coder.

27.52.2 Constructor & Destructor Documentation

27.52.2.1 virtual gdcm::Coder::∼Coder ( ) [inline], [virtual]

27.52.3 Member Function Documentation

27.52.3.1 virtual bool gdcm::Coder::CanCode (TransferSyntax const &) const [pure virtual]

Return whether this coder support this transfer syntax (can code it)

Implemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::JPE↩GLSCodec, gdcm::ImageCodec, gdcm::PNMCodec, gdcm::PGXCodec, gdcm::KAKADUCodec, gdcm::RAWCodec, gdcm::AudioCodec, and gdcm::PDFCodec.

27.52.3.2 virtual bool gdcm::Coder::Code (DataElement const & *in_*, DataElement & *out_*) [inline], [virtual]

Code.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

27.52.3.3 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

27.53 gdcm::CodeString Class Reference

CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

`#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 value_type *s)
- CodeString (const value_type *s, size_type n)
- CodeString (const InternalClass &s, size_type pos=0, size_type n=InternalClass::npos)
- 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)

27.53.1 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.

27.53.2 Member Typedef Documentation

27.53.2.1 typedef InternalClass::const_iterator gdcm::CodeString::const_iterator

27.53.2.2 typedef InternalClass::const_reference gdcm::CodeString::const_reference

27.53.2.3 typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator

27.53.2.4 typedef InternalClass::difference_type gdcm::CodeString::difference_type

27.53.2.5 typedef InternalClass::iterator gdcm::CodeString::iterator

27.53.2.6 typedef InternalClass::pointer gdcm::CodeString::pointer

27.53.2.7 typedef InternalClass::reference gdcm::CodeString::reference

27.53.2.8 typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator

27.53.2.9 typedef InternalClass::size_type gdcm::CodeString::size_type

27.53.2.10 typedef InternalClass::value_type gdcm::CodeString::value_type

27.53.3 Constructor & Destructor Documentation

27.53.3.1 gdcm::CodeString::CodeString ( ) [inline]

CodeString constructors.

27.53.3.2 gdcm::CodeString::CodeString (const value_type * *s*) [inline]

27.53.3.3 gdcm::CodeString::CodeString (const value_type * *s*, size_type *n*) [inline]

27.53.3.4 gdcm::CodeString::CodeString (const InternalClass & *s*, size_type *pos* = 0, size_type *n* = InternalClass::npos) [inline]

27.53.4 Member Function Documentation

27.53.4.1 std::string gdcm::CodeString::GetAsString ( ) const [inline]

Return the full code string as std::string.

27.53.4.2 bool gdcm::CodeString::IsValid ( ) const

Check if CodeString obj is correct..

27.53.4.3 size_type gdcm::CodeString::Size ( ) const [inline]

Return the size of the string.

27.53.4.4 std::string gdcm::CodeString::TrimInternal ( ) const [inline], [protected]

27.53.5 Friends And Related Function Documentation

27.53.5.1 bool operator!= (const CodeString & *ref*, const CodeString & *cs*) [friend]

27.53.5.2 std::ostream& operator<< (std::ostream & *os*, const CodeString & *str*) [friend]

27.53.5.3 bool operator== (const CodeString & *ref*, const CodeString & *cs*) [friend]

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

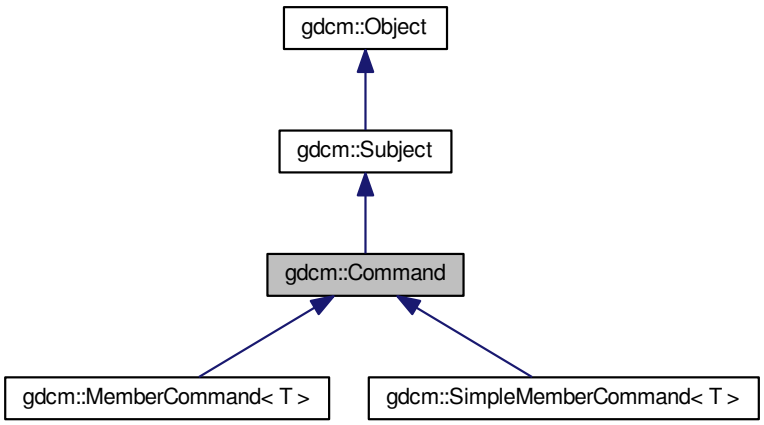
- gdcmCodeString.h

27.54 gdcm::Command Class Reference

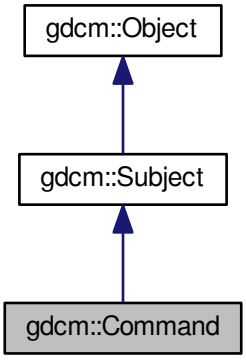
Command superclass for callback/observer methods.

`#include <gdcmCommand.h>`

Inheritance diagram for gdcm::Command:



Collaboration diagram for gdcm::Command:



Public Member Functions

- virtual void Execute (Subject ∗caller, const Event &event)=0

  Abstract method that defines the action to be taken by the command.
- virtual void Execute (const Subject ∗caller, const Event &event)=0

Protected Member Functions

- Command ()
- ~Command ()

27.54.1 Detailed Description

Command superclass for callback/observer methods.

See also

Subject

27.54.2 Constructor & Destructor Documentation

27.54.2.1 gdcm::Command::Command ( ) [protected]

27.54.2.2 gdcm::Command::~Command ( ) [protected]

27.54.3 Member Function Documentation

27.54.3.1 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::SimpleMemberCommand< T >, and gdcm::MemberCommand< T >.

27.54.3.2 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::SimpleMemberCommand< T >, and gdcm::MemberCommand< T >.

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

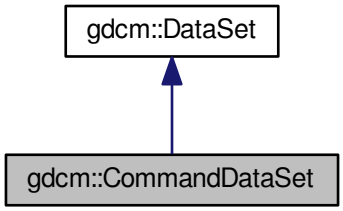
- gdcmCommand.h

27.55 gdcm::CommandDataSet Class Reference

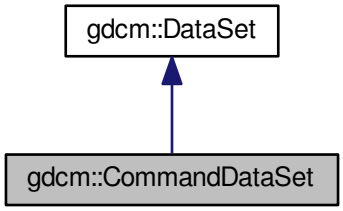
Class to represent a Command DataSet.

`#include <gdcmCommandDataSet.h>`

Inheritance diagram for gdcm::CommandDataSet:



Collaboration diagram for gdcm::CommandDataSet:



Public Member Functions

- CommandDataSet ()
- ~CommandDataSet ()
- 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.

Friends

- std::ostream & operator<< (std::ostream &_os, const CommandDataSet &_val)

Additional Inherited Members

27.55.1 Detailed Description

Class to represent a Command DataSet.

See also

DataSet

27.55.2 Constructor & Destructor Documentation

27.55.2.1 gdcm::CommandDataSet::CommandDataSet ( ) [inline]

27.55.2.2 gdcm::CommandDataSet::~CommandDataSet ( ) [inline]

27.55.3 Member Function Documentation

27.55.3.1 void gdcm::CommandDataSet::Insert (const DataElement & *de*) [inline]

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

27.55.3.2 std::istream& gdcm::CommandDataSet::Read (std::istream & *is*)

Read.

27.55.3.3 void gdcm::CommandDataSet::Replace (const DataElement & *de*) [inline]

References gdcm::DataElement::GetTag().

27.55.3.4 std::ostream& gdcm::CommandDataSet::Write (std::ostream & *os*) const

Write.

27.55.4 Friends And Related Function Documentation

27.55.4.1 std::ostream& operator<< (std::ostream & *_os,* const CommandDataSet & *_val*) [friend]

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

- gdcmCommandDataSet.h

27.56 gdcm::network::CompositeMessageFactory Class Reference

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).

```
#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)
- static std::vector
  < PresentationDataValue > ConstructCStoreRSP (const DataSet *inDataSet, const BasePDU *inPC)

27.56.1 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).

27.56.2 Member Function Documentation

27.56.2.1 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCEchoRQ (const ULConnection & *inConnection*) [static]

27.56.2.2 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCFindRQ (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [static]

27.56.2.3 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCMoveRQ (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [static]

27.56.2.4 static std::vector<PresentationDataValue> gdcm::network::CompositeMessageFactory::ConstructCStoreRQ (const ULConnection & *inConnection,* const File & *file*) [static]

27.56.2.5 static 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

27.57 gdcm::CompositeNetworkFunctions Class Reference

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:

`#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=NULL, const char *call=NULL)
- static bool CFind (const char *remote, uint16_t portno, const BaseRootQuery *query, std::vector< DataSet > &retDataSets, const char *aetitle=NULL, const char *call=NULL)
- static bool CMove (const char *remote, uint16_t portno, const BaseRootQuery *query, uint16_t portscp, const char *aetitle=NULL, const char *call=NULL, const char *outputdir=NULL)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const DataSet &queryds, bool inMove=false)
- static BaseRootQuery * ConstructQuery (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, bool inMove=false)
- static bool CStore (const char *remote, uint16_t portno, const Directory::FilenamesType &filenames, const char *aetitle=NULL, const char *call=NULL)

27.57.1 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)

27.57.2 Member Typedef Documentation

27.57.2.1 typedef std::vector< KeyValuePairType > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType

27.57.2.2 typedef std::pair<Tag, std::string> gdcm::CompositeNetworkFunctions::KeyValuePairType

27.57.3 Member Function Documentation

27.57.3.1 static bool gdcm::CompositeNetworkFunctions::CEcho (const char * *remote,* uint16_t *portno,* const char * *aetitle =* NULL, const char * *call =* NULL) [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.

27.57.3.2 static bool gdcm::CompositeNetworkFunctions::CFind (const char * *remote,* uint16_t *portno,* const **BaseRootQuery** * *query,* std::vector< **DataSet** > & *retDataSets,* const char * *aetitle =* NULL, const char * *call =* NULL) [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.

27.57.3.3 static bool gdcm::CompositeNetworkFunctions::CMove (const char * *remote,* uint16_t *portno,* const **BaseRootQuery** * *query,* uint16_t *portscp,* const char * *aetitle =* NULL, const char * *call =* NULL, const char * *outputdir =* NULL) [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.

27.57.3.4 static BaseRootQuery* gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType *inRootType*, EQueryLevel *inQueryLevel*, const DataSet & *queryds*, bool *inMove* = `false`) [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)

27.57.3.5 static BaseRootQuery* gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType *inRootType*, EQueryLevel *inQueryLevel*, const KeyValuePairArrayType & *keys*, bool *inMove* = `false`) [static]

Deprecated

27.57.3.6 static bool gdcm::CompositeNetworkFunctions::CStore (const char * *remote*, uint16_t *portno*, const Directory::FilenamesType & *filenames*, const char * *aetitle* = `NULL`, const char * *call* = `NULL`) [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

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

- gdcmCompositeNetworkFunctions.h

27.58 gdcm::ConstCharWrapper Class Reference

Do not use me.

`#include <gdcmConstCharWrapper.h>`

Public Member Functions

- ConstCharWrapper (const char *i=0)
- operator const char * () const

27.58.1 Detailed Description

Do not use me.

27.58.2 Constructor & Destructor Documentation

27.58.2.1 gdcm::ConstCharWrapper::ConstCharWrapper (const char * *i* = `0`) [inline]

27.58.3 Member Function Documentation

27.58.3.1 gdcm::ConstCharWrapper::operator const char * ( ) const [inline]

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

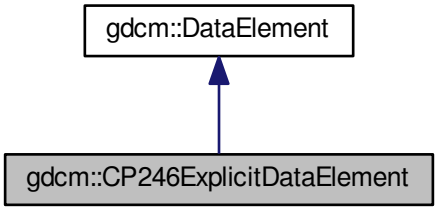
- gdcmConstCharWrapper.h

27.59 gdcm::CP246ExplicitDataElement Class Reference

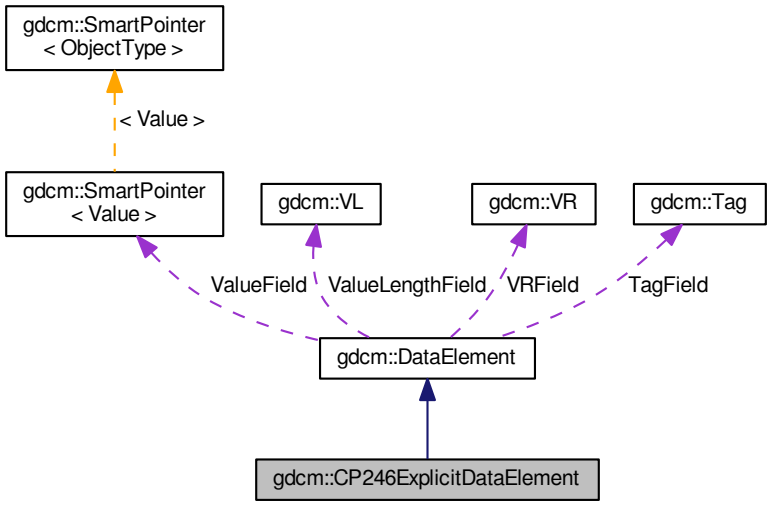
Class to read/write a DataElement as CP246Explicit Data Element.

`#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)

Additional Inherited Members

27.59.1 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

27.59.2 Member Function Documentation

27.59.2.1 VL gdcm::CP246ExplicitDataElement::GetLength ( ) const

27.59.2.2 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::Read (std::istream & *is*)

27.59.2.3 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::ReadPreValue (std::istream & *is*)

27.59.2.4 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::ReadValue (std::istream & *is*, bool *readvalues* = `true`)

27.59.2.5 template<typename TSwap > std::istream& gdcm::CP246ExplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*)

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

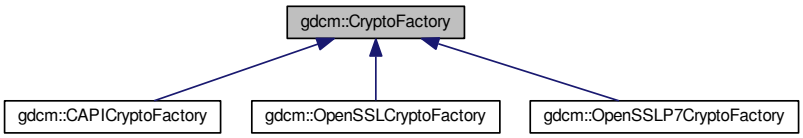
- gdcmCP246ExplicitDataElement.h

27.60 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 (CryptoLib id)
- CryptoFactory ()
- ~CryptoFactory ()

27.60.1 Detailed Description

Class to do handle the crypto factory.

GDCM needs to access in a platform independant 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.

27.60.2 Member Enumeration Documentation

27.60.2.1 enum gdcm::CryptoFactory::CryptoLib

Enumerator

DEFAULT
OPENSSL
CAPI
OPENSSLP7

27.60.3 Constructor & Destructor Documentation

27.60.3.1 gdcm::CryptoFactory::CryptoFactory (CryptoLib *id*) [inline], [protected]

27.60.3.2 gdcm::CryptoFactory::CryptoFactory ( ) [inline], [protected]

27.60.3.3 gdcm::CryptoFactory::~CryptoFactory ( ) [inline], [protected]

27.60.4 Member Function Documentation

27.60.4.1 virtual CryptographicMessageSyntax* gdcm::CryptoFactory::CreateCMSProvider ( ) [pure virtual]

Implemented in gdcm::OpenSSLCryptoFactory, gdcm::OpenSSLP7CryptoFactory, and gdcm::CAPICryptoFactory.

27.60.4.2 static CryptoFactory* gdcm::CryptoFactory::GetFactoryInstance (CryptoLib *id* = DEFAULT) [static]

Examples:

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

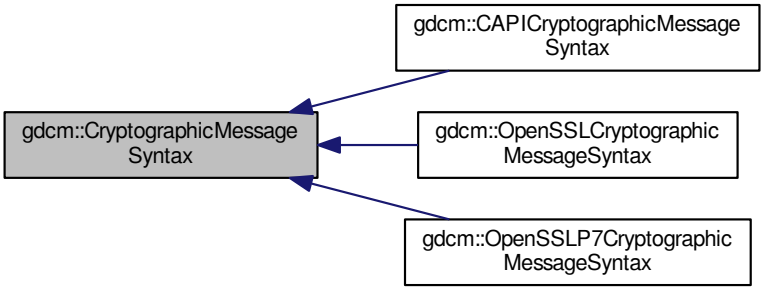
The documentation for this class was generated from the following file:

- gdcmCryptoFactory.h

27.61 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 ()
- virtual ~CryptographicMessageSyntax ()
- 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
- 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

27.61.1 Member Enumeration Documentation

27.61.1.1 enum gdcm::CryptographicMessageSyntax::CipherTypes

Enumerator

DES3_CIPHER

AES128_CIPHER

AES192_CIPHER

AES256_CIPHER

27.61.2 Constructor & Destructor Documentation

27.61.2.1 gdcm::CryptographicMessageSyntax::CryptographicMessageSyntax ( ) [inline]

27.61.2.2 virtual gdcm::CryptographicMessageSyntax::~CryptographicMessageSyntax ( ) [inline], [virtual]

27.61.3 Member Function Documentation

27.61.3.1 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::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

27.61.3.2 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::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

27.61.3.3 virtual CipherTypes gdcm::CryptographicMessageSyntax::GetCipherType ( ) const [pure virtual]

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

27.61.3.4 virtual bool gdcm::CryptographicMessageSyntax::ParseCertificateFile (const char * *filename*) [pure virtual]

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

27.61.3.5 virtual bool gdcm::CryptographicMessageSyntax::ParseKeyFile (const char * *filename*) [pure virtual]

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

27.61.3.6 virtual void gdcm::CryptographicMessageSyntax::SetCipherType (CipherTypes *type*) [pure virtual]

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

27.61.3.7 virtual bool gdcm::CryptographicMessageSyntax::SetPassword (const char * *pass*, size_t *passLen*) [pure virtual]

Implemented in gdcm::OpenSSLP7CryptographicMessageSyntax, gdcm::CAPICryptographicMessageSyntax, and gdcm::OpenSSLCryptographicMessageSyntax.

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

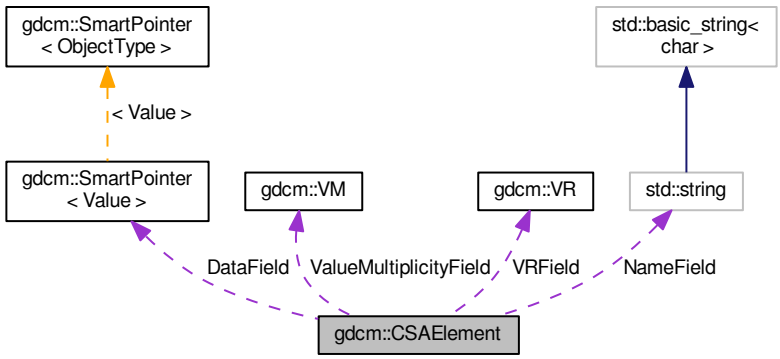
- gdcmCryptographicMessageSyntax.h

27.62 gdcm::CSAElement Class Reference

Class to represent a CSA Element.

`#include <gdcmCSAElement.h>`

Collaboration diagram for gdcm::CSAElement:



Public Member Functions

- CSAElement (unsigned int kf=0)
- CSAElement (const CSAElement &_val)
- 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 const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):

- Value & GetValue ()
- 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)
- 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)

27.62.1 Detailed Description

Class to represent a CSA Element.

See also

CSAHeader

Examples:

csa2img.cxx, and MrProtocol.cxx.

27.62.2 Member Typedef Documentation

27.62.2.1 typedef SmartPointer<Value> gdcm::CSAElement::DataPtr [protected]

27.62.3 Constructor & Destructor Documentation

27.62.3.1 gdcm::CSAElement::CSAElement (unsigned int *kf* = 0) [inline]

27.62.3.2 gdcm::CSAElement::CSAElement (const CSAElement & *_val*) [inline]

27.62.4 Member Function Documentation

27.62.4.1 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:

MrProtocol.cxx.

27.62.4.2 unsigned int gdcm::CSAElement::GetKey ( ) const [inline]

Set/Get Key.

Referenced by operator<().

27.62.4.3 const char* gdcm::CSAElement::GetName ( ) const [inline]

Set/Get Name.

27.62.4.4 unsigned int gdcm::CSAElement::GetNoOfItems ( ) const [inline]

Set/Get NoOfItems.

27.62.4.5 unsigned int gdcm::CSAElement::GetSyngoDT ( ) const [inline]

Set/Get SyngoDT.

27.62.4.6 Value const& gdcm::CSAElement::GetValue ( ) const [inline]

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples:

csa2img.cxx.

27.62.4.7 Value& gdcm::CSAElement::GetValue ( ) [inline]

27.62.4.8 const VM& gdcm::CSAElement::GetVM ( ) const [inline]

Set/Get VM.

27.62.4.9 VR const& gdcm::CSAElement::GetVR ( ) const [inline]

Set/Get VR.

27.62.4.10 bool gdcm::CSAElement::IsEmpty ( ) const [inline]

Check if CSA Element is empty.

Examples:

csa2img.cxx.

27.62.4.11 bool gdcm::CSAElement::operator< (const CSAElement & *de*) const [inline]

References GetKey().

27.62.4.12 CSAElement& gdcm::CSAElement::operator= (const CSAElement & *de*) [inline]

References DataField, KeyField, NameField, NoOfItemsField, SyngoDTField, ValueMultiplicityField, and VRField.

27.62.4.13 bool gdcm::CSAElement::operator== (const CSAElement & *de*) const [inline]

References KeyField, NameField, SyngoDTField, ValueMultiplicityField, and VRField.

27.62.4.14 void gdcm::CSAElement::SetByteValue (const char * *array*, VL *length*) [inline]

Set.

27.62.4.15 void gdcm::CSAElement::SetKey (unsigned int *key*) [inline]

27.62.4.16 void gdcm::CSAElement::SetName (const char * *name*) [inline]

27.62.4.17 void gdcm::CSAElement::SetNoOfItems (unsigned int *items*) [inline]

27.62.4.18 void gdcm::CSAElement::SetSyngoDT (unsigned int *syngodt*) [inline]

27.62.4.19 void gdcm::CSAElement::SetValue (Value const & *vl*) [inline]

27.62.4.20 void gdcm::CSAElement::SetVM (const VM & *vm*) [inline]

27.62.4.21 void gdcm::CSAElement::SetVR (VR const & *vr*) [inline]

27.62.5 Friends And Related Function Documentation

27.62.5.1 std::ostream& operator<< (std::ostream & *os,* const CSAElement & *val*) [friend]

27.62.6 Member Data Documentation

27.62.6.1 DataPtr gdcm::CSAElement::DataField [protected]

Referenced by gdcm::operator<<(), and operator=().

27.62.6.2 unsigned int gdcm::CSAElement::KeyField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

27.62.6.3 std::string gdcm::CSAElement::NameField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

27.62.6.4 unsigned int gdcm::CSAElement::NoOfItemsField [protected]

Referenced by gdcm::operator<<(), and operator=().

27.62.6.5 unsigned int gdcm::CSAElement::SyngoDTField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

27.62.6.6 VM gdcm::CSAElement::ValueMultiplicityField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

27.62.6.7 VR gdcm::CSAElement::VRField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

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

- gdcmCSAElement.h

27.63 gdcm::CSAHeader Class Reference

Class for CSAHeader.

`#include <gdcmCSAHeader.h>`

Public Types

- enum CSAHeaderType {
  UNKNOWN = 0,
  SV10,
  NOMAGIC,
  DATASET_FORMAT,
  INTERFILE,
  ZEROED_OUT }

  Divers format of CSAHeader as found 'in the wild'.

Public Member Functions

- CSAHeader ()
- ~CSAHeader ()
- 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 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)
- template<typename TSwap >
  std::istream & Read (std::istream &is)
- template<typename TSwap >
  const std::ostream & Write (std::ostream &os) const

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)

27.63.1 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/NO↩ MAGIC 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:

csa2img.cxx, and MrProtocol.cxx.

27.63.2 Member Enumeration Documentation

27.63.2.1 enum gdcm::CSAHeader::CSAHeaderType

Divers format of CSAHeader as found 'in the wild'.

Enumerator

UNKNOWN

SV10

NOMAGIC

DATASET_FORMAT

INTERFILE

ZEROED_OUT

27.63.3 Constructor & Destructor Documentation

27.63.3.1 gdcm::CSAHeader::CSAHeader ( ) [inline]

27.63.3.2 gdcm::CSAHeader::~CSAHeader ( ) [inline]

27.63.4 Member Function Documentation

27.63.4.1 bool gdcm::CSAHeader::FindCSAElementByName (const char * *name*)

Return true if the CSA element matching 'name' is found or not

Warning

Case Sensitive

Examples:

csa2img.cxx, and MrProtocol.cxx.

27.63.4.2 static const PrivateTag& gdcm::CSAHeader::GetCSADataInfo ( ) [static]

Return the private tag used by SIEMENS to store the CSA Data Info This is: PrivateTag(0x0029,0x0010,"SIEMENS CSA NON-IMAGE");

27.63.4.3 const CSAElement& gdcm::CSAHeader::GetCSAEEnd ( ) const [protected]

27.63.4.4 const CSAElement& gdcm::CSAHeader::GetCSAElementByName (const char * *name*)

Return the CSAElement corresponding to name 'name'

Warning

Case Sensitive

Examples:

csa2img.cxx, and MrProtocol.cxx.

27.63.4.5 static const PrivateTag& gdcm::CSAHeader::GetCSAImageHeaderInfoTag ( ) [static]

Return the private tag used by SIEMENS to store the CSA Image Header This is: PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER");

Examples:

csa2img.cxx, and PublicDict.cxx.

27.63.4.6 static const PrivateTag& gdcm::CSAHeader::GetCSASeriesHeaderInfoTag ( ) [static]

Return the private tag used by SIEMENS to store the CSA Series Header This is: PrivateTag(0x0029,0x0020,"SIEM↩ ENS CSA HEADER");

Examples:

MrProtocol.cxx.

27.63.4.7 const DataSet& gdcm::CSAHeader::GetDataSet ( ) const [inline]

Return the DataSet output (use only if Format == DATASET_FORMAT)

27.63.4.8 CSAHeaderType gdcm::CSAHeader::GetFormat ( ) const

return the format of the CSAHeader SV10 and NOMAGIC are equivalent.

27.63.4.9 const char* gdcm::CSAHeader::GetInterfile ( ) const [inline]

Return the string output (use only if Format == Interfile)

27.63.4.10 bool gdcm::CSAHeader::LoadFromDataElement (DataElement const & *de*)

Decode the CSAHeader from element 'de'.

Examples:

csa2img.cxx, and MrProtocol.cxx.

27.63.4.11 void gdcm::CSAHeader::Print (std::ostream & *os*) const

Print the CSAHeader (use only if Format == SV10 or NOMAGIC)

Examples:

csa2img.cxx.

Referenced by gdcm::operator<<().

27.63.4.12 template<typename TSwap > std::istream& gdcm::CSAHeader::Read (std::istream & *is*)

27.63.4.13 template<typename TSwap > const std::ostream& gdcm::CSAHeader::Write (std::ostream & *os*) const

27.63.5 Friends And Related Function Documentation

27.63.5.1 std::ostream& operator<< (std::ostream & *_os,* const CSAHeader & *d*) [friend]

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

- gdcmCSAHeader.h

27.64 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 ()
- void AddCSAHeaderDictEntry (const CSAHeaderDictEntry &de)
- ConstIterator Begin () const
- ConstIterator End () const
- const CSAHeaderDictEntry & GetCSAHeaderDictEntry (const char *name) const
- bool IsEmpty () const

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const CSAHeaderDict &_val)

27.64.1 Detailed Description

Class to represent a map of CSAHeaderDictEntry.

Examples:

MrProtocol.cxx.

27.64.2 Member Typedef Documentation

27.64.2.1 typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator

27.64.2.2 typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator

27.64.2.3 typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry

27.64.3 Constructor & Destructor Documentation

27.64.3.1 gdcm::CSAHeaderDict::CSAHeaderDict ( ) [inline]

27.64.4 Member Function Documentation

27.64.4.1 void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (const CSAHeaderDictEntry & *de*) [inline]

27.64.4.2 ConstIterator gdcm::CSAHeaderDict::Begin ( ) const [inline]

27.64.4.3 ConstIterator gdcm::CSAHeaderDict::End ( ) const [inline]

27.64.4.4 const CSAHeaderDictEntry& gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (const char * *name*) const [inline]

Examples:

MrProtocol.cxx.

27.64.4.5 bool gdcm::CSAHeaderDict::IsEmpty ( ) const [inline]

27.64.4.6 void gdcm::CSAHeaderDict::LoadDefault ( ) [protected]

27.64.5 Friends And Related Function Documentation

27.64.5.1 friend class Dicts [friend]

27.64.5.2 std::ostream& operator<< (std::ostream & *_os,* const CSAHeaderDict & *_val*) [friend]

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

- gdcmCSAHeaderDict.h

27.65 gdcm::CSAHeaderDictEntry Class Reference

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.

`#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)

27.65.1 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.

27.65.2 Constructor & Destructor Documentation

27.65.2.1 gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry (const char * *name* = " ", VR const & *vr* = VR::INVALID, VM const & *vm* = VM::VM0, const char * *desc* = " ") [inline]

27.65.3 Member Function Documentation

27.65.3.1 const char* gdcm::CSAHeaderDictEntry::GetDescription ( ) const [inline]

Set/Get Description.

27.65.3.2 const char* gdcm::CSAHeaderDictEntry::GetName ( ) const [inline]

Set/Get Name.

Referenced by operator<().

27.65.3.3 const VM& gdcm::CSAHeaderDictEntry::GetVM ( ) const [inline]

Set/Get VM.

27.65.3.4 const VR& gdcm::CSAHeaderDictEntry::GetVR ( ) const [inline]

Set/Get VR.

27.65.3.5 bool gdcm::CSAHeaderDictEntry::operator< (const CSAHeaderDictEntry & *entry*) const [inline]

References GetName().

27.65.3.6 void gdcm::CSAHeaderDictEntry::SetDescription (const char * *desc*) [inline]

27.65.3.7 void gdcm::CSAHeaderDictEntry::SetName (const char * *name*) [inline]

27.65.3.8 void gdcm::CSAHeaderDictEntry::SetVM (VM const & *vm*) [inline]

27.65.3.9 void gdcm::CSAHeaderDictEntry::SetVR (const VR & *vr*) [inline]

27.65.4 Friends And Related Function Documentation

27.65.4.1 std::ostream& operator<< (std::ostream & *_os*, const CSAHeaderDictEntry & *_val*) [friend]

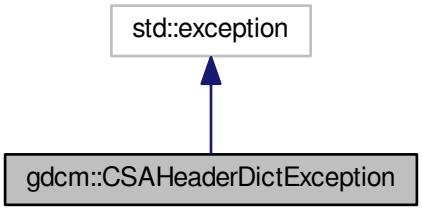
The documentation for this class was generated from the following file:

- gdcmCSAHeaderDictEntry.h

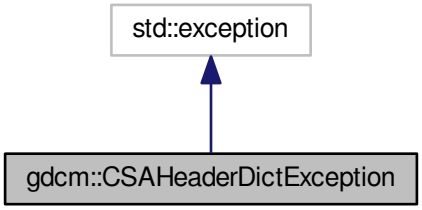
27.66 gdcm::CSAHeaderDictException Class Reference

`#include <gdcmCSAHeaderDict.h>`

Inheritance diagram for gdcm::CSAHeaderDictException:



Collaboration diagram for gdcm::CSAHeaderDictException:



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

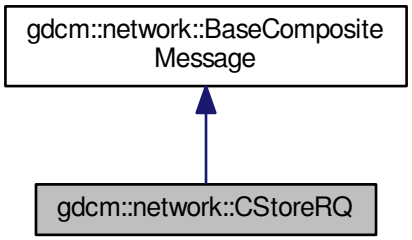
- gdcmCSAHeaderDict.h

27.67 gdcm::network::CStoreRQ Class Reference

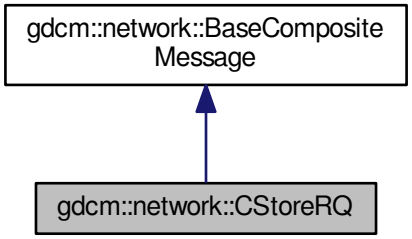
CStoreRQ this file defines the messages for the cecho action.

`#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)

27.67.1 Detailed Description

CStoreRQ this file defines the messages for the cecho action.

27.67.2 Member Function Documentation

27.67.2.1 std::vector<PresentationDataValue> gdcm::network::CStoreRQ::ConstructPDV (const ULConnection & *inConnection,* const File & *file*)

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

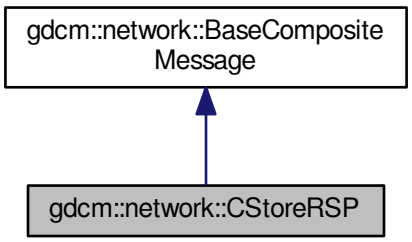
- gdcmCStoreMessages.h

27.68 gdcm::network::CStoreRSP Class Reference

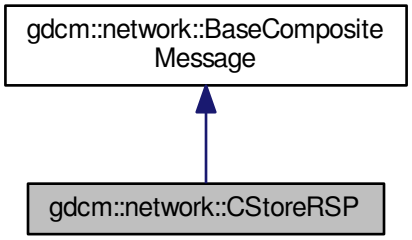
CStoreRSP this file defines the messages for the cecho action.

```
#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)

27.68.1 Detailed Description

CStoreRSP this file defines the messages for the cecho action.

27.68.2 Member Function Documentation

27.68.2.1 std::vector<PresentationDataValue> gdcm::network::CStoreRSP::ConstructPDV (const DataSet * *inDataSet,* const BasePDU * *inPC*)

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

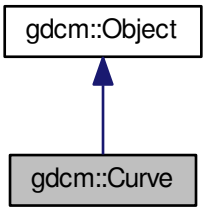
- gdcmCStoreMessages.h

27.69 gdcm::Curve Class Reference

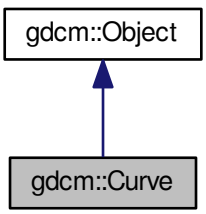
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

`#include <gdcmCurve.h>`

Inheritance diagram for gdcm::Curve:



Collaboration diagram for gdcm::Curve:



Public Member Functions

- Curve ()
- Curve (Curve const &ov)
- ~Curve ()
- 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
- 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)

Static Public Member Functions

- static unsigned int GetNumberOfCurves (DataSet const &ds)

Additional Inherited Members

27.69.1 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

27.69.2 Constructor & Destructor Documentation

27.69.2.1 gdcm::Curve::Curve ( )

27.69.2.2 gdcm::Curve::~Curve ( )

27.69.2.3 gdcm::Curve::Curve (Curve const & *ov*)

27.69.3 Member Function Documentation

27.69.3.1 void gdcm::Curve::Decode (std::istream & *is,* std::ostream & *os*)

27.69.3.2 void gdcm::Curve::GetAsPoints (float * *array*) const

27.69.3.3 std::vector<unsigned short> const& gdcm::Curve::GetCurveDataDescriptor ( ) const

27.69.3.4 unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const

27.69.3.5 unsigned short gdcm::Curve::GetDimensions ( ) const

27.69.3.6 unsigned short gdcm::Curve::GetGroup ( ) const

27.69.3.7 static unsigned int gdcm::Curve::GetNumberOfCurves (DataSet const & *ds*) [static]

27.69.3.8 unsigned short gdcm::Curve::GetNumberOfPoints ( ) const

27.69.3.9 const char* gdcm::Curve::GetTypeOfData ( ) const

27.69.3.10 const char* gdcm::Curve::GetTypeOfDataDescription ( ) const

27.69.3.11 bool gdcm::Curve::IsEmpty ( ) const

27.69.3.12 void gdcm::Curve::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Object.

27.69.3.13 void gdcm::Curve::SetCoordinateStartValue (unsigned short *v*)

27.69.3.14 void gdcm::Curve::SetCoordinateStepValue (unsigned short *v*)

27.69.3.15 void gdcm::Curve::SetCurve (const char * *array,* unsigned int *length*)

27.69.3.16 void gdcm::Curve::SetCurveDataDescriptor (const uint16_t * *values,* size_t *num*)

27.69.3.17 void gdcm::Curve::SetCurveDescription (const char * *curvedescription*)

27.69.3.18 void gdcm::Curve::SetDataValueRepresentation (unsigned short *datavaluerepresentation*)

27.69.3.19 void gdcm::Curve::SetDimensions (unsigned short *dimensions*)

27.69.3.20 void gdcm::Curve::SetGroup (unsigned short *group*)

27.69.3.21 void gdcm::Curve::SetNumberOfPoints (unsigned short *numberofpoints*)

27.69.3.22 void gdcm::Curve::SetTypeOfData (const char * *typeofdata*)

27.69.3.23 void gdcm::Curve::Update (const DataElement & *de*)

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

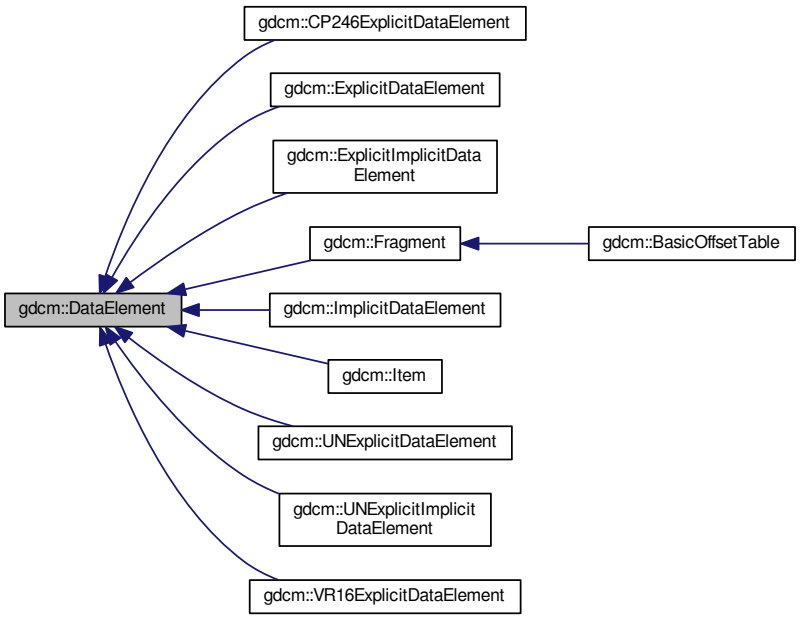
- gdcmCurve.h

27.70 gdcm::DataElement Class Reference

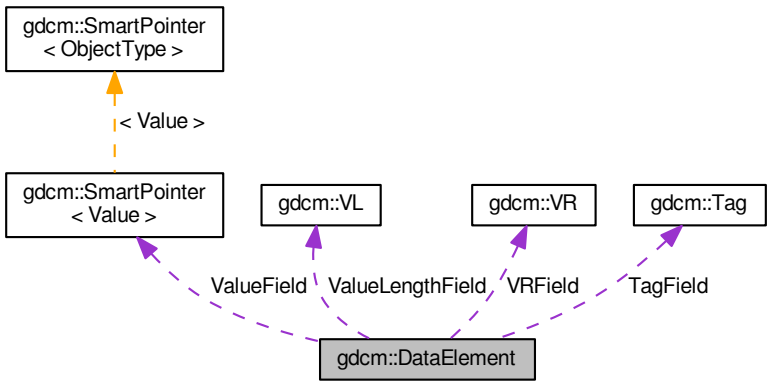
Class to represent a Data Element either Implicit or Explicit.

`#include <gdcmDataElement.h>`

Inheritance diagram for gdcm::DataElement:



Collaboration diagram for gdcm::DataElement:



Public Member Functions

- DataElement (const Tag &t=Tag(0), const VL &vl=0, const VR &vr=VR::INVALID)
- DataElement (const DataElement &_val)
- 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
- const SequenceOfFragments * GetSequenceOfFragments () const
- SequenceOfFragments * GetSequenceOfFragments ()
- const Tag & GetTag () const

  Get Tag.
- Tag & GetTag ()
- Value const & GetValue () const

  Set/Get Value (bytes array, SQ of items, SQ of fragments):
- Value & GetValue ()
- SmartPointer< SequenceOfItems > GetValueAsSQ () const
- const VL & GetVL () const

  Get VL.
- VL & GetVL ()
- 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 &de)
- 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)

27.70.1 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:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImage↩ HeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncrypted↩ Content.cxx, ExtractIconFromFile.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_↩ Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, Gen↩ FakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, Get↩ SequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and Stream↩ ImageReaderTest.cxx.

27.70.2 Member Typedef Documentation

27.70.2.1 typedef SmartPointer<Value> gdcm::DataElement::ValuePtr [protected]

27.70.3 Constructor & Destructor Documentation

27.70.3.1 gdcm::DataElement::DataElement (const Tag & *t* = Tag (0), const VL & *vl* = 0, const VR & *vr* = VR::INVALID) [inline]

27.70.3.2 gdcm::DataElement::DataElement (const DataElement & *_val*) [inline]

27.70.4 Member Function Documentation

27.70.4.1 void gdcm::DataElement::Clear ( ) [inline]

Clear Data Element (make Value empty and invalidate Tag & VR)

References gdcm::VR::INVALID.

Referenced by gdcm::Item::Clear().

27.70.4.2 void gdcm::DataElement::Empty ( ) [inline]

Make Data Element empty (no Value)

27.70.4.3 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, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetSubSequenceData.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::operator<<(), gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

27.70.4.4 template<typename TDE > VL gdcm::DataElement::GetLength ( ) const [inline]

27.70.4.5 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:

FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and GetJPEGSamplePrecision.cxx.

27.70.4.6 SequenceOfFragments* gdcm::DataElement::GetSequenceOfFragments ( )

27.70.4.7 const Tag& gdcm::DataElement::GetTag ( ) const [inline]

Get Tag.

Examples:

DumpGEMSMovieGroup.cxx, DuplicatePCDE.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

Referenced by gdcm::CommandDataSet::Insert(), gdcm::FileMetaInformation::Insert(), gdcm::DataSet::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 >::SetFromDataElement().

27.70.4.8 Tag& gdcm::DataElement::GetTag ( ) [inline]

27.70.4.9 Value const& gdcm::DataElement::GetValue ( ) const [inline]

Set/Get Value (bytes array, SQ of items, SQ of fragments):

Examples:

ReadAndDumpDICOMDIR.cxx.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), and gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement().

27.70.4.10 Value& gdcm::DataElement::GetValue ( ) [inline]

27.70.4.11 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 simpliest 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, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

27.70.4.12 const VL& gdcm::DataElement::GetVL ( ) const [inline]

Get VL.

Referenced by gdcm::DataSet::InsertDataElement(), gdcm::SequenceOfItems::Read(), and gdcm::SequenceOfFragments::ReadValue().

27.70.4.13 VL& gdcm::DataElement::GetVL ( ) [inline]

27.70.4.14 VR const& gdcm::DataElement::GetVR ( ) const [inline]

Get VR do not set VR::SQ on bytevalue data element

Examples:

DuplicatePCDE.cxx, and GenFakeIdentifyFile.cxx.

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement().

27.70.4.15 bool gdcm::DataElement::IsEmpty ( ) const [inline]

Check if Data Element is empty.

Examples:

DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, ELSCINT1WaveToText.cxx, FixJAIBugJPEGLS.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

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 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

27.70.4.16 bool gdcm::DataElement::IsUndefinedLength ( ) const [inline]

return if Value Length if of undefined length

27.70.4.17 bool gdcm::DataElement::operator< (const DataElement & *de*) const [inline]

References GetTag().

27.70.4.18 DataElement& gdcm::DataElement::operator= (const DataElement & *de*) [inline]

References TagField, ValueField, ValueLengthField, and VRField.

27.70.4.19 bool gdcm::DataElement::operator== (const DataElement & *de*) const [inline]

References TagField, ValueField, ValueLengthField, and VRField.

27.70.4.20 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::Read (std::istream & *is*) [inline]

27.70.4.21 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadOrSkip (std::istream & *is*, std::set< Tag > const & *skiptags*) [inline]

27.70.4.22 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadPreValue (std::istream & *is*, std::set< Tag > const & *skiptags*) [inline]

27.70.4.23 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadValue (std::istream & *is*, std::set< Tag > const & *skiptags*) [inline]

27.70.4.24 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadValueWithLength (std::istream & *is*, VL & *length*, std::set< Tag > const & *skiptags*) [inline]

27.70.4.25 template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadWithLength (std::istream & *is*, VL & *length*) [inline]

27.70.4.26 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:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, 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, iU22tomultisc.cxx, NewSequence.cs, and StreamImageReaderTest.cxx.

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement().

27.70.4.27 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.

27.70.4.28 void gdcm::DataElement::SetValue (Value const & *vl*) [inline]

Warning

you need to set the ValueLengthField explicitely

Examples:

DuplicatePCDE.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

References gdcm::Value::GetLength().

27.70.4.29 void gdcm::DataElement::SetValueFieldLength (VL *vl,* bool *readvalues*) [protected]

27.70.4.30 void gdcm::DataElement::SetVL (const VL & *vl*) [inline]

Set VL Use with cautious (need to match Part 6), advanced user only

See also

SetByteValue

27.70.4.31 void gdcm::DataElement::SetVLToUndefined ( )

Examples:

Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, and NewSequence.cs.

27.70.4.32 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, iU22tomultisc.cxx, NewSequence.cs, and StreamImageReaderTest.cxx.

References gdcm::VR::IsVRFile().

Referenced by gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement().

27.70.4.33 template<typename TDE , typename TSwap > const std::ostream& gdcm::DataElement::Write (std::ostream & *os*) const [inline]

27.70.5 Friends And Related Function Documentation

27.70.5.1 std::ostream& operator<< (std::ostream & *_os,* const DataElement & *_val*) [friend]

27.70.6 Member Data Documentation

27.70.6.1 Tag gdcm::DataElement::TagField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

27.70.6.2 ValuePtr gdcm::DataElement::ValueField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

27.70.6.3 VL gdcm::DataElement::ValueLengthField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

27.70.6.4 VR gdcm::DataElement::VRField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

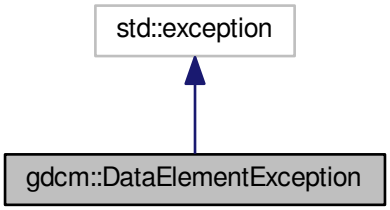
The documentation for this class was generated from the following file:

- gdcmDataElement.h

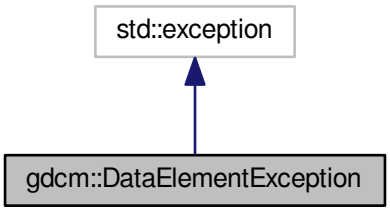
27.71 gdcm::DataElementException Class Reference

`#include <gdcmDataSet.h>`

Inheritance diagram for gdcm::DataElementException:



Collaboration diagram for gdcm::DataElementException:



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

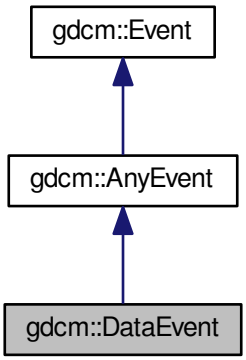
- gdcmDataSet.h

27.72 gdcm::DataEvent Class Reference

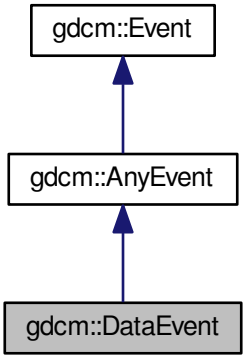
DataEvent.

`#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=0, size_t len=0)
- DataEvent (const Self &s)
- virtual ~DataEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- const char * GetData () const
- size_t GetDataLength () const
- virtual const char * GetEventName () const
- virtual ::gdcm::Event * MakeObject () const
- void SetData (const char *bytes, size_t len)

27.72.1 Detailed Description

DataEvent.

27.72.2 Member Typedef Documentation

27.72.2.1 typedef DataEvent gdcm::DataEvent::Self

27.72.2.2 typedef AnyEvent gdcm::DataEvent::Superclass

27.72.3 Constructor & Destructor Documentation

27.72.3.1 gdcm::DataEvent::DataEvent (const char * *bytes* = 0, size_t *len* = 0) [inline]

27.72.3.2 virtual gdcm::DataEvent::~DataEvent ( ) [inline], [virtual]

27.72.3.3 gdcm::DataEvent::DataEvent (const Self & *s*) [inline]

27.72.4 Member Function Documentation

27.72.4.1 virtual bool gdcm::DataEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

27.72.4.2 const char* gdcm::DataEvent::GetData ( ) const [inline]

27.72.4.3 size_t gdcm::DataEvent::GetDataLength ( ) const [inline]

27.72.4.4 virtual const char* gdcm::DataEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

27.72.4.5 virtual ::gdcm::Event* gdcm::DataEvent::MakeObject ( ) const [inline], [virtual]

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

Implements gdcm::Event.

27.72.4.6 void gdcm::DataEvent::SetData (const char * *bytes*, size_t *len*) [inline]

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

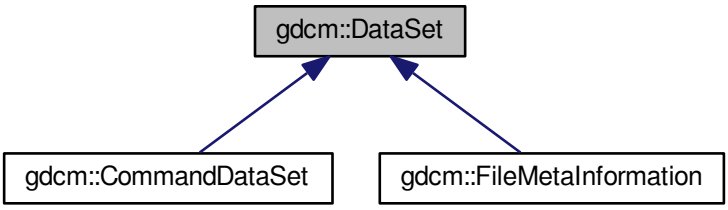
- gdcmDataEvent.h

27.73 gdcm::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object.

`#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

- ConstIterator Begin () const
- Iterator Begin ()
- void Clear ()
- template<typename TDE > unsigned int ComputeGroupLength (Tag const &tag) const
- ConstIterator End () const
- Iterator End ()
- 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 Tag &t) const
- const DataElement & GetDataElement (const PrivateTag &t) const

  Return the dataelement.
- const DataElementSet & GetDES () const
- DataElementSet & GetDES ()
- template<typename TDE > VL GetLength () const
- MediaStorage GetMediaStorage () const
- std::string GetPrivateCreator (const Tag &t) const

  Return the private creator of the private tag 't':
- 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 &val)
- 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)

27.73.1 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, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImage↩HeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncrypted↩Content.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSample↩Precision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDump↩DICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, SortImage.cxx, StreamImageReaderTest.cxx, and VolumeSorter.cxx.

27.73.2 Member Typedef Documentation

27.73.2.1 typedef DataElementSet::const_iterator gdcm::DataSet::ConstIterator

27.73.2.2 typedef std::set<DataElement> gdcm::DataSet::DataElementSet

27.73.2.3 typedef DataElementSet::iterator gdcm::DataSet::Iterator

27.73.2.4 typedef DataElementSet::size_type gdcm::DataSet::SizeType

27.73.3 Member Function Documentation

27.73.3.1 ConstIterator gdcm::DataSet::Begin ( ) const [inline]

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

27.73.3.2 Iterator gdcm::DataSet::Begin ( ) [inline]

27.73.3.3 void gdcm::DataSet::Clear ( ) [inline]

Referenced by gdcm::Item::Read().

27.73.3.4 Tag gdcm::DataSet::ComputeDataElement (const PrivateTag & *t*) const [protected]

27.73.3.5 template<typename TDE > unsigned int gdcm::DataSet::ComputeGroupLength (Tag const & *tag*) const [inline]

References gdcm::Tag::GetElement(), and gdcm::Tag::GetGroup().

27.73.3.6 ConstIterator gdcm::DataSet::End ( ) const [inline]

Examples:

DiffFile.cxx, DumpGEMSMovieGroup.cxx, and DuplicatePCDE.cxx.

27.73.3.7 Iterator gdcm::DataSet::End ( ) [inline]

27.73.3.8 bool gdcm::DataSet::FindDataElement (const PrivateTag & *t*) const

Look up if private tag 't' is present in the dataset:

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, csa2img.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

27.73.3.9 bool gdcm::DataSet::FindDataElement (const Tag & *t*) const [inline]

27.73.3.10 const DataElement& gdcm::DataSet::FindNextDataElement (const Tag & *t*) const [inline]

Examples:

DuplicatePCDE.cxx.

27.73.3.11 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, csa2img.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

Referenced by 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, TV↩
R, TVM >::SetFromDataSet(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcm::↩
Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

27.73.3.12 const DataElement& gdcm::DataSet::GetDataElement (const PrivateTag & *t*) const

Return the dataelement.

27.73.3.13 const DataElement& gdcm::DataSet::GetDEEnd ( ) const [protected]

27.73.3.14 const DataElementSet& gdcm::DataSet::GetDES ( ) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

27.73.3.15 DataElementSet& gdcm::DataSet::GetDES ( ) [inline]

27.73.3.16 template<typename TDE > VL gdcm::DataSet::GetLength ( ) const [inline]

27.73.3.17 MediaStorage gdcm::DataSet::GetMediaStorage ( ) const

27.73.3.18 std::string gdcm::DataSet::GetPrivateCreator (const Tag & *t*) const

Return the private creator of the private tag 't':

Examples:

DuplicatePCDE.cxx.

27.73.3.19 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, DuplicatePCDE.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image↩
_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, NewSequence.cs, and StreamImageReaderTest.cxx.

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

27.73.3.20 void gdcm::DataSet::InsertDataElement (const DataElement & *de*) [inline], [protected]

References gdcmWarningMacro, gdcm::Value::GetLength(), gdcm::DataElement::GetValue(), gdcm::DataElement::↩
GetVL(), and gdcm::DataElement::IsEmpty().

27.73.3.21 bool gdcm::DataSet::IsEmpty ( ) const [inline]

Returns if the dataset is empty.

Referenced by gdcm::Item::Read().

27.73.3.22 const DataElement& gdcm::DataSet::operator() (uint16_t *group*, uint16_t *element*) const [inline]

27.73.3.23 DataSet& gdcm::DataSet::operator= (DataSet const & *val*) [inline]

27.73.3.24 const DataElement& gdcm::DataSet::operator[] (const Tag & *t*) const [inline]

27.73.3.25 void gdcm::DataSet::Print (std::ostream & *os*, std::string const & *indent* = " ") const [inline]

Referenced by gdcm::operator<<().

27.73.3.26 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::Read (std::istream & *is*)

27.73.3.27 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadNested (std::istream & *is*)

27.73.3.28 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedPrivateTags (std::istream & *is*, const std::set< PrivateTag > & *tags*, bool *readvalues* = true)

27.73.3.29 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedPrivateTagsWithLength (std::istream & *is*, const std::set< PrivateTag > & *tags*, VL & *length*, bool *readvalues* = true)

27.73.3.30 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTags (std::istream & *is*, const std::set< Tag > & *tags*, bool *readvalues* = true)

27.73.3.31 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTagsWithLength (std::istream & *is*, const std::set< Tag > & *tags*, VL & *length*, bool *readvalues* = true)

27.73.3.32 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTag (std::istream & *is*, const Tag & *t*, std::set< Tag > const & *skiptags*)

27.73.3.33 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTagWithLength (std::istream & *is*, const Tag & *t*, std::set< Tag > const & *skiptags*, VL & *length*)

27.73.3.34 template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadWithLength (std::istream & *is*, VL & *length*)

27.73.3.35 SizeType gdcm::DataSet::Remove (const Tag & *tag*) [inline]

Completely remove a dataelement from the dataset.

Examples:

ClinicalTrialIdentificationWorkflow.cs, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, pmsct_rgb1.cxx, ReformatFile.cs, rle2img.cxx, and StandardizeFiles.cs.

27.73.3.36 void gdcm::DataSet::Replace (const DataElement & *de*) [inline]

Replace a dataelement with another one.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateFakeRTDOSE.cxx, FixBrokenJ2K.cxx, FixJAI↩ BugJPEGLS.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, Large↩ VRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

27.73.3.37 void gdcm::DataSet::ReplaceEmpty (const DataElement & *de*) [inline]

Only replace a DICOM attribute when it is missing or empty.

27.73.3.38 SizeType gdcm::DataSet::Size ( ) const [inline]

Examples:

DumpGEMSMovieGroup.cxx.

Referenced by gdcm::SequenceOfItems::Read().

27.73.3.39 template<typename TDE , typename TSwap > std::ostream const& gdcm::DataSet::Write (std::ostream & *os*) const

27.73.4 Friends And Related Function Documentation

27.73.4.1 friend class CSAHeader [friend]

27.73.4.2 std::ostream& operator<< (std::ostream & *_os,* const DataSet & *val*) [friend]

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

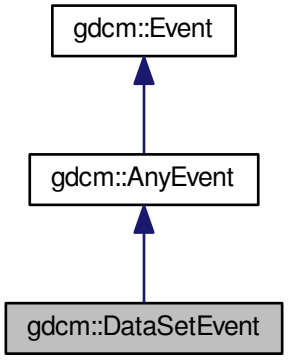
- gdcmDataSet.h

27.74 gdcm::DataSetEvent Class Reference

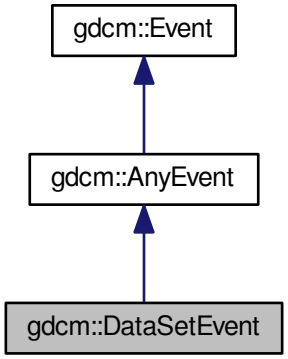
DataSetEvent Special type of event triggered during the DataSet store/move process.

`#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 (DataSet const *ds=NULL)
- DataSetEvent (const Self &s)

- virtual ~DataSetEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- DataSet const & GetDataSet () const
- virtual const char * GetEventName () const
- virtual ::gdcm::Event * MakeObject () const

27.74.1 Detailed Description

DataSetEvent Special type of event triggered during the DataSet store/move process.

See also

27.74.2 Member Typedef Documentation

27.74.2.1 typedef DataSetEvent gdcm::DataSetEvent::Self

27.74.2.2 typedef AnyEvent gdcm::DataSetEvent::Superclass

27.74.3 Constructor & Destructor Documentation

27.74.3.1 gdcm::DataSetEvent::DataSetEvent (DataSet const * *ds* = NULL) [inline]

27.74.3.2 virtual gdcm::DataSetEvent::~DataSetEvent ( ) [inline], [virtual]

27.74.3.3 gdcm::DataSetEvent::DataSetEvent (const Self & *s*) [inline]

27.74.4 Member Function Documentation

27.74.4.1 virtual bool gdcm::DataSetEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

27.74.4.2 DataSet const& gdcm::DataSetEvent::GetDataSet ( ) const [inline]

27.74.4.3 virtual const char* gdcm::DataSetEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

27.74.4.4 virtual ::gdcm::Event* gdcm::DataSetEvent::MakeObject ( ) const [inline], [virtual]

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

Implements gdcm::Event.

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

- gdcmDataSetEvent.h

27.75 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)

27.75.1 Detailed Description

DataSetHelper (internal class, not intended for user level)

27.75.2 Member Function Documentation

27.75.2.1 static 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

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

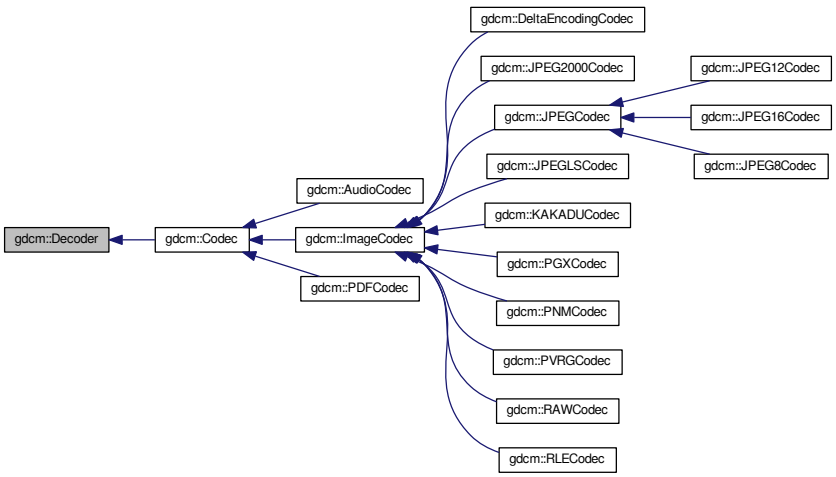
- gdcmDataSetHelper.h

27.76 gdcm::Decoder Class Reference

Decoder.

`#include <gdcmDecoder.h>`

Inheritance diagram for gdcm::Decoder:



Public Member Functions

- virtual ~Decoder ()
- 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 &)

27.76.1 Detailed Description

Decoder.

27.76.2 Constructor & Destructor Documentation

27.76.2.1 virtual gdcm::Decoder::~Decoder ( ) [inline], [virtual]

27.76.3 Member Function Documentation

27.76.3.1 virtual bool gdcm::Decoder::CanDecode (TransferSyntax const &) const [pure virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::ImageCodec, gdcm::JPEGLSCodec, gdcm::PNMCodec, gdcm::RAWCodec, gdcm::AudioCodec, gdcm::PDFCodec, gdcm::PGXCodec, and gdcm::KAKADUCodec.

27.76.3.2 virtual bool gdcm::Decoder::Decode (DataElement const & , DataElement &) [inline], [virtual]

Decode.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::ImageCodec, gdcm::DeltaEncodingCodec, gdcm::KAKADUCodec, gdcm::RAWCodec, gdcm::AudioCodec, and gdcm::PDFCodec.

27.76.3.3 virtual bool gdcm::Decoder::DecodeByStreams (std::istream & , std::ostream &) [inline], [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::ImageCodec, gdcm::RAWCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

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

- gdcmDecoder.h

27.77 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 ()

27.77.1 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.

27.77.2 Constructor & Destructor Documentation

27.77.2.1 gdcm::DefinedTerms::DefinedTerms ( ) [inline]

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

- gdcmDefinedTerms.h

27.78 gdcm::Defs Class Reference

FIXME I do not like the name 'Defs'.

`#include <gdcmDefs.h>`

Public Member Functions

- Defs ()
- ~Defs ()
- const IOD & GetIODFromFile (const File &file) const
- const IODs & GetIODs () const
- IODs & GetIODs ()
- const Macros & GetMacros () const
- Macros & GetMacros ()

- const Modules & GetModules () const
- Modules & GetModules ()
- Type GetTypeFromTag (const File &file, const Tag &tag) const
- bool IsEmpty () const
- bool Verify (const File &file) const
- bool Verify (const DataSet &ds) 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

27.78.1 Detailed Description

FIXME I do not like the name 'Defs'.

Note

bla

Examples:

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

27.78.2 Constructor & Destructor Documentation

27.78.2.1 gdcm::Defs::Defs ( )

27.78.2.2 gdcm::Defs::~Defs ( )

27.78.3 Member Function Documentation

27.78.3.1 const IOD& gdcm::Defs::GetIODFromFile (const File & *file*) const

27.78.3.2 static const char* gdcm::Defs::GetIODNameFromMediaStorage (MediaStorage const & *ms*) [static]

Examples:

GenerateStandardSOPClasses.cxx.

27.78.3.3 const IODs& gdcm::Defs::GetIODs ( ) const [inline]

Examples:

TraverseModules.cxx.

27.78.3.4 IODs& gdcm::Defs::GetIODs ( ) [inline]

27.78.3.5 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 conviently wraped within Modules. See gdcm::Module API directly

Examples:

TraverseModules.cxx.

27.78.3.6 Macros& gdcm::Defs::GetMacros ( ) [inline]

27.78.3.7 const Modules& gdcm::Defs::GetModules ( ) const [inline]

Examples:

TraverseModules.cxx.

27.78.3.8 Modules& gdcm::Defs::GetModules ( ) [inline]

27.78.3.9 Type gdcm::Defs::GetTypeFromTag (const File & *file*, const Tag & *tag*) const

27.78.3.10 bool gdcm::Defs::IsEmpty ( ) const [inline]

27.78.3.11 void gdcm::Defs::LoadDefaults ( ) [protected]

27.78.3.12 void gdcm::Defs::LoadFromFile (const char * *filename*) [protected]

27.78.3.13 bool gdcm::Defs::Verify (const File & *file*) const

27.78.3.14 bool gdcm::Defs::Verify (const DataSet & *ds*) const

27.78.4 Friends And Related Function Documentation

27.78.4.1 friend class Global [friend]

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

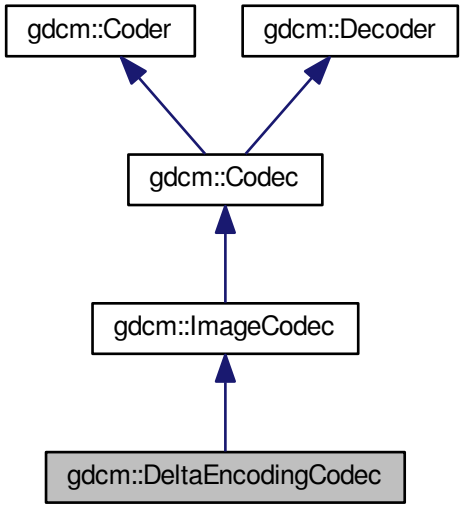
- gdcmDefs.h

27.79 gdcm::DeltaEncodingCodec Class Reference

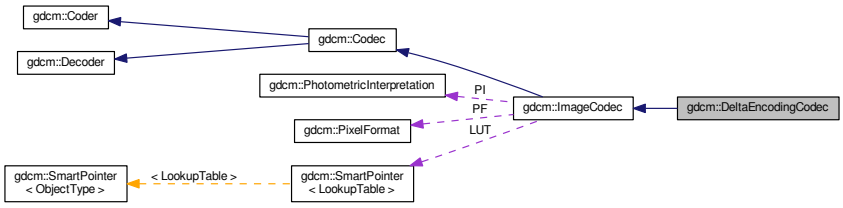
DeltaEncodingCodec compression used by some private vendor.

`#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.

Protected Member Functions

- bool Decode (std::istream &is, std::ostream &os)

Additional Inherited Members

27.79.1 Detailed Description

DeltaEncodingCodec compression used by some private vendor.

27.79.2 Constructor & Destructor Documentation

27.79.2.1 gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )

27.79.2.2 gdcm::DeltaEncodingCodec::~DeltaEncodingCodec ( )

27.79.3 Member Function Documentation

27.79.3.1 bool gdcm::DeltaEncodingCodec::CanDecode (TransferSyntax const & *ts*)

27.79.3.2 bool gdcm::DeltaEncodingCodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

27.79.3.3 bool gdcm::DeltaEncodingCodec::Decode (std::istream & *is*, std::ostream & *os*) [protected]

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

- gdcmDeltaEncodingCodec.h

27.80 gdcm::DICOMDIR Class Reference

DICOMDIR class.

`#include <gdcmDICOMDIR.h>`

Public Member Functions

- DICOMDIR ()
- DICOMDIR (const FileSet &fs)

27.80.1 Detailed Description

DICOMDIR class.

Structured for handling DICOMDIR

27.80.2 Constructor & Destructor Documentation

27.80.2.1 gdcm::DICOMDIR::DICOMDIR ( ) [inline]

27.80.2.2 gdcm::DICOMDIR::DICOMDIR (const FileSet & *fs*) [inline]

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

- gdcmDICOMDIR.h

27.81 gdcm::DICOMDIRGenerator Class Reference

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.

`#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 ()

27.81.1 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

27.81.2 Member Typedef Documentation

27.81.2.1 typedef Directory::FilenamesType gdcm::DICOMDIRGenerator::FilenamesType

27.81.2.2 typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType

27.81.3 Constructor & Destructor Documentation

27.81.3.1 gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )

27.81.3.2 gdcm::DICOMDIRGenerator::~DICOMDIRGenerator ( )

27.81.4 Member Function Documentation

27.81.4.1 bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( ) [protected]

27.81.4.2 bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( ) [protected]

27.81.4.3 bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( ) [protected]

27.81.4.4 bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( ) [protected]

27.81.4.5 bool gdcm::DICOMDIRGenerator::Generate ( )

Main function to generate the DICOMDIR.

27.81.4.6 File& gdcm::DICOMDIRGenerator::GetFile ( )

27.81.4.7 Scanner& gdcm::DICOMDIRGenerator::GetScanner ( ) [protected]

27.81.4.8 void gdcm::DICOMDIRGenerator::SetDescriptor (const char * *d*)

Set the File Set ID.

Warning

this need to be a valid VR::CS value

27.81.4.9 void gdcm::DICOMDIRGenerator::SetFile (const File & *f*)

Set/Get file. The DICOMDIR file will be valid once a call to Generate has been done.

27.81.4.10 void gdcm::DICOMDIRGenerator::SetFilenames (FilenamesType const & *fns*)

Set the list of filenames from which the DICOMDIR should be generated from.

27.81.4.11 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

27.82 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 ()
- 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

Protected Member Functions

- void LoadDefault ()

Friends

- class Dicts
- std::ostream & operator<< (std::ostream &_os, const Dict &_val)

27.82.1 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.

27.82.2 Member Typedef Documentation

27.82.2.1 typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator

27.82.2.2 typedef MapDictEntry::iterator gdcm::Dict::Iterator

27.82.2.3 typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry

27.82.3 Constructor & Destructor Documentation

27.82.3.1 gdcm::Dict::Dict ( ) [inline]

27.82.4 Member Function Documentation

27.82.4.1 void gdcm::Dict::AddDictEntry (const Tag & *tag,* const DictEntry & *de*) [inline]

27.82.4.2 ConstIterator gdcm::Dict::Begin ( ) const [inline]

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

27.82.4.3 ConstIterator gdcm::Dict::End ( ) const [inline]

Examples:

GenAllVR.cxx, and GenFakeIdentifyFile.cxx.

27.82.4.4 const DictEntry& gdcm::Dict::GetDictEntry (const Tag & *tag*) const [inline]

Examples:

GenFakeIdentifyFile.cxx, and PublicDict.cxx.

27.82.4.5 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.

27.82.4.6 const DictEntry& gdcm::Dict::GetDictEntryByName (const char * *name,* Tag & *tag*) const [inline]

Inefficient way of looking up tag by name. Technically DICOM does not garantee 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.

27.82.4.7 const char* gdcm::Dict::GetKeywordFromTag (Tag const & *tag*) const [inline]

Function to return the Keyword from a Tag.

27.82.4.8 bool gdcm::Dict::IsEmpty ( ) const [inline]

Referenced by gdcm::Dicts::IsEmpty().

27.82.4.9 void gdcm::Dict::LoadDefault ( ) [protected]

27.82.5 Friends And Related Function Documentation

27.82.5.1 friend class Dicts [friend]

27.82.5.2 std::ostream& operator<< (std::ostream & *_os,* const Dict & *_val*) [friend]

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

- gdcmDict.h

27.83 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 ()

27.83.1 Detailed Description

Class to convert a .dic file into something else:

- CXX code : embeded dict into shared lib (DICT_DEFAULT)
- Debug mode (DICT_DEBUG)
- XML dict (DICT_XML)

Note

27.83.2 Member Enumeration Documentation

27.83.2.1 enum gdcm::DictConverter::OutputTypes

Enumerator

DICT_DEFAULT

DICT_DEBUG

DICT_XML

27.83.3 Constructor & Destructor Documentation

27.83.3.1 gdcm::DictConverter::DictConverter ( )

27.83.3.2 gdcm::DictConverter::~DictConverter ( )

27.83.4 Member Function Documentation

27.83.4.1 void gdcm::DictConverter::AddGroupLength ( ) [protected]

27.83.4.2 void gdcm::DictConverter::Convert ( )

27.83.4.3 bool gdcm::DictConverter::ConvertToCXX (const char * *raw,* std::string & *cxx*) [protected]

27.83.4.4 bool gdcm::DictConverter::ConvertToXML (const char * *raw,* std::string & *cxx*) [protected]

27.83.4.5 const std::string& gdcm::DictConverter::GetDictName ( ) const

27.83.4.6 const std::string& gdcm::DictConverter::GetInputFilename ( ) const

27.83.4.7 const std::string& gdcm::DictConverter::GetOutputFilename ( ) const

27.83.4.8 int gdcm::DictConverter::GetOutputType ( ) const [inline]

27.83.4.9 static bool gdcm::DictConverter::Readuint16 (const char * *raw,* uint16_t & *ov*) [static]

27.83.4.10 static bool gdcm::DictConverter::ReadVM (const char * *raw,* VM::VMType & *type*) [static]

27.83.4.11 static bool gdcm::DictConverter::ReadVR (const char * *raw,* VR::VRType & *type*) [static]

27.83.4.12 void gdcm::DictConverter::SetDictName (const char * *name*)

27.83.4.13 void gdcm::DictConverter::SetInputFileName (const char * *filename*)

27.83.4.14 void gdcm::DictConverter::SetOutputFileName (const char * *filename*)

27.83.4.15 void gdcm::DictConverter::SetOutputType (int *type*) [inline]

27.83.4.16 void gdcm::DictConverter::WriteFooter ( ) [protected]

27.83.4.17 void gdcm::DictConverter::WriteHeader ( ) [protected]

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

- gdcmDictConverter.h

27.84 gdcm::DictEntry Class Reference

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.

`#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)

27.84.1 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.

27.84.2 Constructor & Destructor Documentation

27.84.2.1 gdcm::DictEntry::DictEntry (const char * *name* = " ", const char * *keyword* = " ", VR const & *vr* = VR::INVALID, VM const & *vm* = VM::VM0, bool *ret* = `false`) [inline]

27.84.3 Member Function Documentation

27.84.3.1 const char* gdcm::DictEntry::GetKeyword ( ) const [inline]

same as GetName but without spaces...

27.84.3.2 const char* gdcm::DictEntry::GetName ( ) const [inline]

Set/Get Name.

Referenced by gdcm::PrivateDict::PrintXML().

27.84.3.3 bool gdcm::DictEntry::GetRetired ( ) const [inline]

Set/Get Retired flag.

Examples:

GenAllVR.cxx.

27.84.3.4 const VM& gdcm::DictEntry::GetVM ( ) const [inline]

Set/Get VM.

Referenced by gdcm::PrivateDict::AddDictEntry(), and gdcm::PrivateDict::PrintXML().

27.84.3.5 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().

27.84.3.6 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)

27.84.3.7 void gdcm::DictEntry::SetElementXX (bool *v*) [inline]

Set whether element is shared in multiple elements (Source Image IDs typically)

27.84.3.8 void gdcm::DictEntry::SetGroupXX (bool *v*) [inline]

Set whether element is shared in multiple groups (Curve/Overlay typically)

27.84.3.9 void gdcm::DictEntry::SetKeyword (const char * *keyword*) [inline]

27.84.3.10 void gdcm::DictEntry::SetName (const char * *name*) [inline]

27.84.3.11 void gdcm::DictEntry::SetRetired (bool *retired*) [inline]

27.84.3.12 void gdcm::DictEntry::SetVM (VM const & *vm*) [inline]

27.84.3.13 void gdcm::DictEntry::SetVR (const VR & *vr*) [inline]

Referenced by gdcm::PrivateDict::AddDictEntry().

27.84.4 Friends And Related Function Documentation

27.84.4.1 friend class Dict [friend]

27.84.4.2 std::ostream& operator<< (std::ostream & *_os,* const DictEntry & *_val*) [friend]

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

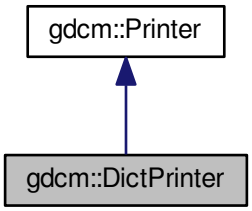
- gdcmDictEntry.h

27.85 gdcm::DictPrinter Class Reference

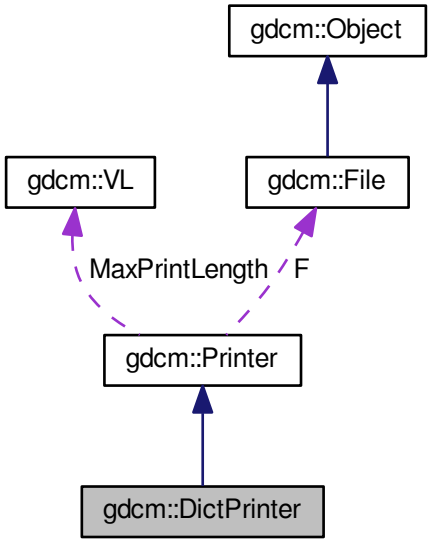
DictPrinter class.

`#include <gdcmDictPrinter.h>`

Inheritance diagram for gdcm::DictPrinter:



Collaboration diagram for gdcm::DictPrinter:



Public Member Functions

- DictPrinter ()
- ~DictPrinter ()
- void Print (std::ostream &os)

Protected Member Functions

- void PrintDataElement2 (std::ostream &os, const DataSet &ds, const DataElement &ide)
- void PrintDataSet2 (std::ostream &os, const DataSet &ds)

Additional Inherited Members

27.85.1 Detailed Description

DictPrinter class.

27.85.2 Constructor & Destructor Documentation

27.85.2.1 gdcm::DictPrinter::DictPrinter ( )

27.85.2.2 gdcm::DictPrinter::~DictPrinter ( )

27.85.3 Member Function Documentation

27.85.3.1 void gdcm::DictPrinter::Print (std::ostream & *os*)

27.85.3.2 void gdcm::DictPrinter::PrintDataElement2 (std::ostream & *os,* const DataSet & *ds,* const DataElement & *ide*) [protected]

27.85.3.3 void gdcm::DictPrinter::PrintDataSet2 (std::ostream & *os,* const DataSet & *ds*) [protected]

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

- gdcmDictPrinter.h

27.86 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 CSAHeaderDict & GetCSAHeaderDict () const
- const DictEntry & GetDictEntry (const Tag &tag, const char *owner=NULL) const

  NOT THREAD SAFE.
- const DictEntry & GetDictEntry (const PrivateTag &tag) const
- const PrivateDict & GetPrivateDict () const
- PrivateDict & GetPrivateDict ()
- const Dict & GetPublicDict () const
- bool IsEmpty () const

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)

27.86.1 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.

27.86.2 Member Enumeration Documentation

27.86.2.1 enum gdcm::Dicts::ConstructorType [protected]

Enumerator

PHILIPS

GEMS

SIEMENS

27.86.3 Constructor & Destructor Documentation

27.86.3.1 gdcm::Dicts::Dicts ( )

27.86.3.2 gdcm::Dicts::~Dicts ( )

27.86.4 Member Function Documentation

27.86.4.1 static const char* gdcm::Dicts::GetConstructorString (ConstructorType *type*) [static], [protected]

27.86.4.2 const CSAHeaderDict& gdcm::Dicts::GetCSAHeaderDict ( ) const

Examples:

MrProtocol.cxx.

27.86.4.3 const DictEntry& gdcm::Dicts::GetDictEntry (const Tag & *tag*, const char * *owner* = NULL) const

NOT 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.

27.86.4.4 const DictEntry& gdcm::Dicts::GetDictEntry (const PrivateTag & *tag*) const

27.86.4.5 const PrivateDict& gdcm::Dicts::GetPrivateDict ( ) const

27.86.4.6 PrivateDict& gdcm::Dicts::GetPrivateDict ( )

27.86.4.7 const Dict& gdcm::Dicts::GetPublicDict ( ) const

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, and ReadAndPrintAttributes.cxx.

27.86.4.8 bool gdcm::Dicts::IsEmpty ( ) const [inline]

References gdcm::Dict::IsEmpty().

27.86.4.9 void gdcm::Dicts::LoadDefaults ( ) [protected]

27.86.5 Friends And Related Function Documentation

27.86.5.1 friend class Global [friend]

27.86.5.2 std::ostream& operator<< (std::ostream & *_os*, const Dicts & *d*) [friend]

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

- gdcmDicts.h

27.87 gdcm::network::DIMSE Class Reference

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)

`#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 }

27.87.1 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)

27.87.2 Member Enumeration Documentation

27.87.2.1 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

27.88 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)

27.88.1 Detailed Description

class to handle DirectionCosines

Examples:

DiscriminateVolume.cxx.

27.88.2 Constructor & Destructor Documentation

27.88.2.1 gdcm::DirectionCosines::DirectionCosines ( )

27.88.2.2 gdcm::DirectionCosines::DirectionCosines (const double *dircos[6]*)

27.88.2.3 gdcm::DirectionCosines::~DirectionCosines ( )

27.88.3 Member Function Documentation

27.88.3.1 double gdcm::DirectionCosines::ComputeDistAlongNormal (const double *ipp[3]*) const

Compute the distance along the normal.

27.88.3.2 void gdcm::DirectionCosines::Cross (double *z[3]*) const

Compute Cross product.

27.88.3.3 double gdcm::DirectionCosines::CrossDot (DirectionCosines const & *dc*) const

Compute the Dot product of the two cross vector of both DirectionCosines object.

Examples:

DiscriminateVolume.cxx.

27.88.3.4 double gdcm::DirectionCosines::Dot ( ) const

Compute Dot.

27.88.3.5 bool gdcm::DirectionCosines::IsValid ( ) const

Return whether or not this is a valid direction cosines.

27.88.3.6 void gdcm::DirectionCosines::Normalize ( )

Normalize in-place.

27.88.3.7 gdcm::DirectionCosines::operator const double * ( ) const [inline]

Make the class behave like a const double *.

27.88.3.8 void gdcm::DirectionCosines::Print (std::ostream &) const

Print.

27.88.3.9 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

27.89 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 ()
- ~Directory ()
- 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)

27.89.1 Detailed Description

Class for manipulation directories.

Note

This implementation provide a cross platform implementation for manipulating directores: 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:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

27.89.2 Member Typedef Documentation

27.89.2.1 typedef std::vector<FilenameType> gdcm::Directory::FilenamesType

Examples:

DiscriminateVolume.cxx.

27.89.2.2 typedef std::string gdcm::Directory::FilenameType

27.89.3 Constructor & Destructor Documentation

27.89.3.1 gdcm::Directory::Directory ( ) [inline]

27.89.3.2 gdcm::Directory::~Directory ( ) [inline]

27.89.4 Member Function Documentation

27.89.4.1 unsigned int gdcm::Directory::Explore (FilenameType const & *name*, bool *recursive*) [protected]

Return number of file found when 'recursive'ly exploring directory `name`

27.89.4.2 FilenamesType const& gdcm::Directory::GetDirectories ( ) const [inline]

Return the Directories traversed.

27.89.4.3 FilenamesType const& gdcm::Directory::GetFilenames ( ) const [inline]

Set/Get the file names within the directory.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

27.89.4.4 FilenameType const& gdcm::Directory::GetToplevel ( ) const [inline]

Get the name of the toplevel directory.

27.89.4.5 unsigned int gdcm::Directory::Load (FilenameType const & *name*, bool *recursive* = false) [inline]

construct a list of filenames and subdirectory beneath directory: name

Warning

: hidden file and hidden directory are not loaded.

Examples:

DecompressImageMultiframe.cs, DiscriminateVolume.cxx, DumpToSQLITE3.cxx, gdcmorthoplanes.cxx, GenerateRTSTRUCT.cxx, ReadUTF8QtDir.cxx, reslicesphere.cxx, SortImage.cxx, threadgdcm.cxx, and VolumeSorter.cxx.

27.89.4.6 void gdcm::Directory::Print (std::ostream & *os* = std::cout) const

Print.

Examples:

SortImage.cxx.

Referenced by gdcm::operator<<().

27.89.5 Friends And Related Function Documentation

27.89.5.1 std::ostream& operator<< (std::ostream & *_os*, const Directory & *d*) [friend]

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

- gdcmDirectory.h

27.90 gdcm::DirectoryHelper Class Reference

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.

`#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)

27.90.1 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.

27.90.2 Member Function Documentation

27.90.2.1 static Directory::FilenamesType gdcm::DirectoryHelper::GetCTImageSeriesUIDs (const std::string & *inDirectory*) [static]

27.90.2.2 static Directory::FilenamesType gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs (const std::string & *inDirectory,* const std::string & *inSeriesUID*) [static]

Examples:

GenerateRTSTRUCT.cxx.

27.90.2.3 static std::string gdcm::DirectoryHelper::GetFrameOfReference (const std::vector< DataSet > & *inDS*) [static]

27.90.2.4 static Directory::FilenamesType gdcm::DirectoryHelper::GetMRImageSeriesUIDs (const std::string & *inDirectory*) [static]

27.90.2.5 static Directory::FilenamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs (const std::string & *inDirectory*) [static]

Examples:

GenerateRTSTRUCT.cxx.

27.90.2.6 static Directory::FilenamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID (const std::string & *inDirectory,* const std::string & *inSOPClassUID*) [static]

27.90.2.7 static std::string gdcm::DirectoryHelper::GetSOPClassUID (const std::vector< DataSet > & *inDS*) [static]

27.90.2.8 static std::string gdcm::DirectoryHelper::GetStringValueFromTag (const Tag & *t,* const DataSet & *ds*) [static]

27.90.2.9 static std::vector<DataSet> gdcm::DirectoryHelper::LoadImageFromFiles (const std::string & *inDirectory,* const std::string & *inSeriesUID*) [static]

27.90.2.10 static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex (int *inIndex,* const std::vector< DataSet > & *inDS*) [static]

27.90.2.11 static 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

27.91 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

`#include <gdcmDummyValueGenerator.h>`

Static Public Member Functions

- static const char * Generate (const char *input)

27.91.1 Detailed Description

Class for generating dummy value.

See also

Anonymizer

27.91.2 Member Function Documentation

27.91.2.1 static 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:

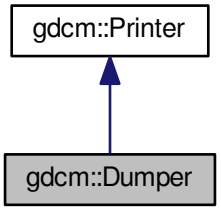
- gdcmDummyValueGenerator.h

27.92 gdcm::Dumper Class Reference

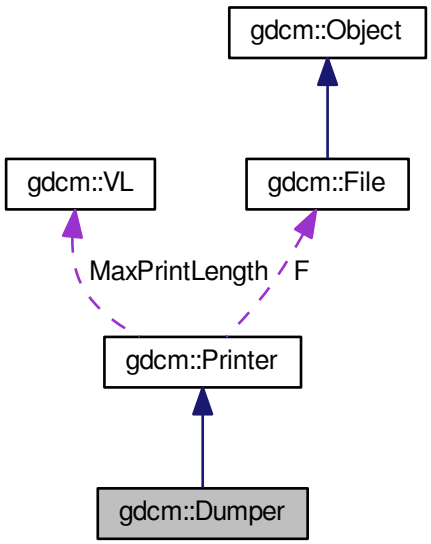
Codec class.

`#include <gdcmDumper.h>`

Inheritance diagram for gdcm::Dumper:



Collaboration diagram for gdcm::Dumper:



Public Member Functions

- Dumper ()
- ~Dumper ()

Additional Inherited Members

27.92.1 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

27.92.2 Constructor & Destructor Documentation

27.92.2.1 gdcm::Dumper::Dumper ( ) [inline]

27.92.2.2 gdcm::Dumper::~Dumper ( ) [inline]

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

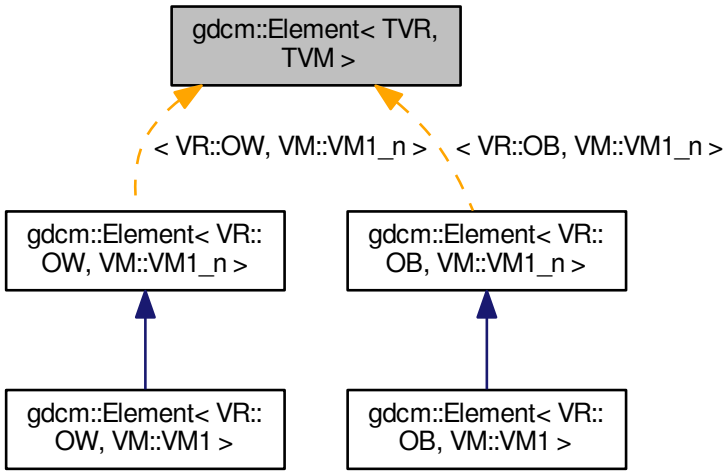
- gdcmDumper.h

27.93 gdcm::Element< TVR, TVM > Class Template Reference

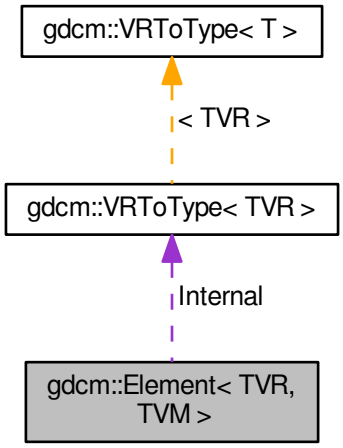
Element class.

`#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
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- 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)

27.93.1 Detailed Description

template<int TVR, int TVM>class gdcm::Element< TVR, TVM >

Element class.

Note

TODO

Examples:

csa2img.cxx, DumpADAC.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, Extracting_All_Resolution.↩cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GetSubSequenceData.cxx, and iU22tomultisc.cxx.

27.93.2 Member Typedef Documentation

27.93.2.1 template<int TVR, int TVM> typedef VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Type

27.93.3 Member Function Documentation

27.93.3.1 template<int TVR, int TVM> DataElement gdcm::Element< TVR, TVM >::GetAsDataElement ( ) const [inline]

27.93.3.2 template<int TVR, int TVM> unsigned long gdcm::Element< TVR, TVM >::GetLength ( ) const [inline]

27.93.3.3 template<int TVR, int TVM> const VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (unsigned int *idx* = 0) const [inline]

27.93.3.4 template<int TVR, int TVM> VRToType<TVR>::Type& gdcm::Element< TVR, TVM >::GetValue (unsigned int *idx* = 0) [inline]

27.93.3.5 template<int TVR, int TVM> const VRToType<TVR>::Type* gdcm::Element< TVR, TVM >::GetValues ( ) const [inline]

27.93.3.6 template<int TVR, int TVM> static VM gdcm::Element< TVR, TVM >::GetVM ( ) [inline], [static]

27.93.3.7 template<int TVR, int TVM> static VR gdcm::Element< TVR, TVM >::GetVR ( ) [inline], [static]

27.93.3.8 template<int TVR, int TVM> VRToType<TVR>::Type gdcm::Element< TVR, TVM >::operator[] (unsigned int *idx*) const [inline]

27.93.3.9 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Print (std::ostream & *_os*) const [inline]

27.93.3.10 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Read (std::istream & *_is*) [inline]

27.93.3.11 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Set (Value const & *v*) [inline]

27.93.3.12 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetFromDataElement (DataElement< TVR, TVM > const & *de*) [inline]

27.93.3.13 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetNoSwap (Value const & *v*) [inline], [protected]

27.93.3.14 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetValue (typename VRToType< TVR >::Type *v*, unsigned int *idx* = 0) [inline]

27.93.3.15 template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Write (std::ostream & *_os*) const [inline]

27.93.4 Member Data Documentation

27.93.4.1 template<int TVR, int TVM> VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]

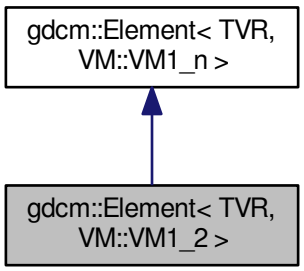
The documentation for this class was generated from the following file:

- gdcmElement.h

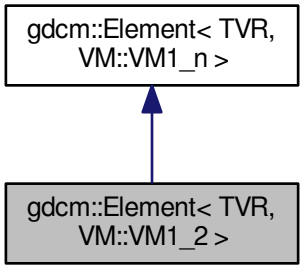
27.94 gdcm::Element< TVR, VM::VM1_2 > Class Template Reference

`#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

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

27.94.1 Member Typedef Documentation

27.94.1.1 template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM1_2 >::Parent

27.94.2 Member Function Documentation

27.94.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM1_2 >::SetLength (int *len*) [inline]

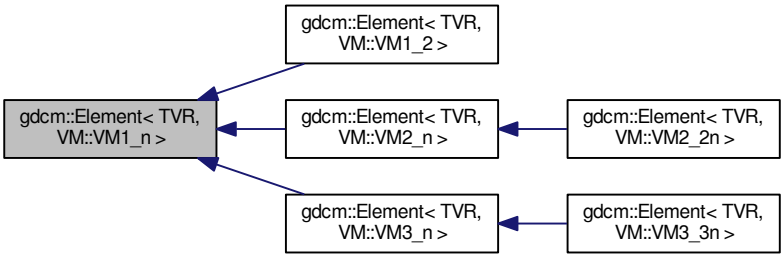
The documentation for this class was generated from the following file:

- gdcmElement.h

27.95 gdcm::Element< TVR, VM::VM1_n > Class Template Reference

`#include <gdcmElement.h>`

Inheritance 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
- const VRToType< TVR >::Type & GetValue (unsigned int idx=0) const
- VRToType< TVR >::Type & GetValue (unsigned int idx=0)
- 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)

27.95.1 Member Typedef Documentation

27.95.1.1 template<int TVR> typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type

27.95.2 Constructor & Destructor Documentation

27.95.2.1 template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element ( ) [inline], [explicit]

27.95.2.2 template<int TVR> gdcm::Element< TVR, VM::VM1_n >::~Element ( ) [inline]

27.95.2.3 template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element (const Element< TVR, VM::VM1_n > & *_val*) [inline]

27.95.3 Member Function Documentation

27.95.3.1 template<int TVR> DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]

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

27.95.3.2 template<int TVR> unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength ( ) const [inline]

27.95.3.3 template<int TVR> const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) const [inline]

27.95.3.4 template<int TVR> VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int *idx* = 0) [inline]

27.95.3.5 template<int TVR> static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( ) [inline], [static]

References gdcm::VM::VM1_n.

27.95.3.6 template<int TVR> static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( ) [inline], [static]

27.95.3.7 template<int TVR> Element& gdcm::Element< TVR, VM::VM1_n >::operator= (const Element< TVR, VM::VM1_n > & *_val*) [inline]

27.95.3.8 template<int TVR> VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (unsigned int *idx*) const [inline]

27.95.3.9 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Print (std::ostream & *_os*) const [inline]

27.95.3.10 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Read (std::istream & *_is*) [inline]

27.95.3.11 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Set (Value const & *v*) [inline]

References gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), and gdcm::VR::VRBINARY.

27.95.3.12 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetArray (const Type * *array,* unsigned long *len,* bool *save* = false) [inline]

27.95.3.13 template<int 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, and gdcm::VR::UN.

27.95.3.14 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetLength (unsigned long *len*) [inline]

27.95.3.15 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetNoSwap (Value const & *v*) [inline], [protected]

References gdcm::ByteValue::GetLength(), gdcm::ByteValue::GetPointer(), and gdcm::VR::VRBINARY.

27.95.3.16 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetValue (typename VRToType< TVR >::Type *v,* unsigned int *idx* = 0) [inline]

27.95.3.17 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Write (std::ostream & *_os*) const [inline]

27.95.3.18 template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::WriteASCII (std::ostream & *os*) const [inline]

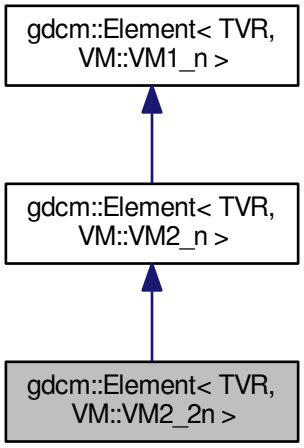
The documentation for this class was generated from the following file:

- gdcmElement.h

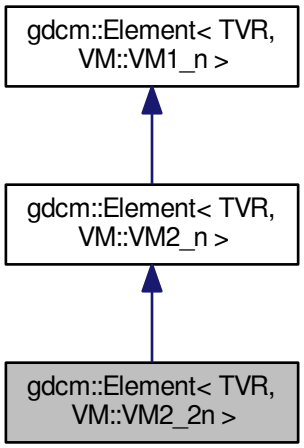
27.96 gdcm::Element< TVR, VM::VM2_2n > Class Template Reference

`#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

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

27.96.1 Member Typedef Documentation

27.96.1.1 template<int TVR> typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent

27.96.2 Member Function Documentation

27.96.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM2_2n >::SetLength (int *len*) [inline]

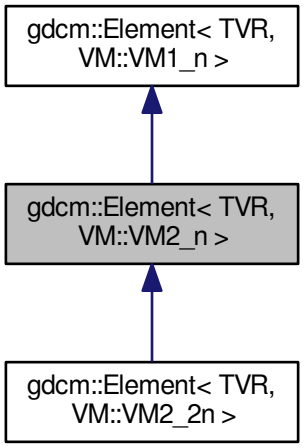
The documentation for this class was generated from the following file:

- gdcmElement.h

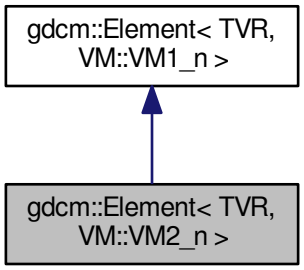
27.97 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

`#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

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

27.97.1 Member Typedef Documentation

27.97.1.1 template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM2_n >::Parent

27.97.2 Member Function Documentation

27.97.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM2_n >::SetLength (int *len*) [inline]

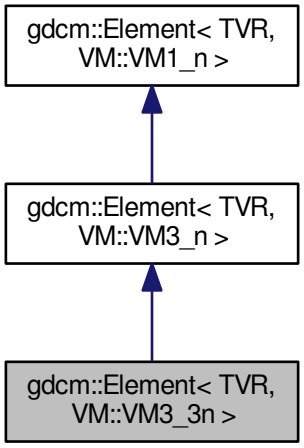
The documentation for this class was generated from the following file:

- gdcmElement.h

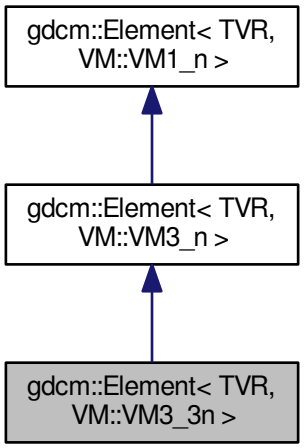
27.98 gdcm::Element< TVR, VM::VM3_3n > Class Template Reference

`#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

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

27.98.1 Member Typedef Documentation

27.98.1.1 template<int TVR> typedef Element<TVR, VM::VM3_n> gdcm::Element< TVR, VM::VM3_3n >::Parent

27.98.2 Member Function Documentation

27.98.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM3_3n >::SetLength (int *len*) [inline]

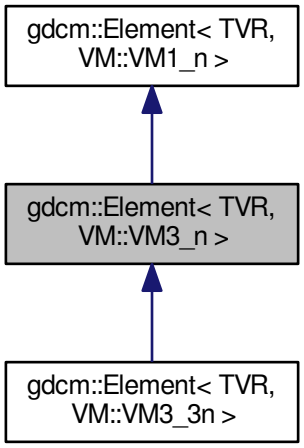
The documentation for this class was generated from the following file:

- gdcmElement.h

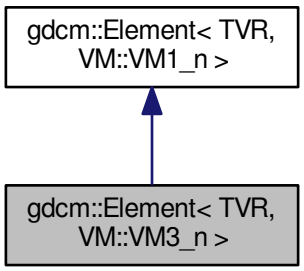
27.99 gdcm::Element< TVR, VM::VM3_n > Class Template Reference

`#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

Public Member Functions

- void SetLength (int len)

Additional Inherited Members

27.99.1 Member Typedef Documentation

27.99.1.1 template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_n >::Parent

27.99.2 Member Function Documentation

27.99.2.1 template<int TVR> void gdcm::Element< TVR, VM::VM3_n >::SetLength (int *len*) [inline]

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

- gdcmElement.h

27.100 gdcm::Element< VR::AS, VM::VM5 > Class Template Reference

`#include <gdcmElement.h>`

Public Member Functions

- unsigned long GetLength () const
- void Print (std::ostream &_os) const

Public Attributes

- char Internal [VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]

27.100.1 Member Function Documentation

27.100.1.1 unsigned long gdcm::Element< VR::AS, VM::VM5 >::GetLength ( ) const [inline]

27.100.1.2 void gdcm::Element< VR::AS, VM::VM5 >::Print (std::ostream & *_os*) const [inline]

27.100.2 Member Data Documentation

27.100.2.1 char gdcm::Element< VR::AS, VM::VM5 >::Internal[VMToLength< VM::VM5 >::Length *sizeof(VRToType< VR::AS >::Type)]

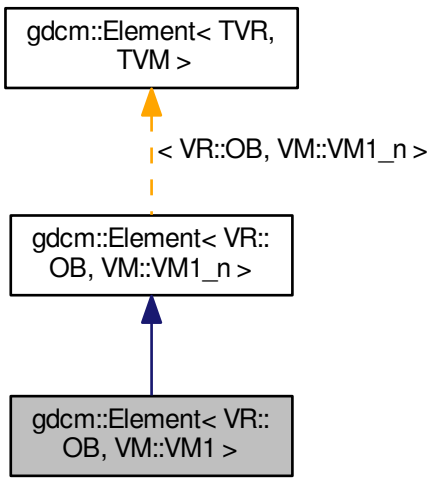
The documentation for this class was generated from the following file:

- gdcmElement.h

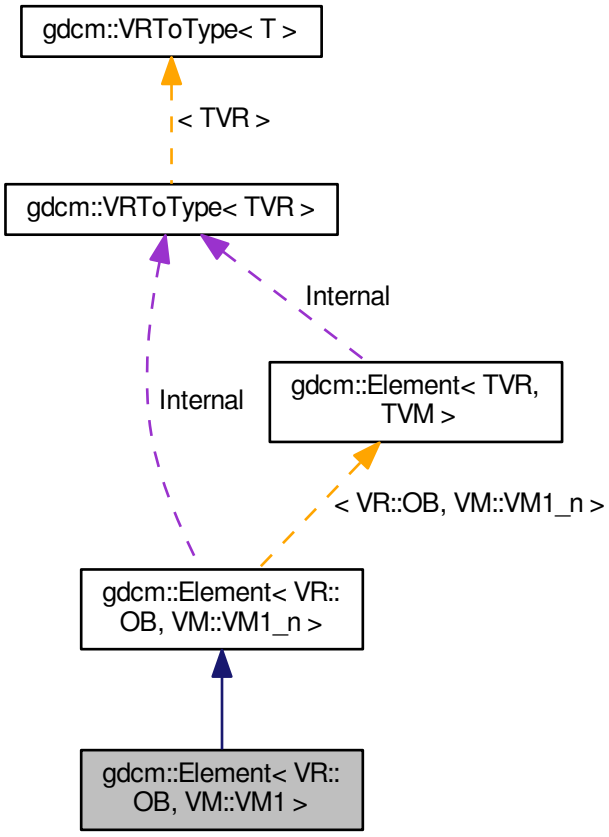
27.101 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference

`#include <gdcmElement.h>`

Inheritance diagram for gdcm::Element< VR::OB, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OB, VM::VM1 >:



Additional Inherited Members

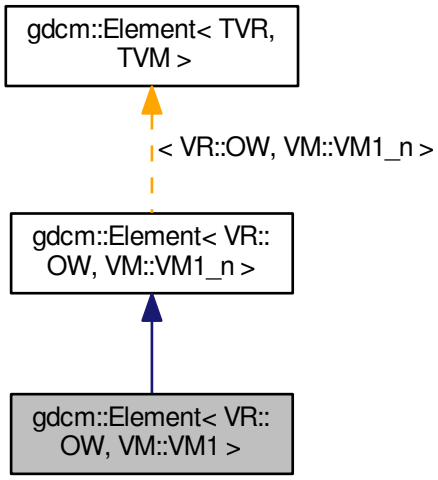
The documentation for this class was generated from the following file:

- gdcmElement.h

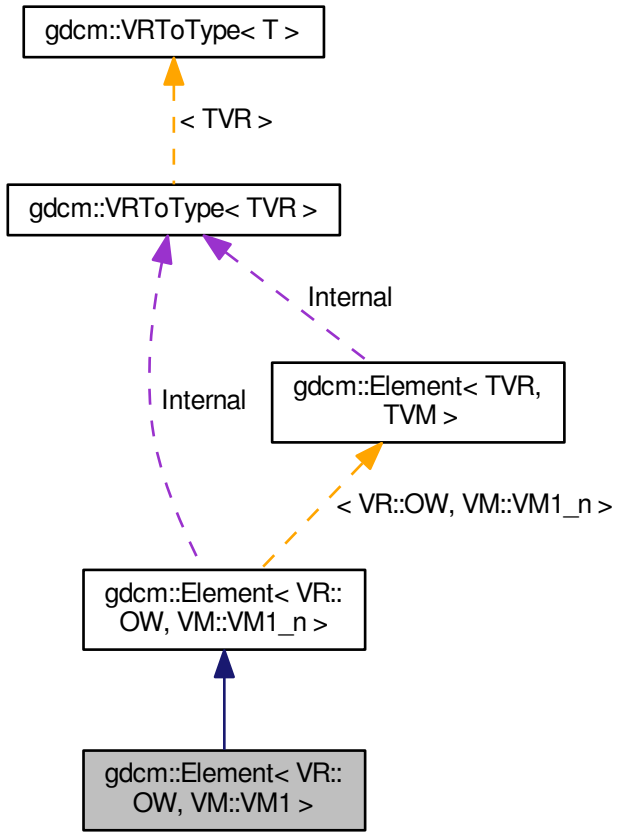
27.102 gdcm::Element< VR::OW, VM::VM1 > Class Template Reference

`#include <gdcmElement.h>`

Inheritance diagram for gdcm::Element< VR::OW, VM::VM1 >:



Collaboration diagram for gdcm::Element< VR::OW, VM::VM1 >:



Additional Inherited Members

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

- gdcmElement.h

27.103 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>`

27.103.1 Detailed Description

template<int 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

27.104 gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference

`#include <gdcmElement.h>`

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

- gdcmElement.h

27.105 gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference

`#include <gdcmElement.h>`

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

- gdcmElement.h

27.106 gdcm::EncapsulatedDocument Class Reference

EncapsulatedDocument.

`#include <gdcmEncapsulatedDocument.h>`

Public Member Functions

- EncapsulatedDocument ()

27.106.1 Detailed Description

EncapsulatedDocument.

27.106.2 Constructor & Destructor Documentation

27.106.2.1 gdcm::EncapsulatedDocument::EncapsulatedDocument ( ) [inline]

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

- gdcmEncapsulatedDocument.h

27.107 gdcm::EncodingImplementation< T > Singleton Reference

EncodingImplementation.

`#include <gdcmElement.h>`

27.107.1 Detailed Description

template<int T>singleton gdcm::EncodingImplementation< T >

EncodingImplementation.

Note

TODO

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

- gdcmElement.h

27.108 gdcm::EncodingImplementation< VR::VRASCII > Class Template Reference

`#include <gdcmElement.h>`

Public Member Functions

- template<>
  void Write (const float *data, unsigned long length, std::ostream &_os)
- 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)

27.108.1 Member Function Documentation

27.108.1.1 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Read (T * *data*, unsigned long *length*, std::istream & *_is*) [inline], [static]

27.108.1.2 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (T * *data,* unsigned int & *length,* std::istream & *_is*) [inline], [static]

References gdcm::backslash().

27.108.1.3 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (T * *data,* unsigned long *length,* std::istream & *_is*) [inline], [static]

27.108.1.4 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Write (const T * *data,* unsigned long *length,* std::ostream & *_os*) [inline], [static]

27.108.1.5 template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write (const float * *data,* unsigned long *length,* std::ostream & *_os*) [inline]

References gdcm::to_string().

27.108.1.6 template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write (const double * *data,* unsigned long *length,* std::ostream & *_os*) [inline]

References gdcm::to_string().

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

- gdcmElement.h

27.109 gdcm::EncodingImplementation< VR::VRBINARY > Class Template Reference

`#include <gdcmElement.h>`

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)

27.109.1 Member Function Documentation

27.109.1.1 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (T * *data,* unsigned long *length,* std::istream & *_is*) [inline], [static]

References gdcm::SwapperNoOp::SwapArray().

27.109.1.2 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (T * *data,* unsigned int & *length,* std::istream & *_is*) [inline], [static]

27.109.1.3 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (T * *data,* unsigned long *length,* std::istream & *_is*) [inline], [static]

27.109.1.4 template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::Write (const T * *data,* unsigned long *length,* std::ostream & *_os*) [inline], [static]

References gdcm::SwapperNoOp::Swap().

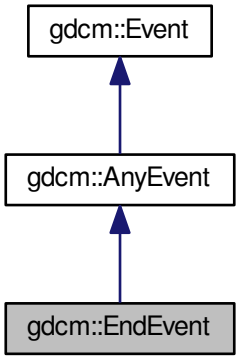
The documentation for this class was generated from the following file:

- gdcmElement.h

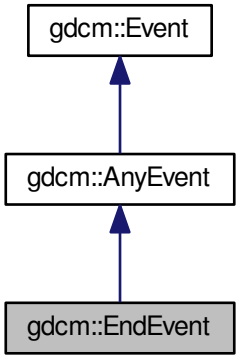
27.110 gdcm::EndEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::EndEvent:



Collaboration diagram for gdcm::EndEvent:



Additional Inherited Members

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

- gdcmEvent.h

27.111 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 ()

27.111.1 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.

27.111.2 Constructor & Destructor Documentation

27.111.2.1 gdcm::EnumeratedValues::EnumeratedValues ( ) [inline]

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

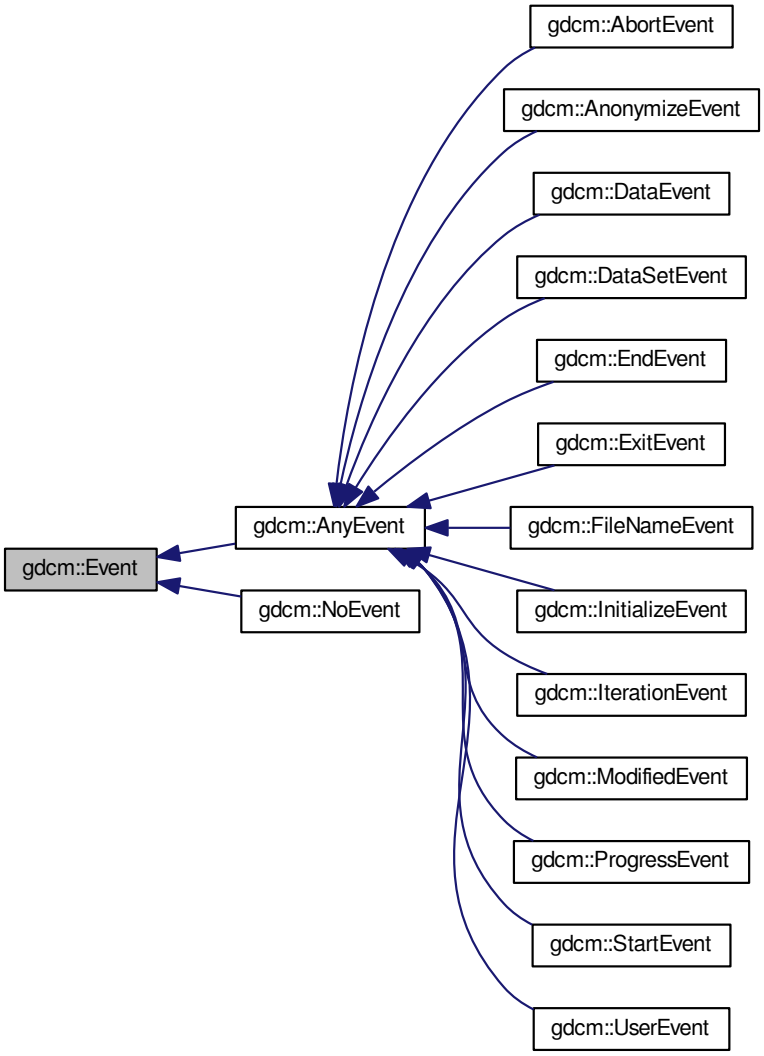
- gdcmEnumeratedValues.h

27.112 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 (void) const =0
- virtual Event * MakeObject () const =0
- virtual void Print (std::ostream &os) const

27.112.1 Detailed Description

superclass for callback/observer methods

See also

Command Subject

Examples:

SimpleScanner.cxx.

27.112.2 Constructor & Destructor Documentation

27.112.2.1 gdcm::Event::Event ( )

27.112.2.2 gdcm::Event::Event (const Event &)

27.112.2.3 virtual gdcm::Event::~Event ( ) [virtual]

27.112.3 Member Function Documentation

27.112.3.1 virtual bool gdcm::Event::CheckEvent (const Event *) const [pure virtual]

Check if given event matches or derives from this event.

27.112.3.2 virtual const char* gdcm::Event::GetEventName (void) const [pure virtual]

Return the StringName associated with the event.

Implemented in gdcm::FileNameEvent, gdcm::ProgressEvent, gdcm::DataSetEvent, gdcm::AnonymizeEvent, and gdcm::DataEvent.

27.112.3.3 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::FileNameEvent, gdcm::ProgressEvent, gdcm::DataSetEvent, gdcm::AnonymizeEvent, and gdcm::DataEvent.

27.112.3.4 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:

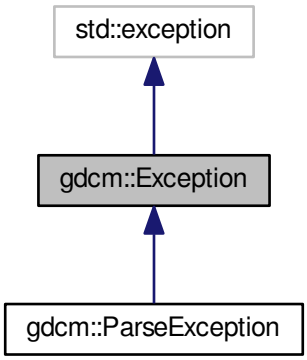
- gdcmEvent.h

27.113 gdcm::Exception Class Reference

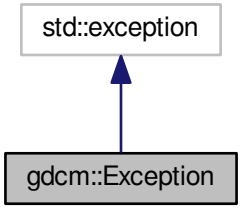
Exception.

`#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="")
- virtual ~Exception () throw ()
- const char * GetDescription () const

  Return the Description.
- const char * what () const throw ()

  what implementation

27.113.1 Detailed Description

Exception.

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

27.113.2 Constructor & Destructor Documentation

27.113.2.1 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.

27.113.2.2 virtual gdcm::Exception::~Exception ( ) throw) [inline], [virtual]

27.113.3 Member Function Documentation

27.113.3.1 const char* gdcm::Exception::GetDescription ( ) const [inline]

Return the Description.

Referenced by gdcm::SequenceOfItems::Read().

27.113.3.2 const char* gdcm::Exception::what ( ) const throw) [inline]

what implementation

Referenced by gdcm::SequenceOfFragments::ReadValue().

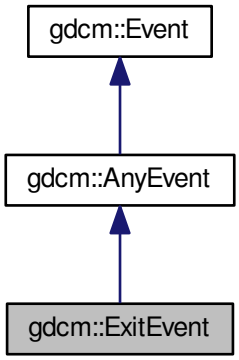
The documentation for this class was generated from the following file:

- gdcmException.h

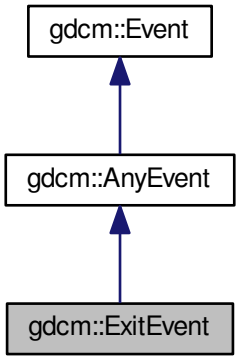
27.114 gdcm::ExitEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::ExitEvent:



Collaboration diagram for gdcm::ExitEvent:



Additional Inherited Members

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

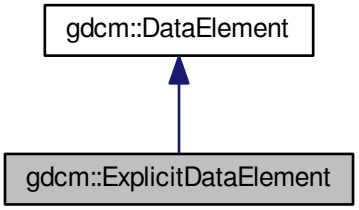
- gdcmEvent.h

27.115 gdcm::ExplicitDataElement Class Reference

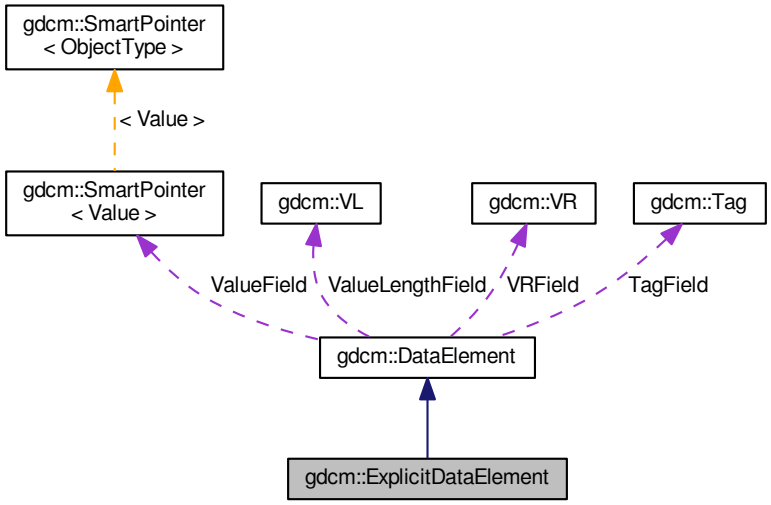
Class to read/write a DataElement as Explicit Data Element.

`#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

Additional Inherited Members

27.115.1 Detailed Description

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

Note

bla

27.115.2 Member Function Documentation

27.115.2.1 VL gdcm::ExplicitDataElement::GetLength ( ) const

27.115.2.2 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::Read (std::istream & *is*)

27.115.2.3 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadPreValue (std::istream & *is*)

27.115.2.4 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadValue (std::istream & *is*, bool *readvalues* = `true`)

27.115.2.5 template<typename TSwap > std::istream& gdcm::ExplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*)

27.115.2.6 template<typename TSwap > const std::ostream& gdcm::ExplicitDataElement::Write (std::ostream & *os*) const

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

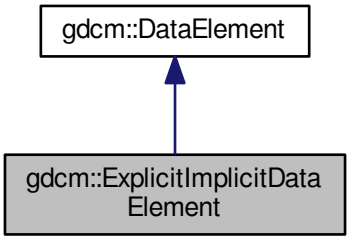
- gdcmExplicitDataElement.h

27.116 gdcm::ExplicitImplicitDataElement Class Reference

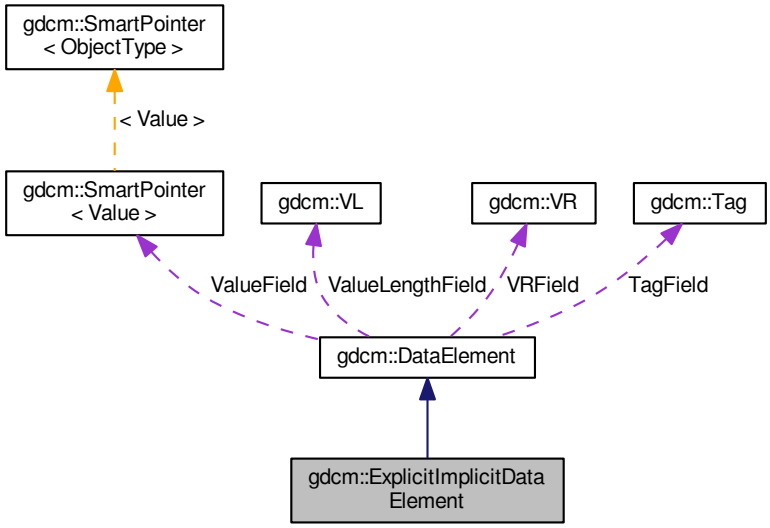
Class to read/write a DataElement as ExplicitImplicit Data Element.

`#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)

Additional Inherited Members

27.116.1 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.

27.116.2 Member Function Documentation

27.116.2.1 VL gdcm::ExplicitImplicitDataElement::GetLength ( ) const

27.116.2.2 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::Read (std::istream & *is*)

27.116.2.3 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadPreValue (std::istream & *is*)

27.116.2.4 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadValue (std::istream & *is,* bool *readvalues* = `true`)

27.116.2.5 template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadWithLength (std::istream & *is,* VL & *length*) `[inline]`

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

- gdcmExplicitImplicitDataElement.h

27.117 gdcm::Fiducials Class Reference

Fiducials.

`#include <gdcmFiducials.h>`

Public Member Functions

- Fiducials ()

27.117.1 Detailed Description

Fiducials.

27.117.2 Constructor & Destructor Documentation

27.117.2.1 gdcm::Fiducials::Fiducials ( ) [inline]

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

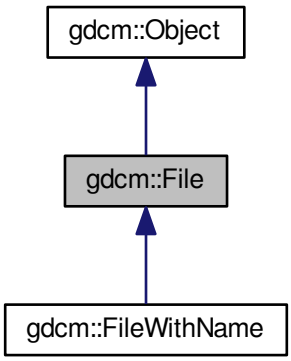
- gdcmFiducials.h

27.118 gdcm::File Class Reference

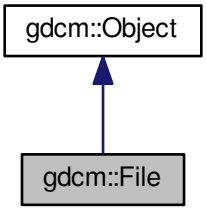
a DICOM File See PS 3.10 File: A File is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the File. Files are identified by a unique File ID and may by written, read and/or deleted.

`#include <gdcmFile.h>`

Inheritance diagram for gdcm::File:



Collaboration diagram for gdcm::File:



Public Member Functions

- File ()
- const DataSet & GetDataSet () const

  Get Data Set.
- DataSet & GetDataSet ()

  Get Data Set.
- const FileMetaInformation & GetHeader () const

  Get File Meta Information.
- FileMetaInformation & GetHeader ()

  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.

Friends

- std::ostream & operator<< (std::ostream &os, const File &val)

Additional Inherited Members

27.118.1 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, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DiffFile.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, EncapsulateFileInRaw↩Data.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_↩Writer.cxx, FixBrokenJ2K.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, Hello↩World.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.↩cxx, ReadAndPrintAttributes.cxx, ReadGEMSSDO.cxx, and StreamImageReaderTest.cxx.

27.118.2 Constructor & Destructor Documentation

27.118.2.1 gdcm::File::File ( ) [inline]

27.118.3 Member Function Documentation

27.118.3.1 const DataSet& gdcm::File::GetDataSet ( ) const [inline]

Get Data Set.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

27.118.3.2 DataSet& gdcm::File::GetDataSet ( ) [inline]

Get Data Set.

27.118.3.3 const FileMetaInformation& gdcm::File::GetHeader ( ) const [inline]

Get File Meta Information.

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetJPEGSamplePrecision.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MpegVideoInfo.cs, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

Referenced by gdcm::operator<<().

27.118.3.4 FileMetaInformation& gdcm::File::GetHeader ( ) [inline]

Get File Meta Information.

27.118.3.5 std::istream& gdcm::File::Read (std::istream & *is*)

Read.

27.118.3.6 void gdcm::File::SetDataSet (const DataSet & *ds*) [inline]

Set Data Set.

27.118.3.7 void gdcm::File::SetHeader (const FileMetaInformation & *fmi*) [inline]

Set File Meta Information.

27.118.3.8 std::ostream const& gdcm::File::Write (std::ostream & *os*) const

Write.

27.118.4 Friends And Related Function Documentation

27.118.4.1 std::ostream& operator<< (std::ostream & *os,* const File & *val*) [friend]

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

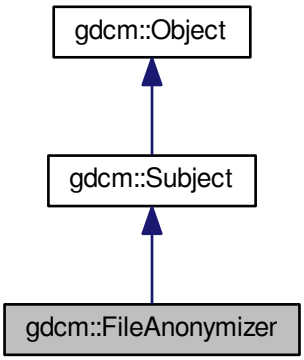
- gdcmFile.h

27.119 gdcm::FileAnonymizer Class Reference

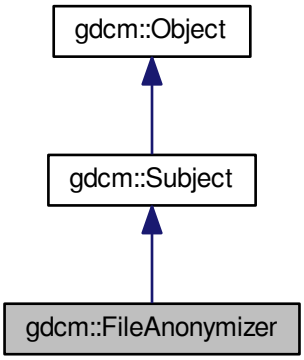
FileAnonymizer.

`#include <gdcmFileAnonymizer.h>`

Inheritance diagram for gdcm::FileAnonymizer:



Collaboration diagram for gdcm::FileAnonymizer:



Public Member Functions

- FileAnonymizer ()
- ~FileAnonymizer ()
- 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_str)
- void Replace (Tag const &t, const char *value_data, VL const &vl)
- void SetInputFileName (const char *filename_native)

  Set input filename.
- void SetOutputFileName (const char *filename_native)

  Set output filename.
- bool Write ()

  Write the output file.

Additional Inherited Members

27.119.1 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.

27.119.2 Constructor & Destructor Documentation

27.119.2.1 gdcm::FileAnonymizer::FileAnonymizer ( )

27.119.2.2 gdcm::FileAnonymizer::~FileAnonymizer ( )

27.119.3 Member Function Documentation

27.119.3.1 void gdcm::FileAnonymizer::Empty (Tag const & *t*)

Make Tag t empty Warning: does not handle SQ element

27.119.3.2 void gdcm::FileAnonymizer::Remove (Tag const & *t*)

remove a tag (even a SQ can be removed)

27.119.3.3 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 !

27.119.3.4 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

27.119.3.5 void gdcm::FileAnonymizer::SetInputFileName (const char * *filename_native*)

Set input filename.

Examples:

FileAnonymize.cs.

27.119.3.6 void gdcm::FileAnonymizer::SetOutputFileName (const char * *filename_native*)

Set output filename.

27.119.3.7 bool gdcm::FileAnonymizer::Write ( )

Write the output file.

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

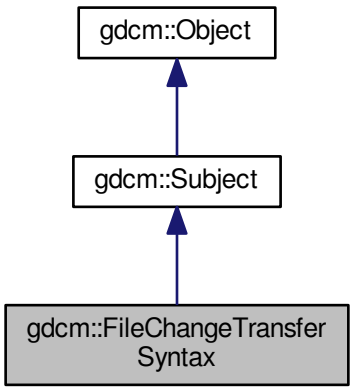
- gdcmFileAnonymizer.h

27.120 gdcm::FileChangeTransferSyntax Class Reference

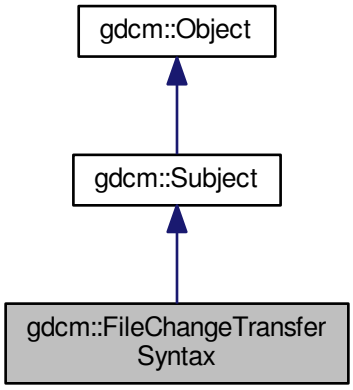
FileChangeTransferSyntax.

`#include <gdcmFileChangeTransferSyntax.h>`

Inheritance diagram for gdcm::FileChangeTransferSyntax:



Collaboration diagram for gdcm::FileChangeTransferSyntax:



Public Member Functions

- FileChangeTransferSyntax ()
- ~FileChangeTransferSyntax ()
- 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.

Static Public Member Functions

- static SmartPointer < FileChangeTransferSyntax > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

27.120.1 Detailed Description

FileChangeTransferSyntax.

This class is a file-based (limited) replacement of the in-memory ImageChangeTransferSyntax.

This class provide a file-based compression-only mecanism. 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

27.120.2 Constructor & Destructor Documentation

27.120.2.1 gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ( )

27.120.2.2 gdcm::FileChangeTransferSyntax::~FileChangeTransferSyntax ( )

27.120.3 Member Function Documentation

27.120.3.1 bool gdcm::FileChangeTransferSyntax::Change ( )

Change the transfer syntax.

27.120.3.2 ImageCodec* gdcm::FileChangeTransferSyntax::GetCodec ( )

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

27.120.3.3 static SmartPointer<FileChangeTransferSyntax> gdcm::FileChangeTransferSyntax::New ( ) [inline], [static]

for wrapped language: instantiate a reference counted object

27.120.3.4 void gdcm::FileChangeTransferSyntax::SetInputFileName (const char * *filename_native*)

Set input filename (raw DICOM)

27.120.3.5 void gdcm::FileChangeTransferSyntax::SetOutputFileName (const char * *filename_native*)

Set output filename (target compressed DICOM)

27.120.3.6 void gdcm::FileChangeTransferSyntax::SetTransferSyntax (TransferSyntax const & *ts*)

Specify the Target Transfer Syntax.

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

- gdcmFileChangeTransferSyntax.h

27.121 gdcm::FileDerivation Class Reference

FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.

`#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 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 ()

27.121.1 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 compulsary when creating a lossy derived image.

Examples:

GenFakeImage.cxx.

27.121.2 Constructor & Destructor Documentation

27.121.2.1 gdcm::FileDerivation::FileDerivation ( )

27.121.2.2 gdcm::FileDerivation::~FileDerivation ( )

27.121.3 Member Function Documentation

27.121.3.1 bool gdcm::FileDerivation::AddDerivationDescription ( ) [protected]

27.121.3.2 bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (DataSet & *ds*) [protected]

27.121.3.3 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:

GenFakeImage.cxx.

27.121.3.4 bool gdcm::FileDerivation::AddSourceImageSequence ( ) [protected]

27.121.3.5 bool gdcm::FileDerivation::Derive ( )

Change.

Examples:

GenFakeImage.cxx.

27.121.3.6 File& gdcm::FileDerivation::GetFile ( ) [inline]

Examples:

GenFakeImage.cxx.

27.121.3.7 const File& gdcm::FileDerivation::GetFile ( ) const [inline]

27.121.3.8 void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (unsigned int *codevalue*)

Specify the Derivation Code Sequence Code Value. Eg 113040.

Examples:

GenFakeImage.cxx.

27.121.3.9 void gdcm::FileDerivation::SetDerivationDescription (const char * *dd*)

Specify the Derivation Description. Eg "lossy conversion".

27.121.3.10 void gdcm::FileDerivation::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

GenFakeImage.cxx.

27.121.3.11 void gdcm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (unsigned int *codevalue*)

Specify the Purpose Of Reference Code Value. Eg. 121320.

Examples:

GenFakeImage.cxx.

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

- gdcmFileDerivation.h

27.122 gdcm::FileExplicitFilter Class Reference

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 reprensentation, the VR is not stored directly in the file.

`#include <gdcmFileExplicitFilter.h>`

Public Member Functions

- FileExplicitFilter ()
- ~FileExplicitFilter ()
- 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)

27.122.1 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 reprensentation, 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.

27.122.2 Constructor & Destructor Documentation

27.122.2.1 gdcm::FileExplicitFilter::FileExplicitFilter ( ) [inline]

27.122.2.2 gdcm::FileExplicitFilter::~FileExplicitFilter ( ) [inline]

27.122.3 Member Function Documentation

27.122.3.1 bool gdcm::FileExplicitFilter::Change ( )

Set FMI Transfer Syntax.

Change

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

27.122.3.2 bool gdcm::FileExplicitFilter::ChangeFMI ( ) [protected]

27.122.3.3 File& gdcm::FileExplicitFilter::GetFile ( ) [inline]

27.122.3.4 bool gdcm::FileExplicitFilter::ProcessDataSet (DataSet & *ds,* Dicts const & *dicts*) [protected]

27.122.3.5 void gdcm::FileExplicitFilter::SetChangePrivateTags (bool *b*) [inline]

Decide whether or not to VR'ify private tags.

27.122.3.6 void gdcm::FileExplicitFilter::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

GenAllVR.cxx, and LargeVRDSExplicit.cxx.

27.122.3.7 void gdcm::FileExplicitFilter::SetRecomputeItemLength (bool *b*)

By default set Sequence & Item length to Undefined to avoid recomputing length:

27.122.3.8 void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (bool *b*)

27.122.3.9 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:

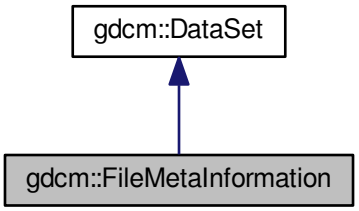
- gdcmFileExplicitFilter.h

27.123 gdcm::FileMetaInformation Class Reference

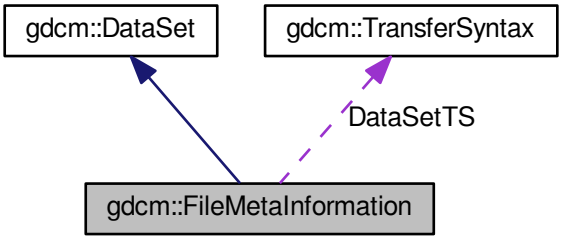
Class to represent a File Meta Information.

`#include <gdcmFileMetaInformation.h>`

Inheritance diagram for gdcm::FileMetaInformation:



Collaboration diagram for gdcm::FileMetaInformation:



Public Member Functions

- FileMetaInformation ()
- FileMetaInformation (FileMetaInformation const &fmi)
- 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
- const Preamble & GetPreamble () const

  Get Preamble.
- Preamble & GetPreamble ()
- void Insert (const DataElement &de)
- bool IsValid () const
- 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.

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)

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

27.123.1 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:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

27.123.2 Constructor & Destructor Documentation

27.123.2.1 gdcm::FileMetaInformation::FileMetaInformation ( ) [inline]

27.123.2.2 gdcm::FileMetaInformation::FileMetaInformation (FileMetaInformation const & *fmi*) [inline]

References DataSetMS, DataSetTS, and MetaInformationTS.

27.123.3 Member Function Documentation

27.123.3.1 static void gdcm::FileMetaInformation::AppendImplementationClassUID (const char * *imp*) [static]

27.123.3.2 void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( ) [protected]

27.123.3.3 void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( ) [protected]

27.123.3.4 void gdcm::FileMetaInformation::Default ( ) [protected]

27.123.3.5 void gdcm::FileMetaInformation::FillFromDataSet (DataSet const & *ds*)

Construct a FileMetaInformation from an already existing DataSet:

27.123.3.6 const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const [inline]

Examples:

GetJPEGSamplePrecision.cxx, and MergeTwoFiles.cxx.

27.123.3.7 static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion ( ) [static], [protected]

27.123.3.8 VL gdcm::FileMetaInformation::GetFullLength ( ) const [inline]

References gdcm::VL::GetLength().

27.123.3.9 static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( ) [static], [protected]

27.123.3.10 static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( ) [static], [protected]

27.123.3.11 static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( ) [static], [protected]

27.123.3.12 static const char* gdcm::FileMetaInformation::GetImplementationClassUID ( ) [static]

27.123.3.13 static const char* gdcm::FileMetaInformation::GetImplementationVersionName ( ) [static]

27.123.3.14 MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const

27.123.3.15 std::string gdcm::FileMetaInformation::GetMediaStorageAsString ( ) const

27.123.3.16 TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const [inline]

27.123.3.17 const Preamble& gdcm::FileMetaInformation::GetPreamble ( ) const [inline]

Get Preamble.

Referenced by gdcm::operator<<().

27.123.3.18 Preamble& gdcm::FileMetaInformation::GetPreamble ( ) [inline]

27.123.3.19 static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ( ) [static]

27.123.3.20 void gdcm::FileMetaInformation::Insert (const DataElement & *de*) [inline]

References gdcmErrorMacro, gdcm::Tag::GetGroup(), and gdcm::DataElement::GetTag().

27.123.3.21 bool gdcm::FileMetaInformation::IsValid ( ) const [inline]

27.123.3.22 std::istream& gdcm::FileMetaInformation::Read (std::istream & *is*)

Read.

27.123.3.23 std::istream& gdcm::FileMetaInformation::ReadCompat (std::istream & *is*)

27.123.3.24 template<typename TSwap > std::istream& gdcm::FileMetaInformation::ReadCompatInternal (std::istream & *is*) [protected]

27.123.3.25 void gdcm::FileMetaInformation::Replace (const **DataElement** & *de*) [inline]

Examples:

LargeVRDSExplicit.cxx.

References gdcm::DataElement::GetTag().

27.123.3.26 void gdcm::FileMetaInformation::SetDataSetTransferSyntax (const **TransferSyntax** & *ts*)

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_↩ Stream_Image_Writer.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, LargeVRDSExplicit.↩ cxx, MpegVideoInfo.cs, pmsct_rgb1.cxx, QIDO-RS.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

27.123.3.27 static void gdcm::FileMetaInformation::SetImplementationClassUID (const char * *imp*) [static]

Override the GDCM default values:

27.123.3.28 static void gdcm::FileMetaInformation::SetImplementationVersionName (const char * *version*) [static]

27.123.3.29 void gdcm::FileMetaInformation::SetPreamble (const **Preamble** & *p*) [inline]

27.123.3.30 static void gdcm::FileMetaInformation::SetSourceApplicationEntityTitle (const char * *title*) [static]

Examples:

ClinicalTrialIdentificationWorkflow.cs, FixJAIBugJPEGLS.cxx, GenerateDICOMDIR.cs, ReformatFile.cs, and StandardizeFiles.cs.

27.123.3.31 std::ostream& gdcm::FileMetaInformation::Write (std::ostream & *os*) const

Write.

27.123.4 Friends And Related Function Documentation

27.123.4.1 std::ostream& operator<< (std::ostream & *_os,* const **FileMetaInformation** & *_val*) [friend]

27.123.5 Member Data Documentation

27.123.5.1 **MediaStorage::MSType** gdcm::FileMetaInformation::DataSetMS [protected]

Referenced by FileMetaInformation().

27.123.5.2 TransferSyntax gdcm::FileMetaInformation::DataSetTS [protected]

Referenced by FileMetaInformation().

27.123.5.3 TransferSyntax::NegociatedType gdcm::FileMetaInformation::MetaInformationTS [protected]

Referenced by FileMetaInformation().

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

- gdcmFileMetaInformation.h

27.124 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 foward slash (UNIX style) to windows style slash.

Static Public Member Functions

- static const char * Join (const char *path, const char *filename)

27.124.1 Detailed Description

Class to manipulate file name's.

Note

OS independant representation of a filename (to query path, name and extension from a filename)

27.124.2 Constructor & Destructor Documentation

27.124.2.1 gdcm::Filename::Filename (const char * *filename* = " ") [inline]

27.124.3 Member Function Documentation

27.124.3.1 bool gdcm::Filename::EndWith (const char *ending[]*) const

Does the filename ends with a particular string ?

27.124.3.2 const char* gdcm::Filename::GetExtension ( )

return only the extension part of a filename

27.124.3.3 const char* gdcm::Filename::GetFileName ( ) const [inline]

Return the full filename.

27.124.3.4 const char* gdcm::Filename::GetName ( )

return only the name part of a filename

27.124.3.5 const char* gdcm::Filename::GetPath ( )

Return only the path component of a filename.

Examples:

ClinicalTrialIdentificationWorkflow.cs.

27.124.3.6 bool gdcm::Filename::IsEmpty ( ) const [inline]

return whether the filename is empty

27.124.3.7 bool gdcm::Filename::IsIdentical (Filename const & *fn*) const

27.124.3.8 static const char* gdcm::Filename::Join (const char * *path*, const char * *filename*) [static]

Join two paths NOT THREAD SAFE

Examples:

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

27.124.3.9 gdcm::Filename::operator const char * ( ) const [inline]

Simple operator to allow Filename myfilename("..."); const char * s = myfilename;

27.124.3.10 const char* gdcm::Filename::ToUnixSlashes ( )

Convert backslash (windows style) to UNIX style slash.

27.124.3.11 const char* gdcm::Filename::ToWindowsSlashes ( )

Convert foward slash (UNIX style) to windows style slash.

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

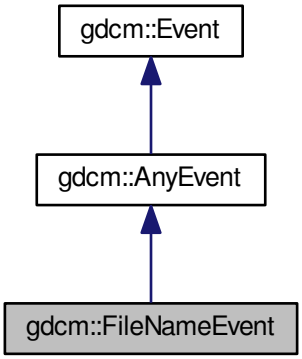
- gdcmFilename.h

27.125 gdcm::FileNameEvent Class Reference

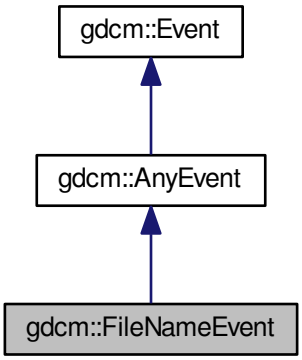
FileNameEvent Special type of event triggered during processing of FileSet.

`#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)
- virtual ~FileNameEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- virtual const char * GetEventName () const
- const char * GetFileName () const
- virtual ::gdcm::Event * MakeObject () const
- void SetFileName (const char *f)

27.125.1 Detailed Description

FileNameEvent Special type of event triggered during processing of FileSet.

See also

AnyEvent

Examples:

SimpleScanner.cxx.

27.125.2 Member Typedef Documentation

27.125.2.1 typedef FileNameEvent gdcm::FileNameEvent::Self

27.125.2.2 typedef AnyEvent gdcm::FileNameEvent::Superclass

27.125.3 Constructor & Destructor Documentation

27.125.3.1 gdcm::FileNameEvent::FileNameEvent (const char * *s* = " ") [inline]

27.125.3.2 virtual gdcm::FileNameEvent::~FileNameEvent ( ) [inline], [virtual]

27.125.3.3 gdcm::FileNameEvent::FileNameEvent (const Self & *s*) [inline]

27.125.4 Member Function Documentation

27.125.4.1 virtual bool gdcm::FileNameEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

27.125.4.2 virtual const char* gdcm::FileNameEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

27.125.4.3 const char* gdcm::FileNameEvent::GetFileName ( ) const [inline]

Examples:

SimpleScanner.cxx.

27.125.4.4 virtual ::gdcm::Event* gdcm::FileNameEvent::MakeObject ( ) const [inline], [virtual]

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

Implements gdcm::Event.

27.125.4.5 void gdcm::FileNameEvent::SetFileName (const char * *f*) [inline]

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

- gdcmFileNameEvent.h

27.126 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 ()
- 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.

27.126.1 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.

27.126.2 Member Typedef Documentation

27.126.2.1 typedef std::vector<FilenameType> gdcm::FilenameGenerator::FilenamesType

27.126.2.2 typedef std::string gdcm::FilenameGenerator::FilenameType

27.126.2.3 typedef FilenamesType::size_type gdcm::FilenameGenerator::SizeType

27.126.3 Constructor & Destructor Documentation

27.126.3.1 gdcm::FilenameGenerator::FilenameGenerator ( ) [inline]

27.126.3.2 gdcm::FilenameGenerator::~FilenameGenerator ( ) [inline]

27.126.4 Member Function Documentation

27.126.4.1 bool gdcm::FilenameGenerator::Generate ( )

Generate (return success)

Examples:

ConvertMultiFrameToSingleFrame.cxx.

27.126.4.2 const char* gdcm::FilenameGenerator::GetFilename (SizeType *n*) const

Get a particular filename (call after Generate)

Examples:

ConvertMultiFrameToSingleFrame.cxx.

27.126.4.3 FilenamesType const& gdcm::FilenameGenerator::GetFilenames ( ) const [inline]

Return all filenames.

27.126.4.4 SizeType gdcm::FilenameGenerator::GetNumberOfFilenames ( ) const

Examples:

ConvertMultiFrameToSingleFrame.cxx.

27.126.4.5 const char* gdcm::FilenameGenerator::GetPattern ( ) const [inline]

27.126.4.6 const char* gdcm::FilenameGenerator::GetPrefix ( ) const [inline]

27.126.4.7 void gdcm::FilenameGenerator::SetNumberOfFilenames (SizeType *nfiles*)

Set/Get the number of filenames to generate.

Examples:

ConvertMultiFrameToSingleFrame.cxx.

27.126.4.8 void gdcm::FilenameGenerator::SetPattern (const char * *pattern*) [inline]

Set/Get pattern.

Examples:

ConvertMultiFrameToSingleFrame.cxx.

27.126.4.9 void gdcm::FilenameGenerator::SetPrefix (const char * *prefix*) [inline]

Set/Get prefix.

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

- gdcmFilenameGenerator.h

27.127 gdcm::FileSet Class Reference

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.

`#include <gdcmFileSet.h>`

Public Types

- typedef std::vector< FileType > FilesType
- typedef std::string FileType

Public Member Functions

- FileSet ()
- void AddFile (File const &)
- bool AddFile (const char *filename)
- FilesType const & GetFiles () const
- void SetFiles (FilesType const &files)

Friends

- std::ostream & operator<< (std::ostream &_os, const FileSet &d)

27.127.1 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.

27.127.2 Member Typedef Documentation

27.127.2.1 typedef std::vector<FileType> gdcm::FileSet::FilesType

27.127.2.2 typedef std::string gdcm::FileSet::FileType

27.127.3 Constructor & Destructor Documentation

27.127.3.1 gdcm::FileSet::FileSet ( ) [inline]

27.127.4 Member Function Documentation

27.127.4.1 void gdcm::FileSet::AddFile (File const &) [inline]

Deprecated . Does nothing

27.127.4.2 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.

27.127.4.3 FilesType const& gdcm::FileSet::GetFiles ( ) const [inline]

27.127.4.4 void gdcm::FileSet::SetFiles (FilesType const & *files*)

27.127.5 Friends And Related Function Documentation

27.127.5.1 std::ostream& operator<< (std::ostream & *_os,* const FileSet & *d*) [friend]

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

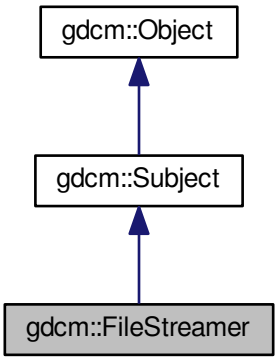
- gdcmFileSet.h

27.128 gdcm::FileStreamer Class Reference

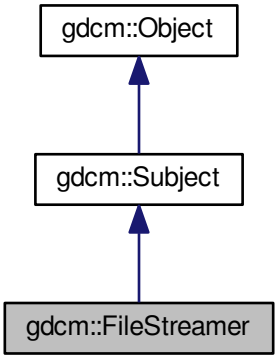
FileStreamer This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data.

`#include <gdcmFileStreamer.h>`

Inheritance diagram for gdcm::FileStreamer:



Collaboration diagram for gdcm::FileStreamer:



Public Member Functions

- FileStreamer ()
- ~FileStreamer ()
- 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.

Static Public Member Functions

- static SmartPointer< FileStreamer > New ()

  for wrapped language: instantiate a reference counted object

Additional Inherited Members

27.128.1 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.

27.128.2 Constructor & Destructor Documentation

27.128.2.1 gdcm::FileStreamer::FileStreamer ( )

27.128.2.2 gdcm::FileStreamer::~FileStreamer ( )

27.128.3 Member Function Documentation

27.128.3.1 bool gdcm::FileStreamer::AppendToDataElement (const Tag & *t*, const char * *array*, size_t *len*)

Append to previously started Tag t.

27.128.3.2 bool gdcm::FileStreamer::AppendToGroupDataElement (const PrivateTag & *pt*, const char * *array*, size_t *len*)

Append to previously started private creator.

27.128.3.3 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

27.128.3.4 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.

27.128.3.5 static SmartPointer<FileStreamer> gdcm::FileStreamer::New ( ) [inline], [static]

for wrapped language: instantiate a reference counted object

27.128.3.6 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

27.128.3.7 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

27.128.3.8 void gdcm::FileStreamer::SetOutputFileName (const char * *filename_native*)

Set output filename (target file)

27.128.3.9 void gdcm::FileStreamer::SetTemplateFileName (const char * *filename_native*)

Set input DICOM template filename.

Examples:

FileStreaming.cs.

27.128.3.10 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.

27.128.3.11 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.

27.128.3.12 bool gdcm::FileStreamer::StopDataElement (const Tag & *t*)

Stop appending to tag t. This will compute the proper attribute length.

27.128.3.13 bool gdcm::FileStreamer::StopGroupDataElement (const PrivateTag & *pt*)

Stop appending to private creator.

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

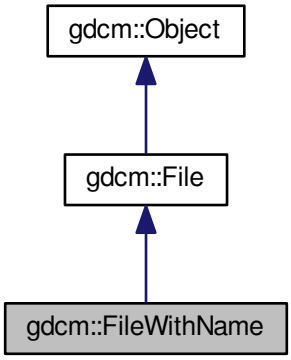
- gdcmFileStreamer.h

27.129 gdcm::FileWithName Class Reference

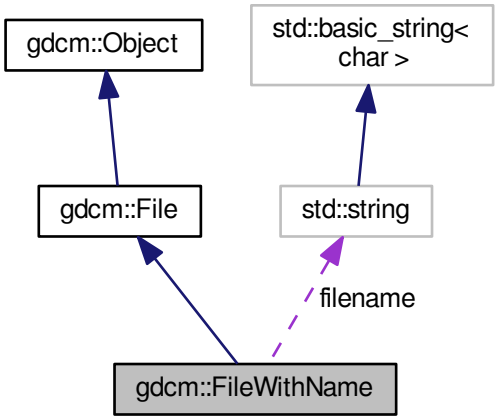
FileWithName.

`#include <gdcmSerieHelper.h>`

Inheritance diagram for gdcm::FileWithName:



Collaboration diagram for gdcm::FileWithName:



Public Member Functions

- FileWithName (File &f)

Public Attributes

- std::string filename

Additional Inherited Members

27.129.1 Detailed Description

FileWithName.

Backward only class do not use in newer code

27.129.2 Constructor & Destructor Documentation

27.129.2.1 gdcm::FileWithName::FileWithName (File & *f*) [inline]

27.129.3 Member Data Documentation

27.129.3.1 std::string gdcm::FileWithName::filename

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

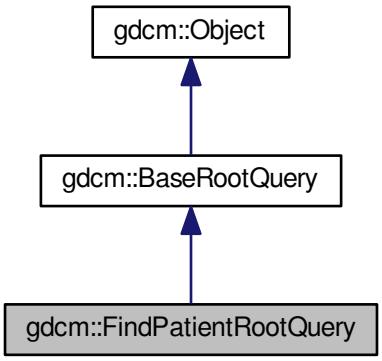
- gdcmSerieHelper.h

27.130 gdcm::FindPatientRootQuery Class Reference

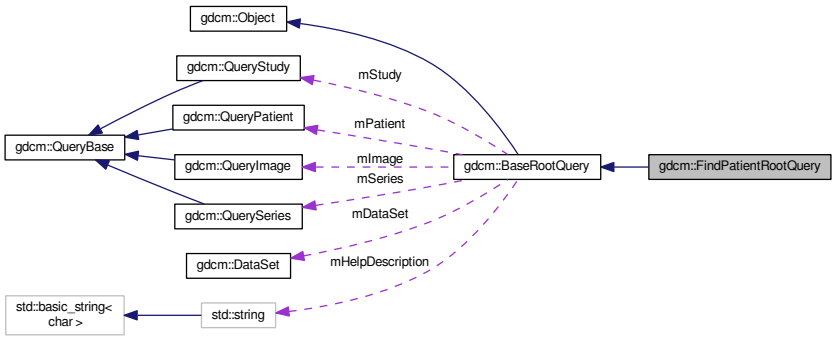
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

`#include <gdcmFindPatientRootQuery.h>`

Inheritance diagram for gdcm::FindPatientRootQuery:



Collaboration diagram for gdcm::FindPatientRootQuery:



Public Member Functions

- FindPatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

27.130.1 Detailed Description

PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

27.130.2 Constructor & Destructor Documentation

27.130.2.1 gdcm::FindPatientRootQuery::FindPatientRootQuery ( )

27.130.3 Member Function Documentation

27.130.3.1 UIDs::TSName gdcm::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseRootQuery.

27.130.3.2 std::vector<Tag> gdcm::FindPatientRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [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.

27.130.3.3 void gdcm::FindPatientRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [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.

27.130.3.4 bool gdcm::FindPatientRootQuery::ValidateQuery (bool *inStrict* = true) const [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.

27.130.4 Friends And Related Function Documentation

27.130.4.1 friend class QueryFactory [friend]

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

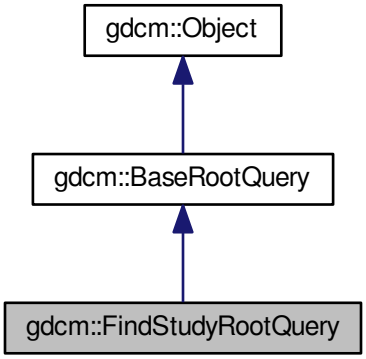
- gdcmFindPatientRootQuery.h

27.131 gdcm::FindStudyRootQuery Class Reference

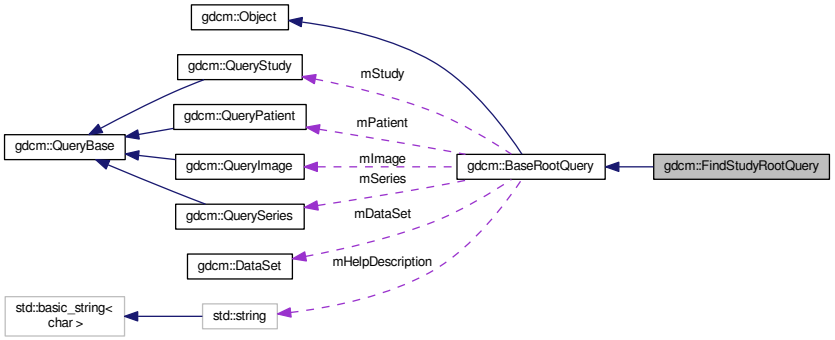
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

```
#include <gdcmFindStudyRootQuery.h>
```

Inheritance diagram for gdcm::FindStudyRootQuery:



Collaboration diagram for gdcm::FindStudyRootQuery:



Public Member Functions

- FindStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

27.131.1 Detailed Description

FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

27.131.2 Constructor & Destructor Documentation

27.131.2.1 gdcm::FindStudyRootQuery::FindStudyRootQuery ( )

27.131.3 Member Function Documentation

27.131.3.1 UIDs::TSName gdcm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseRootQuery.

27.131.3.2 std::vector<Tag> gdcm::FindStudyRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [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.

27.131.3.3 void gdcm::FindStudyRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [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.

27.131.3.4 bool gdcm::FindStudyRootQuery::ValidateQuery (bool *inStrict* = true) const [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.

27.131.4 Friends And Related Function Documentation

27.131.4.1 friend class QueryFactory [friend]

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

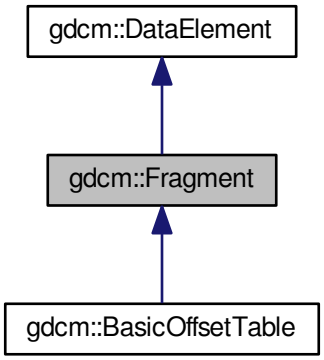
- gdcmFindStudyRootQuery.h

27.132 gdcm::Fragment Class Reference

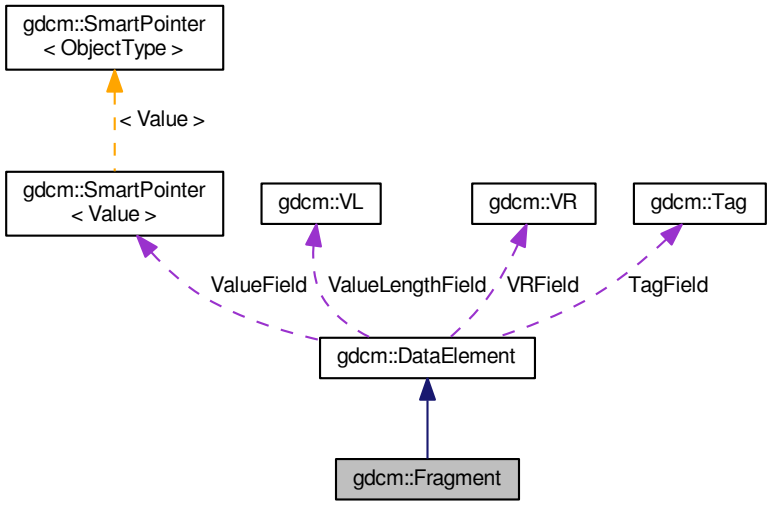
Class to represent a Fragment.

`#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

Friends

- std::ostream & operator<< (std::ostream &os, const Fragment &val)

Additional Inherited Members

27.132.1 Detailed Description

Class to represent a Fragment.

Examples:

FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

27.132.2 Constructor & Destructor Documentation

27.132.2.1 gdcm::Fragment::Fragment ( ) [inline]

27.132.3 Member Function Documentation

27.132.3.1 VL gdcm::Fragment::ComputeLength ( ) const

27.132.3.2 VL gdcm::Fragment::GetLength ( ) const

27.132.3.3 template<typename TSwap > std::istream& gdcm::Fragment::Read (std::istream & *is*) [inline]

Referenced by gdcm::SequenceOfFragments::ReadValue().

27.132.3.4 template<typename TSwap > std::istream& gdcm::Fragment::ReadBacktrack (std::istream & *is*) [inline]

References gdcmErrorMacro, gdcmWarningMacro, and gdcm::ParseException::SetLastElement().

Referenced by gdcm::SequenceOfFragments::ReadValue().

27.132.3.5 template<typename TSwap > std::istream& gdcm::Fragment::ReadPreValue (std::istream & *is*) [inline]

27.132.3.6 template<typename TSwap > std::istream& gdcm::Fragment::ReadValue (std::istream & *is*) [inline]

References gdcmWarningMacro, and gdcm::ParseException::SetLastElement().

27.132.3.7 template<typename TSwap > std::ostream& gdcm::Fragment::Write (std::ostream & *os*) const [inline]

References gdcm::ByteValue::ComputeLength(), gdcm::ByteValue::GetLength(), gdcm::VL::Write(), and gdcm::ByteValue::Write().

27.132.4 Friends And Related Function Documentation

27.132.4.1 std::ostream& operator<< (std::ostream & *os,* const Fragment & *val*) [friend]

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

- gdcmFragment.h

27.133 gdcm::Global Class Reference

Global.

`#include <gdcmGlobal.h>`

Public Member Functions

- Global ()
- ~Global ()
- bool Append (const char *path)
- Defs const & GetDefs () const
- Dicts const & GetDicts () const
- Dicts & GetDicts ()
- bool LoadResourcesFiles ()
- 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)

27.133.1 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:

GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, PublicDict.cxx, ReadAndPrint↩ Attributes.cxx, and TraverseModules.cxx.

27.133.2 Constructor & Destructor Documentation

27.133.2.1 gdcm::Global::Global ( )

27.133.2.2 gdcm::Global::∼Global ( )

27.133.3 Member Function Documentation

27.133.3.1 bool gdcm::Global::Append (const char * *path*)

Append path at the end of the path list

Warning

not thread safe !

27.133.3.2 Defs const& gdcm::Global::GetDefs ( ) const

retrieve the default/internal (Part 3) You need to explicitely call LoadResourcesFiles before

Examples:

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

27.133.3.3 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.

27.133.3.4 Dicts& gdcm::Global::GetDicts ( )

27.133.3.5 static Global& gdcm::Global::GetInstance ( ) [static]

return the singleton instance

Examples:

BasicAnonymizer.cs, ClinicalTrialIdentificationWorkflow.cs, GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, MrProtocol.cxx, PublicDict.cxx, ReadAndPrintAttributes.cxx, and TraverseModules.cxx.

27.133.3.6 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:

GenerateStandardSOPClasses.cxx, and TraverseModules.cxx.

27.133.3.7 const char* gdcm::Global::Locate (const char * *resfile*) const [protected]

Locate a resource file.

27.133.3.8 bool gdcm::Global::Prepend (const char * *path*)

Prepend path at the beginning of the path list

Warning

not thread safe !

27.133.4 Friends And Related Function Documentation

27.133.4.1 std::ostream& operator<< (std::ostream & *_os,* const Global & *g*) [friend]

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

- gdcmGlobal.h

27.134 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 ()
- 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)

27.134.1 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 ?

27.134.2 Member Typedef Documentation

27.134.2.1 typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector

27.134.3 Constructor & Destructor Documentation

27.134.3.1 gdcm::GroupDict::GroupDict ( ) [inline]

27.134.3.2 gdcm::GroupDict::~GroupDict ( ) [inline]

27.134.4 Member Function Documentation

27.134.4.1 void gdcm::GroupDict::Add (std::string const & *abbreviation*, std::string const & *name*) [protected]

27.134.4.2 std::string const& gdcm::GroupDict::GetAbbreviation (uint16_t *num*) const

Referenced by gdcm::operator<<().

27.134.4.3 std::string const& gdcm::GroupDict::GetName (uint16_t *num*) const

Referenced by gdcm::operator<<().

27.134.4.4 void gdcm::GroupDict::Insert (uint16_t *num*, std::string const & *abbreviation*, std::string const & *name*) [protected]

27.134.4.5 size_t gdcm::GroupDict::Size ( ) const [inline]

Referenced by gdcm::operator<<().

27.134.5 Friends And Related Function Documentation

27.134.5.1 std::ostream& operator<< (std::ostream & *_os*, const GroupDict & *_val*) [friend]

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

- gdcmGroupDict.h

27.135 gdcm::IconImageFilter Class Reference

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.

`#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 ()

27.135.1 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.

27.135.2 Constructor & Destructor Documentation

27.135.2.1 gdcm::IconImageFilter::IconImageFilter ( )

27.135.2.2 gdcm::IconImageFilter::~IconImageFilter ( )

27.135.3 Member Function Documentation

27.135.3.1 bool gdcm::IconImageFilter::Extract ( )

Extract all Icon found in File.

Examples:

ExtractIconFromFile.cxx.

27.135.3.2 void gdcm::IconImageFilter::ExtractIconImages ( ) [protected]

27.135.3.3 void gdcm::IconImageFilter::ExtractVeproIconImages ( ) [protected]

27.135.3.4 File& gdcm::IconImageFilter::GetFile ( ) [inline]

27.135.3.5 const File& gdcm::IconImageFilter::GetFile ( ) const [inline]

27.135.3.6 IconImage& gdcm::IconImageFilter::GetIconImage (unsigned int *i*) const

Examples:

ExtractIconFromFile.cxx.

27.135.3.7 unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const

Retrieve extract IconImage (need to call Extract first)

Examples:

ExtractIconFromFile.cxx.

27.135.3.8 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

27.136 gdcm::IconImageGenerator Class Reference

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:

```
#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.

27.136.1 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.

27.136.2 Constructor & Destructor Documentation

27.136.2.1 gdcm::IconImageGenerator::IconImageGenerator ( )

27.136.2.2 gdcm::IconImageGenerator::~IconImageGenerator ( )

27.136.3 Member Function Documentation

27.136.3.1 void gdcm::IconImageGenerator::AutoPixelMinMax (bool *b*)

Instead of explicitely 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.

27.136.3.2 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

27.136.3.3 bool gdcm::IconImageGenerator::Generate ( )

Generate Icon.

Examples:

ExtractIconFromFile.cxx.

27.136.3.4 const IconImage& gdcm::IconImageGenerator::GetIconImage ( ) const [inline]

Retrieve generated Icon.

Examples:

ExtractIconFromFile.cxx.

27.136.3.5 Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) [inline]

27.136.3.6 const Pixmap& gdcm::IconImageGenerator::GetPixmap ( ) const [inline]

27.136.3.7 void gdcm::IconImageGenerator::SetOutputDimensions (const unsigned int *dims[2]*)

Set Target dimension of output Icon.

Examples:

ExtractIconFromFile.cxx.

27.136.3.8 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)

27.136.3.9 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.

27.136.3.10 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

27.137 gdcm::ignore_char Struct Reference

`#include <gdcmElement.h>`

Public Member Functions

- ignore_char (char c)

Public Attributes

- char m_char

27.137.1 Constructor & Destructor Documentation

27.137.1.1 gdcm::ignore_char::ignore_char (char *c*) [inline]

27.137.2 Member Data Documentation

27.137.2.1 char gdcm::ignore_char::m_char

Referenced by gdcm::operator>>().

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

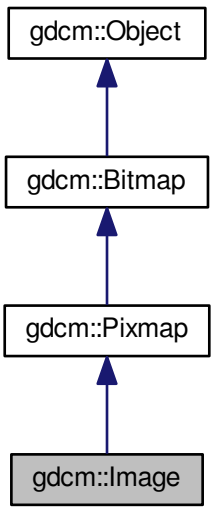
- gdcmElement.h

27.138 gdcm::Image Class Reference

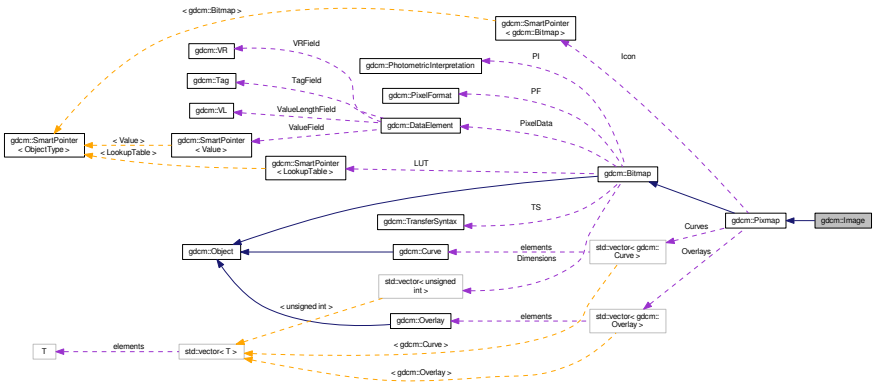
Image This is the container for an Image in the general sense. From this container you should be able to request information like:

```
#include <gdcmImage.h>
```

Inheritance diagram for gdcm::Image:



Collaboration diagram for gdcm::Image:



Public Member Functions

- Image ()
- ~Image ()
- 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

  print
- void SetDirectionCosines (const float *dircos)
- void SetDirectionCosines (const double *dircos)
- void SetDirectionCosines (unsigned int idx, double dircos)
- void SetIntercept (double intercept)

  intercept
- void SetOrigin (const float *ori)
- void SetOrigin (const double *ori)
- void SetOrigin (unsigned int idx, double ori)
- void SetSlope (double slope)

  slope
- void SetSpacing (const double *spacing)
- void SetSpacing (unsigned int idx, double spacing)

Additional Inherited Members

27.138.1 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:

CompressImage.cxx, ConvertToQImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GetJPEGSamplePrecision.cxx, GetSub↩SequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

27.138.2 Constructor & Destructor Documentation

27.138.2.1 gdcm::Image::Image ( ) [inline]

27.138.2.2 gdcm::Image::~Image ( ) [inline]

27.138.3 Member Function Documentation

27.138.3.1 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.

27.138.3.2 double gdcm::Image::GetDirectionCosines (unsigned int *idx*) const

27.138.3.3 double gdcm::Image::GetIntercept ( ) const [inline]

27.138.3.4 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.

27.138.3.5 double gdcm::Image::GetOrigin (unsigned int *idx*) const

27.138.3.6 double gdcm::Image::GetSlope ( ) const [inline]

27.138.3.7 const double* gdcm::Image::GetSpacing ( ) const

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an aribtrary 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

27.138.3.8 double gdcm::Image::GetSpacing (unsigned int *idx*) const

27.138.3.9 void gdcm::Image::Print (std::ostream & *os*) const [virtual]

print

Reimplemented from gdcm::Bitmap.

Examples:

CompressImage.cxx, and PatchFile.cxx.

27.138.3.10 void gdcm::Image::SetDirectionCosines (const float * *dircos*)

27.138.3.11 void gdcm::Image::SetDirectionCosines (const double * *dircos*)

27.138.3.12 void gdcm::Image::SetDirectionCosines (unsigned int *idx,* double *dircos*)

27.138.3.13 void gdcm::Image::SetIntercept (double *intercept*) [inline]

intercept

27.138.3.14 void gdcm::Image::SetOrigin (const float * *ori*)

27.138.3.15 void gdcm::Image::SetOrigin (const double * *ori*)

27.138.3.16 void gdcm::Image::SetOrigin (unsigned int *idx,* double *ori*)

27.138.3.17 void gdcm::Image::SetSlope (double *slope*) [inline]

slope

27.138.3.18 void gdcm::Image::SetSpacing (const double * *spacing*)

Examples:

csa2img.cxx, and iU22tomultisc.cxx.

27.138.3.19 void gdcm::Image::SetSpacing (unsigned int *idx,* double *spacing*)

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

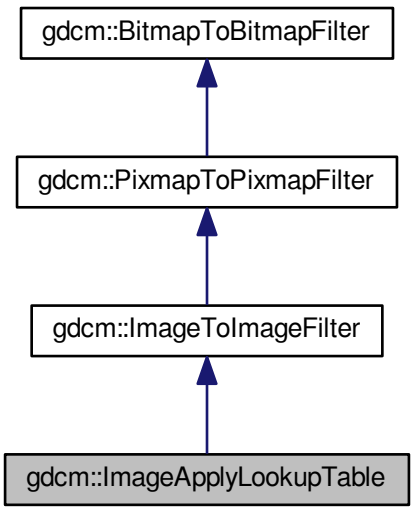
- gdcmImage.h

27.139 gdcm::ImageApplyLookupTable Class Reference

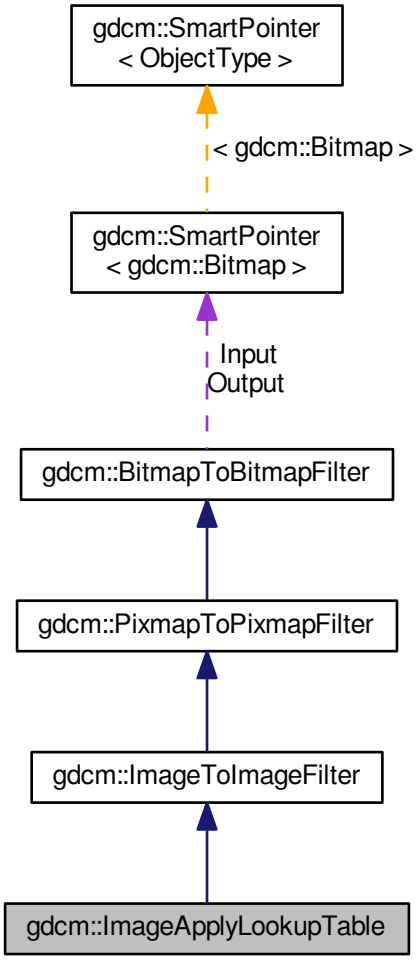
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

`#include <gdcmImageApplyLookupTable.h>`

Inheritance diagram for gdcm::ImageApplyLookupTable:



Collaboration diagram for gdcm::ImageApplyLookupTable:



Public Member Functions

- ImageApplyLookupTable ()
- ~ImageApplyLookupTable ()
- bool Apply ()

  Apply.

Additional Inherited Members

27.139.1 Detailed Description

ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

27.139.2 Constructor & Destructor Documentation

27.139.2.1 gdcm::ImageApplyLookupTable::ImageApplyLookupTable ( ) [inline]

27.139.2.2 gdcm::ImageApplyLookupTable::~ImageApplyLookupTable ( ) [inline]

27.139.3 Member Function Documentation

27.139.3.1 bool gdcm::ImageApplyLookupTable::Apply ( )

Apply.

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

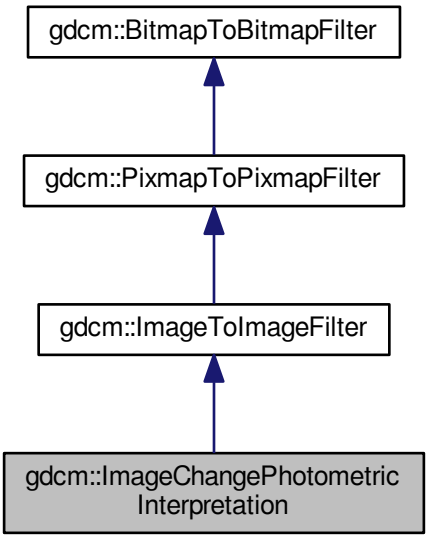
- gdcmImageApplyLookupTable.h

27.140 gdcm::ImageChangePhotometricInterpretation Class Reference

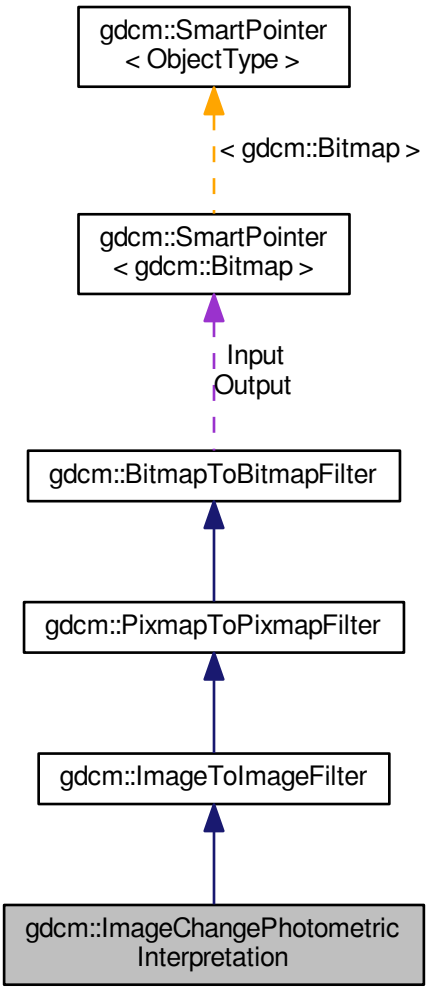
ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.

`#include <gdcmImageChangePhotometricInterpretation.h>`

Inheritance diagram for gdcm::ImageChangePhotometricInterpretation:



Collaboration diagram for gdcm::ImageChangePhotometricInterpretation:



Public Member Functions

- ImageChangePhotometricInterpretation ()
- ~ImageChangePhotometricInterpretation ()
- bool Change ()

  Change.
- const PhotometricInterpretation & GetPhotometricInterpretation () const
- void SetPhotometricInterpretation (PhotometricInterpretation const &pi)

  Set/Get requested PhotometricInterpretation.

Static Public Member Functions

- template<typename T >
  static void RGB2YBR (T ybr[3], const T rgb[3])

  colorspace converstion (based on CCIR Recommendation 601-2)
- template<typename T >
  static void YBR2RGB (T rgb[3], const T ybr[3])

Protected Member Functions

- bool ChangeMonochrome ()

Additional Inherited Members

27.140.1 Detailed Description

ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.

27.140.2 Constructor & Destructor Documentation

27.140.2.1 gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ( ) [inline]

27.140.2.2 gdcm::ImageChangePhotometricInterpretation::~ImageChangePhotometricInterpretation ( ) [inline]

27.140.3 Member Function Documentation

27.140.3.1 bool gdcm::ImageChangePhotometricInterpretation::Change ( )

Change.

27.140.3.2 bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( ) [protected]

27.140.3.3 const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation ( ) const [inline]

27.140.3.4 template<typename T > void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (T *ybr[3]*, const T *rgb[3]*) [static]

colorspace converstion (based on CCIR Recommendation 601-2)

27.140.3.5 void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*) [inline]

Set/Get requested PhotometricInterpretation.

27.140.3.6 template<typename T > void gdcm::ImageChangePhotometricInterpretation::YBR2RGB (T *rgb[3]*, const T *ybr[3]*) [static]

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

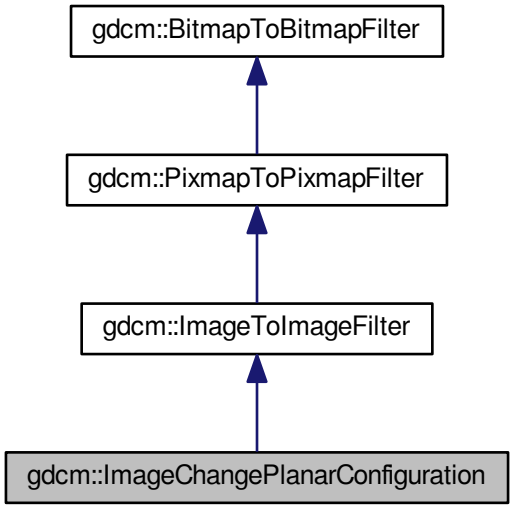
- gdcmImageChangePhotometricInterpretation.h

27.141 gdcm::ImageChangePlanarConfiguration Class Reference

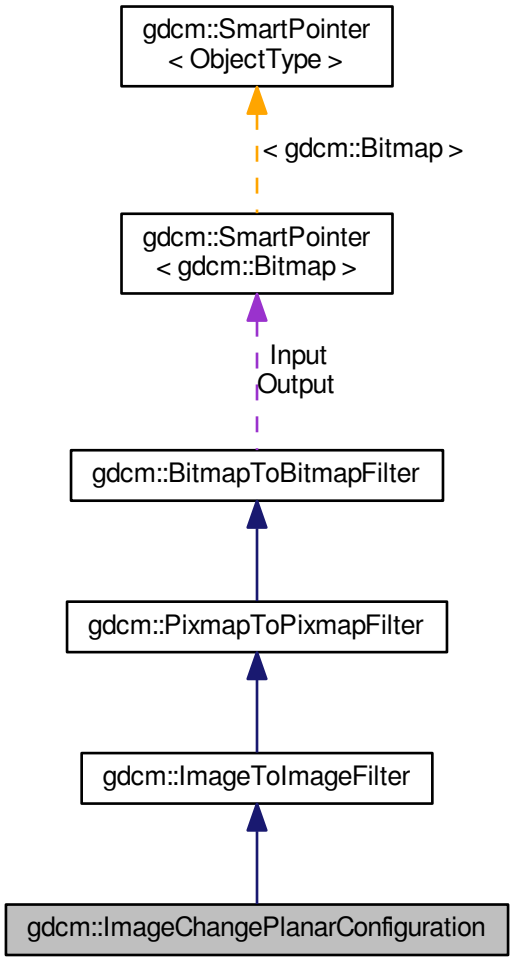
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.

`#include <gdcmImageChangePlanarConfiguration.h>`

Inheritance diagram for gdcm::ImageChangePlanarConfiguration:



Collaboration diagram for gdcm::ImageChangePlanarConfiguration:



Public Member Functions

- ImageChangePlanarConfiguration ()
- ~ImageChangePlanarConfiguration ()
- bool Change ()

  Change.
- unsigned int GetPlanarConfiguration () const
- void SetPlanarConfiguration (unsigned int pc)

  Set/Get requested PlanarConfigation.

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

27.141.1 Detailed Description

ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.

27.141.2 Constructor & Destructor Documentation

27.141.2.1 gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ( ) [inline]

27.141.2.2 gdcm::ImageChangePlanarConfiguration::~ImageChangePlanarConfiguration ( ) [inline]

27.141.3 Member Function Documentation

27.141.3.1 bool gdcm::ImageChangePlanarConfiguration::Change ( )

Change.

27.141.3.2 unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const [inline]

27.141.3.3 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

27.141.3.4 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

27.141.3.5 void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (unsigned int *pc*) [inline]

Set/Get requested PlanarConfigation.

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

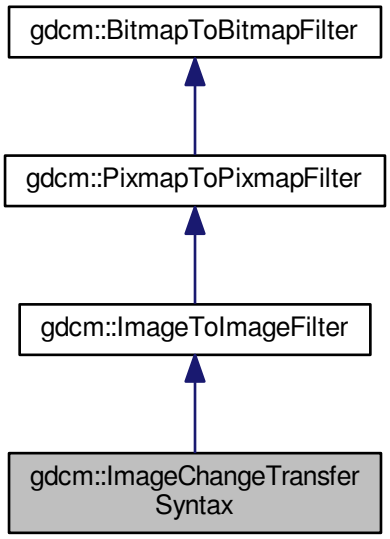
- gdcmImageChangePlanarConfiguration.h

27.142 gdcm::ImageChangeTransferSyntax Class Reference

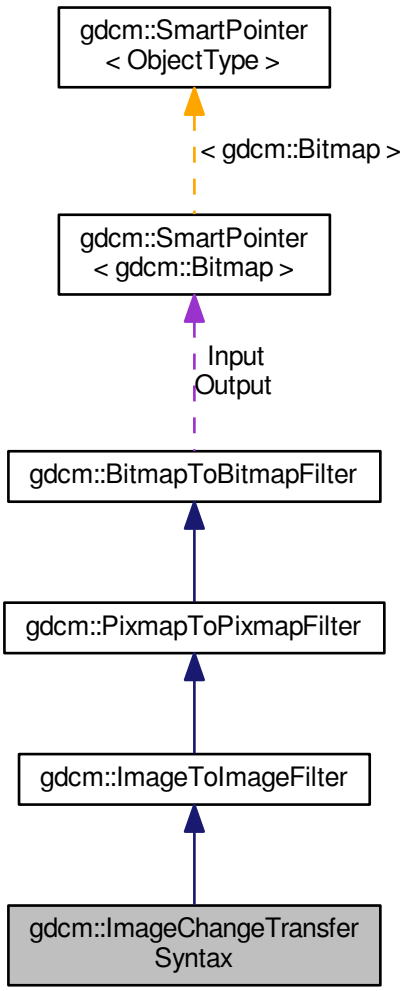
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.

`#include <gdcmImageChangeTransferSyntax.h>`

Inheritance diagram for gdcm::ImageChangeTransferSyntax:



Collaboration diagram for gdcm::ImageChangeTransferSyntax:



Public Member Functions

- ImageChangeTransferSyntax ()
- ~ImageChangeTransferSyntax ()
- 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)

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

27.142.1 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:

CompressImage.cxx.

27.142.2 Constructor & Destructor Documentation

27.142.2.1 gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ( ) [inline]

27.142.2.2 gdcm::ImageChangeTransferSyntax::~ImageChangeTransferSyntax ( ) [inline]

27.142.3 Member Function Documentation

27.142.3.1 bool gdcm::ImageChangeTransferSyntax::Change ( )

Change.

Examples:

CompressImage.cxx.

27.142.3.2 const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const [inline]

Get Transfer Syntax.

27.142.3.3 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

27.142.3.4 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

27.142.3.5 void gdcm::ImageChangeTransferSyntax::SetTransferSyntax (const TransferSyntax & *ts*) [inline]

Set target Transfer Syntax.

Examples:

CompressImage.cxx.

27.142.3.6 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)

27.142.3.7 bool gdcm::ImageChangeTransferSyntax::TryJPEG2000Codec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

27.142.3.8 bool gdcm::ImageChangeTransferSyntax::TryJPEGCodec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

27.142.3.9 bool gdcm::ImageChangeTransferSyntax::TryJPEGLSCodec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

27.142.3.10 bool gdcm::ImageChangeTransferSyntax::TryRAWCodec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

27.142.3.11 bool gdcm::ImageChangeTransferSyntax::TryRLECodec (const DataElement & *pixelde*, Bitmap const & *input*, Bitmap & *output*) [protected]

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

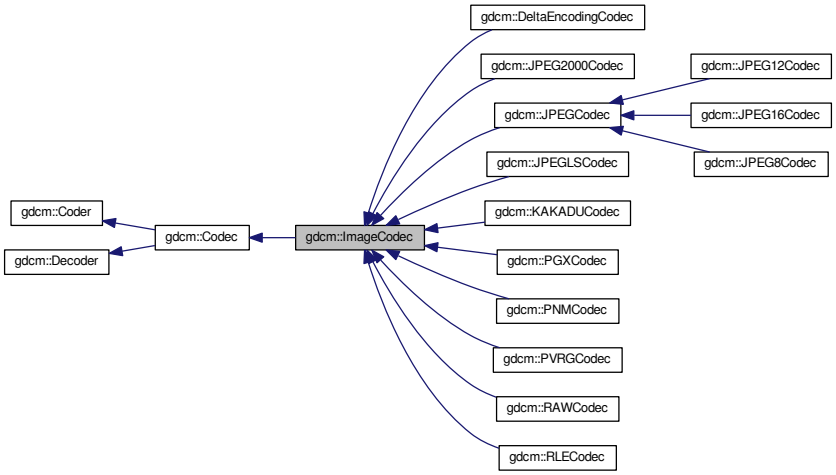
- gdcmImageChangeTransferSyntax.h

27.143 gdcm::ImageCodec Class Reference

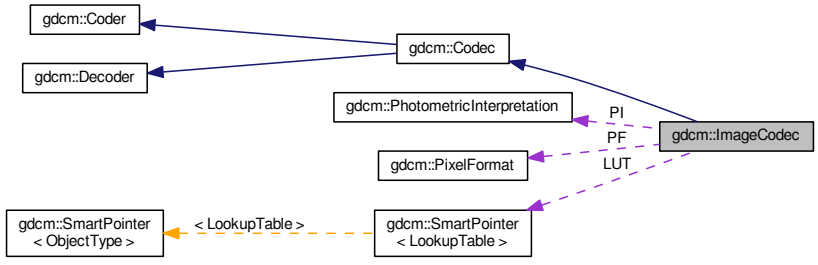
ImageCodec.

`#include <gdcmImageCodec.h>`

Inheritance diagram for gdcm::ImageCodec:



Collaboration diagram for gdcm::ImageCodec:



Public Member Functions

- ImageCodec ()
- ~ImageCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const =0

- bool Decode (DataElement const &is_, DataElement &os)

  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 unsigned int d[3])
- void SetDimensions (const std::vector< unsigned int > &d)
- 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)

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)
- 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)
- 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 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

27.143.1 Detailed Description

ImageCodec.

Note

Main codec, this is a central place for all implementation

27.143.2 Member Typedef Documentation

27.143.2.1 typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr [protected]

27.143.3 Constructor & Destructor Documentation

27.143.3.1 gdcm::ImageCodec::ImageCodec ( )

27.143.3.2 gdcm::ImageCodec::~ImageCodec ( )

27.143.4 Member Function Documentation

27.143.4.1 virtual bool gdcm::ImageCodec::AppendFrameEncode (std::ostream & *out,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

27.143.4.2 virtual bool gdcm::ImageCodec::AppendRowEncode (std::ostream & *out,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

27.143.4.3 bool gdcm::ImageCodec::CanCode (TransferSyntax const &) const [inline], [virtual]

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::JPEGLSCodec, gdcm::PNMCodec, gdcm::PGXCodec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

27.143.4.4 bool gdcm::ImageCodec::CanDecode (TransferSyntax const &) const [inline], [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::JPEGLSCodec, gdcm::PNMCodec, gdcm::RAWCodec, gdcm::PGXCodec, and gdcm::KAKADUCodec.

27.143.4.5 virtual ImageCodec* gdcm::ImageCodec::Clone ( ) const [pure virtual]

Implemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::PNMCodec, gdcm::RAWCodec, gdcm::KAKADUCodec, and gdcm::PGXCodec.

27.143.4.6 bool gdcm::ImageCodec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::PVRGCodec, gdcm::JPEG2000Codec, gdcm::KAKADUCodec, and gdcm::RAWCodec.

27.143.4.7 bool gdcm::ImageCodec::DecodeByStreams (std::istream & *is_*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::Decoder.

Reimplemented in gdcm::JPEGCodec, gdcm::JPEG2000Codec, gdcm::RLECodec, gdcm::RAWCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

27.143.4.8 bool gdcm::ImageCodec::DoByteSwap (std::istream & *is_*, std::ostream & *os*) [protected]

27.143.4.9 bool gdcm::ImageCodec::DoInvertMonochrome (std::istream & *is_*, std::ostream & *os*) [protected]

27.143.4.10 bool gdcm::ImageCodec::DoOverlayCleanup (std::istream & *is_*, std::ostream & *os*) [protected]

27.143.4.11 bool gdcm::ImageCodec::DoPaddedCompositePixelCode (std::istream & *is_*, std::ostream & *os*) [protected]

27.143.4.12 bool gdcm::ImageCodec::DoPlanarConfiguration (std::istream & *is_*, std::ostream & *os*) [protected]

27.143.4.13 bool gdcm::ImageCodec::DoSimpleCopy (std::istream & *is_*, std::ostream & *os*) [protected]

27.143.4.14 bool gdcm::ImageCodec::DoYBR (std::istream & *is_*, std::ostream & *os*) [protected]

27.143.4.15 const unsigned int* gdcm::ImageCodec::GetDimensions ( ) const [inline]

27.143.4.16 virtual bool gdcm::ImageCodec::GetHeaderInfo (std::istream & *is_*, TransferSyntax & *ts*) [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::RLECodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, gdcm::PNMCodec, gdcm::JPEG12Codec, gdcm::JPEG16Codec, gdcm::JPEG8Codec, gdcm::RAWCodec, and gdcm::PGXCodec.

27.143.4.17 bool gdcm::ImageCodec::GetLossyFlag ( ) const

27.143.4.18 const LookupTable& gdcm::ImageCodec::GetLUT ( ) const [inline]

27.143.4.19 bool gdcm::ImageCodec::GetNeedByteSwap ( ) const [inline]

27.143.4.20 unsigned int gdcm::ImageCodec::GetNumberOfDimensions ( ) const

27.143.4.21 const PhotometricInterpretation& gdcm::ImageCodec::GetPhotometricInterpretation ( ) const

27.143.4.22 PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) [inline]

Examples:

GetJPEGSamplePrecision.cxx.

27.143.4.23 const PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) const [inline]

27.143.4.24 unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const [inline]

27.143.4.25 virtual bool gdcm::ImageCodec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

27.143.4.26 bool gdcm::ImageCodec::IsLossy ( ) const

27.143.4.27 virtual bool gdcm::ImageCodec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

27.143.4.28 virtual bool gdcm::ImageCodec::IsValid (PhotometricInterpretation const & *pi*) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec.

27.143.4.29 void gdcm::ImageCodec::SetDimensions (const unsigned int *d[3]*)

Examples:

ExtractIconFromFile.cxx.

27.143.4.30 void gdcm::ImageCodec::SetDimensions (const std::vector< unsigned int > & *d*)

27.143.4.31 void gdcm::ImageCodec::SetLossyFlag (bool *l*)

27.143.4.32 void gdcm::ImageCodec::SetLUT (LookupTable const & *lut*) [inline]

Examples:

ExtractIconFromFile.cxx.

27.143.4.33 void gdcm::ImageCodec::SetNeedByteSwap (bool *b*) [inline]

27.143.4.34 void gdcm::ImageCodec::SetNeedOverlayCleanup (bool *b*) [inline]

27.143.4.35 void gdcm::ImageCodec::SetNumberOfDimensions (unsigned int *dim*)

27.143.4.36 void gdcm::ImageCodec::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*)

Examples:

ExtractIconFromFile.cxx.

27.143.4.37 virtual void gdcm::ImageCodec::SetPixelFormat (PixelFormat const & *pf*) [inline], [virtual]

Reimplemented in gdcm::JPEGCodec.

Examples:

ExtractIconFromFile.cxx.

27.143.4.38 void gdcm::ImageCodec::SetPlanarConfiguration (unsigned int *pc*) [inline]

27.143.4.39 virtual bool gdcm::ImageCodec::StartEncode (std::ostream & *os*) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

27.143.4.40 virtual bool gdcm::ImageCodec::StopEncode (std::ostream & *os*) [protected], [virtual]

Reimplemented in gdcm::JPEGCodec, gdcm::JPEGLSCodec, gdcm::JPEG2000Codec, and gdcm::RLECodec.

27.143.5 Friends And Related Function Documentation

27.143.5.1 friend class FileChangeTransferSyntax [friend]

This is a high level API to encode in a streaming fashion. Each plugin will handle differently the caching mecanism 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

27.143.5.2 friend class ImageChangePhotometricInterpretation [friend]

27.143.6 Member Data Documentation

27.143.6.1 unsigned int gdcm::ImageCodec::Dimensions[3] [protected]

27.143.6.2 bool gdcm::ImageCodec::LossyFlag [protected]

27.143.6.3 LUTPtr gdcm::ImageCodec::LUT [protected]

27.143.6.4 bool gdcm::ImageCodec::NeedByteSwap [protected]

27.143.6.5 bool gdcm::ImageCodec::NeedOverlayCleanup [protected]

27.143.6.6 unsigned int gdcm::ImageCodec::NumberOfDimensions [protected]

27.143.6.7 PixelFormat gdcm::ImageCodec::PF [protected]

27.143.6.8 PhotometricInterpretation gdcm::ImageCodec::PI [protected]

27.143.6.9 unsigned int gdcm::ImageCodec::PlanarConfiguration [protected]

27.143.6.10 bool gdcm::ImageCodec::RequestPaddedCompositePixelCode [protected]

27.143.6.11 bool gdcm::ImageCodec::RequestPlanarConfiguration [protected]

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

- gdcmImageCodec.h

27.144 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)

27.144.1 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

27.144.2 Constructor & Destructor Documentation

27.144.2.1 gdcm::ImageConverter::ImageConverter ( )

27.144.2.2 gdcm::ImageConverter::~ImageConverter ( )

27.144.3 Member Function Documentation

27.144.3.1 void gdcm::ImageConverter::Convert ( )

27.144.3.2 const Image& gdcm::ImageConverter::GetOuput ( ) const

27.144.3.3 void gdcm::ImageConverter::SetInput (Image const & *input*)

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

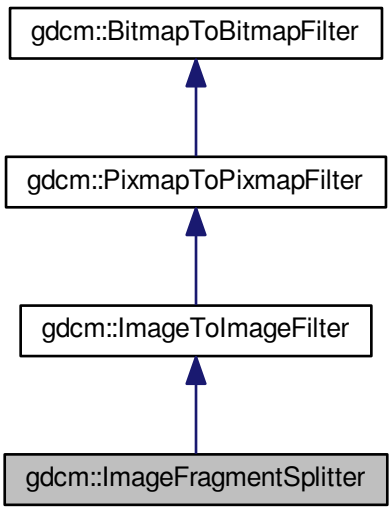
- gdcmImageConverter.h

27.145 gdcm::ImageFragmentSplitter Class Reference

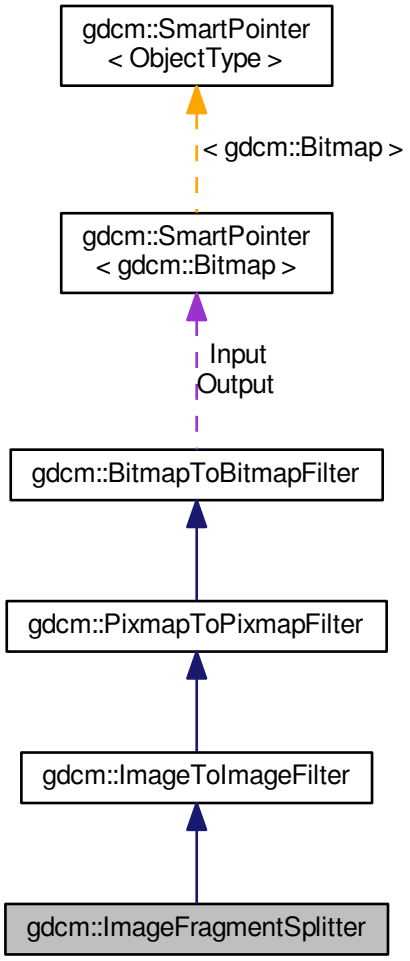
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

`#include <gdcmImageFragmentSplitter.h>`

Inheritance diagram for gdcm::ImageFragmentSplitter:



Collaboration diagram for gdcm::ImageFragmentSplitter:



Public Member Functions

- ImageFragmentSplitter ()
- ~ImageFragmentSplitter ()
- unsigned int GetFragmentSizeMax () const
- void SetForce (bool f)
- void SetFragmentSizeMax (unsigned int fragsize)

  FragmentSizeMax needs to be an even number.
- bool Split ()

  Split.

Additional Inherited Members

27.145.1 Detailed Description

ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

27.145.2 Constructor & Destructor Documentation

27.145.2.1 gdcm::ImageFragmentSplitter::ImageFragmentSplitter ( ) [inline]

27.145.2.2 gdcm::ImageFragmentSplitter::~ImageFragmentSplitter ( ) [inline]

27.145.3 Member Function Documentation

27.145.3.1 unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const [inline]

27.145.3.2 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 !

27.145.3.3 void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (unsigned int *fragsize*)

FragmentSizeMax needs to be an even number.

27.145.3.4 bool gdcm::ImageFragmentSplitter::Split ( )

Split.

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

- gdcmImageFragmentSplitter.h

27.146 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)
- 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 const ByteValue * GetPointerFromElement (Tag const &tag, File const &f)

  Moved from PixampReader to here. Generally used for photometric interpretation.
- static std::vector< double > GetRescaleInterceptSlopeValue (File const &f)
- static std::vector< double > GetSpacingValue (File const &f)

  Set/Get Spacing from/to a File.
- static void SetDimensionsValue (File &f, const Image &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 SetRescaleInterceptSlopeValue (File &f, const Image &img)
- 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)

27.146.1 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.

27.146.2 Member Function Documentation

27.146.2.1 static 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.

27.146.2.2 static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient (const std::vector< double > & *imageposition,* std::vector< double > & *spacing*) [static]

DO NOT USE.

27.146.2.3 static 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:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

27.146.2.4 static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (DataSet const & *ds,* std::vector< double > & *dircos*) [static]

27.146.2.5 static 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

27.146.2.6 static bool gdcm::ImageHelper::GetForcePixelSpacing ( ) [static]

27.146.2.7 static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( ) [static]

27.146.2.8 static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (File const & *f*) [static]

27.146.2.9 static std::vector<double> gdcm::ImageHelper::GetOriginValue (File const & *f*) [static]

Set/Get Origin (IPP) from/to a file.

27.146.2.10 static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (File const & *f*) [static]

27.146.2.11 static 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)

27.146.2.12 static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (const File & *f*) [static]

27.146.2.13 static const ByteValue* gdcm::ImageHelper::GetPointerFromElement (Tag const & *tag,* File const & *f*) [static]

Moved from PixampReader to here. Generally used for photometric interpretation.

27.146.2.14 static 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

27.146.2.15 **static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage (MediaStorage const & *ms*)** [static], [protected]

27.146.2.16 **static std::vector<double> gdcm::ImageHelper::GetSpacingValue (File const & *f*)** [static]

Set/Get Spacing from/to a File.

27.146.2.17 **static Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage (MediaStorage const & *ms*)** [static], [protected]

27.146.2.18 **static void gdcm::ImageHelper::SetDimensionsValue (File & *f,* const Image & *img*)** [static]

27.146.2.19 **static 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.

27.146.2.20 **static 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 explicitely set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel Spacing of the Image

27.146.2.21 **static void gdcm::ImageHelper::SetForceRescaleInterceptSlope (bool)** [static]

GDCM 1.x compatibility issue: when using ReWrite an MR Image Storage would be rewritten with a Rescale Slope/↩ Intercept while the standard would prohibit this (Philips Medical System is still doing that) Unless explicitely set elsewhere by the standard, it will use value from 0028,1052 / 0028,1053 for the Rescale Slope & Rescale Intercept values

27.146.2.22 **static void gdcm::ImageHelper::SetOriginValue (DataSet & *ds,* const Image & *img*)** [static]

27.146.2.23 **static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue (File & *f,* const Image & *img*)** [static]

27.146.2.24 **static void gdcm::ImageHelper::SetSpacingValue (DataSet & *ds,* const std::vector< double > & *spacing*)** [static]

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

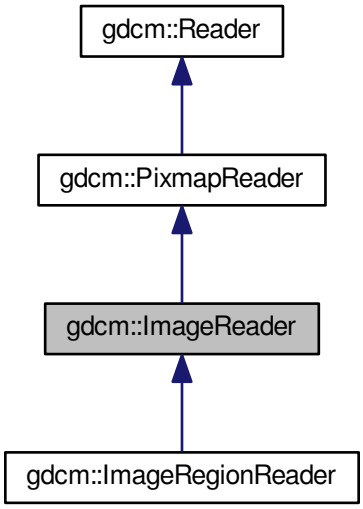
- gdcmImageHelper.h

27.147 gdcm::ImageReader Class Reference

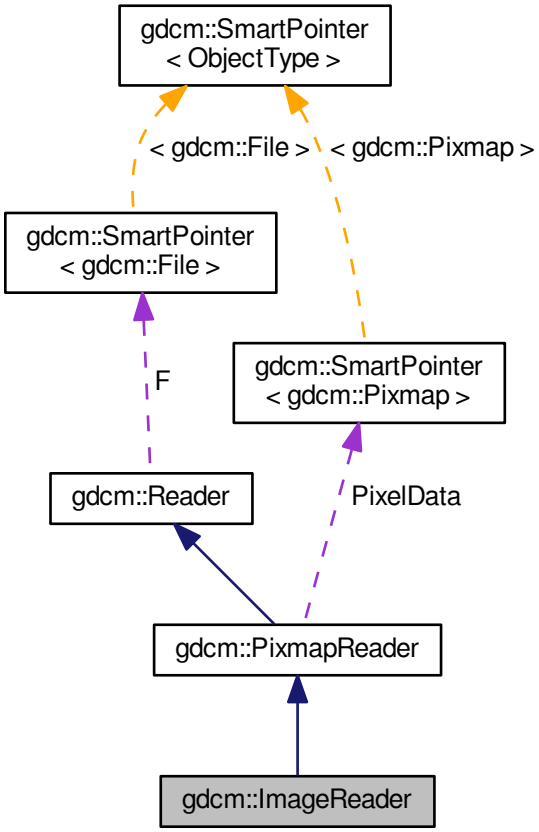
ImageReader.

`#include <gdcmImageReader.h>`

Inheritance diagram for gdcm::ImageReader:



Collaboration diagram for gdcm::ImageReader:



Public Member Functions

- ImageReader ()
- virtual ~ImageReader ()
- const Image & GetImage () const

  Return the read image.
- Image & GetImage ()
- virtual bool Read ()

Protected Member Functions

- bool ReadACRNEMAImage ()
- bool ReadImage (MediaStorage const &ms)

Additional Inherited Members

27.147.1 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:

CheckBigEndianBug.cxx, CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

27.147.2 Constructor & Destructor Documentation

27.147.2.1 gdcm::ImageReader::ImageReader ( )

27.147.2.2 virtual gdcm::ImageReader::~ImageReader ( ) [virtual]

27.147.3 Member Function Documentation

27.147.3.1 const Image& gdcm::ImageReader::GetImage ( ) const

Return the read image.

Examples:

CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

27.147.3.2 Image& gdcm::ImageReader::GetImage ( )

27.147.3.3 virtual bool gdcm::ImageReader::Read ( ) [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::PixmapReader.

Reimplemented in gdcm::ImageRegionReader.

Examples:

CheckBigEndianBug.cxx, CompressImage.cxx, ConvertToQImage.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSamplePrecision.cxx, HelloVizWorld.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, ReadMultiTimesException.cxx, and threadgdcm.cxx.

27.147.3.4 bool gdcm::ImageReader::ReadACRNEMAImage ( ) [protected], [virtual]

Reimplemented from gdcm::PixmapReader.

27.147.3.5 bool gdcm::ImageReader::ReadImage (MediaStorage const & *ms*) [protected], [virtual]

Reimplemented from gdcm::PixmapReader.

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

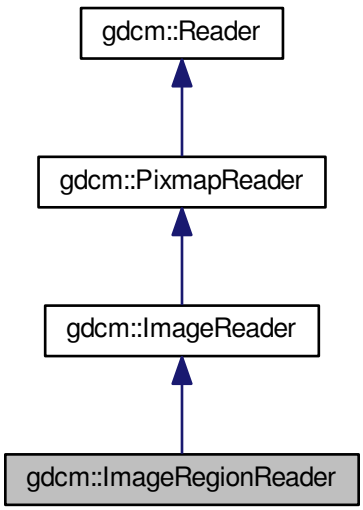
- gdcmImageReader.h

27.148 gdcm::ImageRegionReader Class Reference

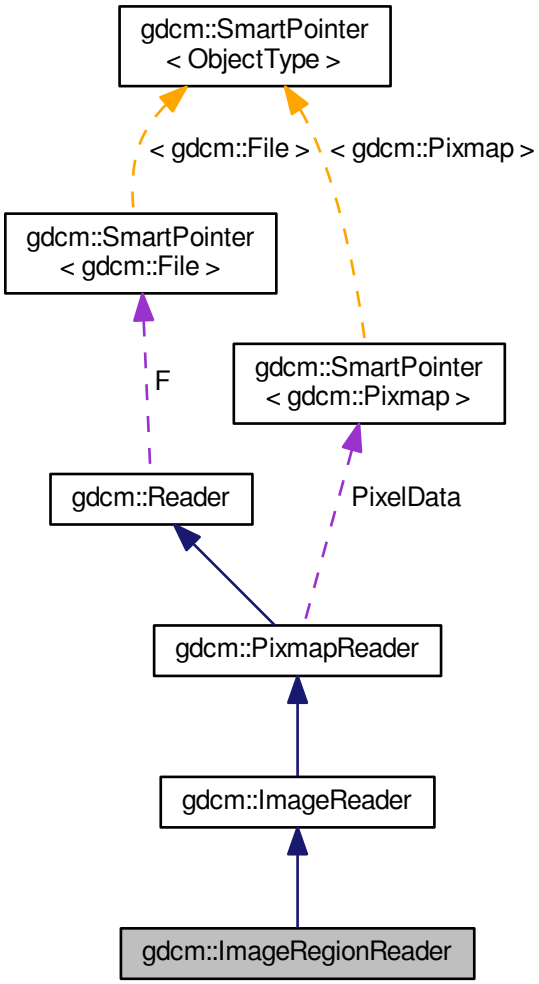
ImageRegionReader.

`#include <gdcmImageRegionReader.h>`

Inheritance diagram for gdcm::ImageRegionReader:



Collaboration diagram for gdcm::ImageRegionReader:



Public Member Functions

- ImageRegionReader ()
- ~ImageRegionReader ()
- 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.

Protected Member Functions

- bool Read ()

  To prevent user from calling super class Read() function.

Additional Inherited Members

27.148.1 Detailed Description

ImageRegionReader.

See also

ImageReader

27.148.2 Constructor & Destructor Documentation

27.148.2.1 gdcm::ImageRegionReader::ImageRegionReader ( )

27.148.2.2 gdcm::ImageRegionReader::~ImageRegionReader ( )

27.148.3 Member Function Documentation

27.148.3.1 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

27.148.3.2 Region const& gdcm::ImageRegionReader::GetRegion ( ) const

27.148.3.3 bool gdcm::ImageRegionReader::Read ( ) [protected], [virtual]

To prevent user from calling super class Read() function.

Reimplemented from gdcm::ImageReader.

27.148.3.4 bool gdcm::ImageRegionReader::ReadInformation ( )

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

27.148.3.5 bool gdcm::ImageRegionReader::ReadIntoBuffer (char * *inreadbuffer,* size_t *buflen*)

Read into buffer:

Returns

false upon error

27.148.3.6 void gdcm::ImageRegionReader::SetRegion (Region const & *region*)

Set/Get Region to be read.

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

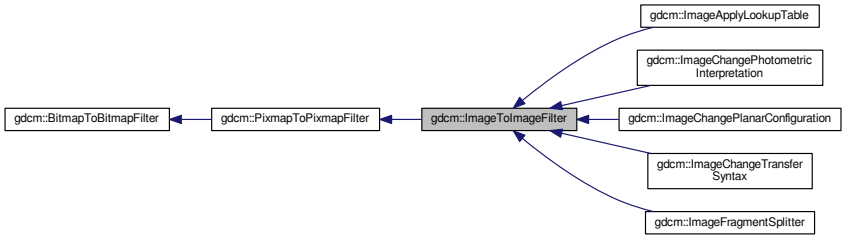
- gdcmImageRegionReader.h

27.149 gdcm::ImageToImageFilter Class Reference

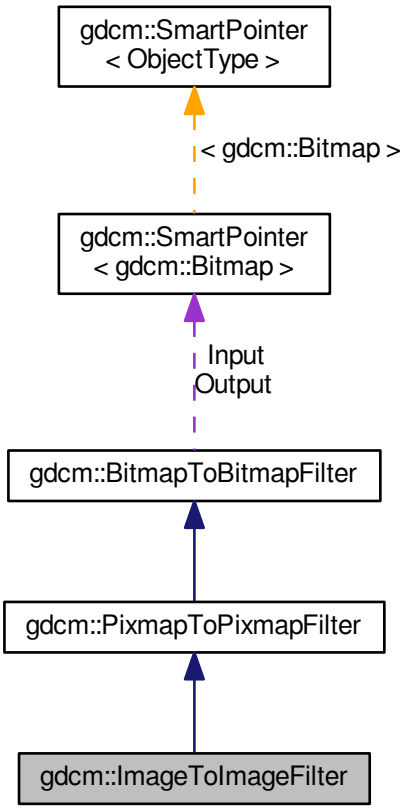
ImageToImageFilter class Super class for all filter taking an image and producing an output image.

`#include <gdcmImageToImageFilter.h>`

Inheritance diagram for gdcm::ImageToImageFilter:



Collaboration diagram for gdcm::ImageToImageFilter:



Public Member Functions

- ImageToImageFilter ()
- ~ImageToImageFilter ()
- Image & GetInput ()
- const Image & GetOutput () const

  Get Output image.

Additional Inherited Members

27.149.1 Detailed Description

ImageToImageFilter class Super class for all filter taking an image and producing an output image.

27.149.2 Constructor & Destructor Documentation

27.149.2.1 gdcm::ImageToImageFilter::ImageToImageFilter ( )

27.149.2.2 gdcm::ImageToImageFilter::~ImageToImageFilter ( ) [inline]

27.149.3 Member Function Documentation

27.149.3.1 Image& gdcm::ImageToImageFilter::GetInput ( )

27.149.3.2 const Image& gdcm::ImageToImageFilter::GetOutput ( ) const

Get Output image.

Examples:

CompressImage.cxx.

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

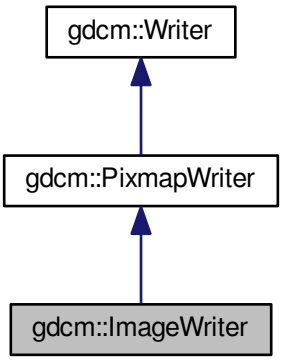
- gdcmImageToImageFilter.h

27.150 gdcm::ImageWriter Class Reference

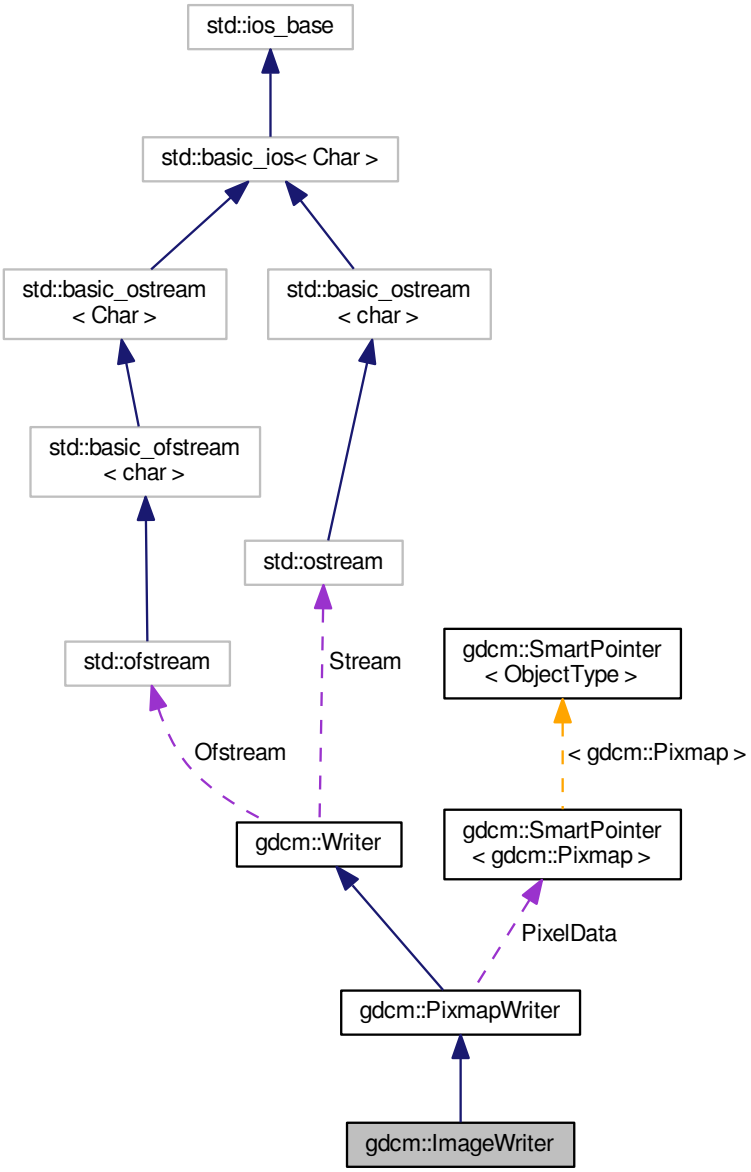
ImageWriter.

`#include <gdcmImageWriter.h>`

Inheritance diagram for gdcm::ImageWriter:



Collaboration diagram for gdcm::ImageWriter:



Public Member Functions

- ImageWriter ()
- ~ImageWriter ()
- const Image & GetImage () const
- Image & GetImage ()
- bool Write ()

Write.

Additional Inherited Members

27.150.1 Detailed Description

ImageWriter.

Examples:

CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, and MergeTwoFiles.cxx.

27.150.2 Constructor & Destructor Documentation

27.150.2.1 gdcm::ImageWriter::ImageWriter ( )

27.150.2.2 gdcm::ImageWriter::~ImageWriter ( )

27.150.3 Member Function Documentation

27.150.3.1 const Image& gdcm::ImageWriter::GetImage ( ) const [inline], [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.

27.150.3.2 Image& gdcm::ImageWriter::GetImage ( ) [inline], [virtual]

Reimplemented from gdcm::PixmapWriter.

27.150.3.3 bool gdcm::ImageWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::Writer.

Examples:

CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, iU22tomultisc.cxx, and MergeTwoFiles.cxx.

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

- gdcmImageWriter.h

27.151 gdcm::network::ImplementationClassUIDSub Class Reference

ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIAT↩ E-RQ)

`#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

27.151.1 Detailed Description

ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIAT↩ E-RQ)

27.151.2 Constructor & Destructor Documentation

27.151.2.1 gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )

27.151.3 Member Function Documentation

27.151.3.1 void gdcm::network::ImplementationClassUIDSub::Print (std::ostream & *os*) const

27.151.3.2 std::istream& gdcm::network::ImplementationClassUIDSub::Read (std::istream & *is*)

27.151.3.3 size_t gdcm::network::ImplementationClassUIDSub::Size ( ) const

27.151.3.4 const std::ostream& gdcm::network::ImplementationClassUIDSub::Write (std::ostream & *os*) const

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

- gdcmImplementationClassUIDSub.h

27.152 gdcm::network::ImplementationUIDSub Class Reference

ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

`#include <gdcmImplementationUIDSub.h>`

Public Member Functions

- ImplementationUIDSub ()
- const std::ostream & Write (std::ostream &os) const

27.152.1 Detailed Description

ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

27.152.2 Constructor & Destructor Documentation

27.152.2.1 gdcm::network::ImplementationUIDSub::ImplementationUIDSub ( )

27.152.3 Member Function Documentation

27.152.3.1 const std::ostream& gdcm::network::ImplementationUIDSub::Write (std::ostream & *os*) const

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

- gdcmImplementationUIDSub.h

27.153 gdcm::network::ImplementationVersionNameSub Class Reference

ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCI↩ ATE-RQ)

`#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

27.153.1 Detailed Description

ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCI↩ ATE-RQ)

27.153.2 Constructor & Destructor Documentation

27.153.2.1 gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )

27.153.3 Member Function Documentation

27.153.3.1 void gdcm::network::ImplementationVersionNameSub::Print (std::ostream & *os*) const

27.153.3.2 std::istream& gdcm::network::ImplementationVersionNameSub::Read (std::istream & *is*)

27.153.3.3 size_t gdcm::network::ImplementationVersionNameSub::Size ( ) const

27.153.3.4 const std::ostream& gdcm::network::ImplementationVersionNameSub::Write (std::ostream & *os*) const

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

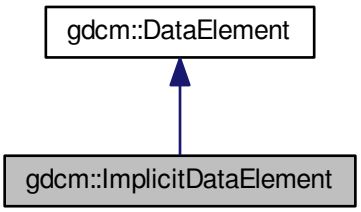
- gdcmImplementationVersionNameSub.h

27.154 gdcm::ImplicitDataElement Class Reference

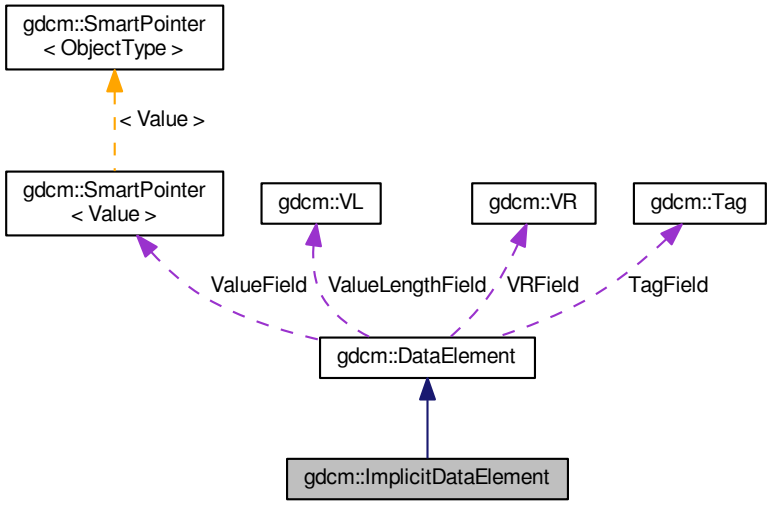
Class to represent an *Implicit VR* Data Element.

`#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

Additional Inherited Members

27.154.1 Detailed Description

Class to represent an *Implicit VR* Data Element.

Note

bla

Examples:

ReadExplicitLengthSQIVR.cxx.

27.154.2 Member Function Documentation

27.154.2.1 VL gdcm::ImplicitDataElement::GetLength ( ) const

27.154.2.2 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::Read (std::istream & *is*)

27.154.2.3 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadPreValue (std::istream & *is*)

27.154.2.4 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValue (std::istream & *is*, bool *readvalues* = `true`)

27.154.2.5 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValueWithLength (std::istream & *is*, VL & *length*, bool *readvalues* = `true`)

27.154.2.6 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*, bool *readvalues* = `true`)

27.154.2.7 template<typename TSwap > const std::ostream& gdcm::ImplicitDataElement::Write (std::ostream & *os*) const

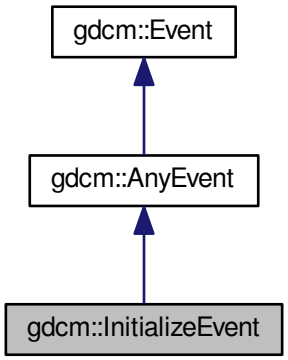
The documentation for this class was generated from the following file:

- gdcmImplicitDataElement.h

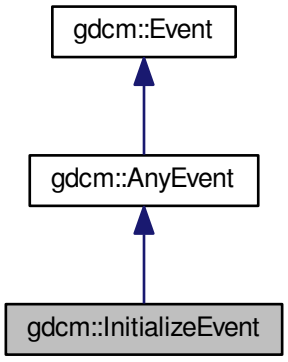
27.155 gdcm::InitializeEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::InitializeEvent:



Collaboration diagram for gdcm::InitializeEvent:



Additional Inherited Members

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

- gdcmEvent.h

27.156 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 ()
- 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)

27.156.1 Detailed Description

Class for representing a IOD.

Note

bla

See also

Dict

Examples:

TraverseModules.cxx.

27.156.2 Member Typedef Documentation

27.156.2.1 typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry

27.156.2.2 typedef MapIODEntry::size_type gdcm::IOD::SizeType

27.156.3 Constructor & Destructor Documentation

27.156.3.1 gdcm::IOD::IOD ( ) [inline]

27.156.4 Member Function Documentation

27.156.4.1 void gdcm::IOD::AddIODEntry (const IODEntry & *iode*) [inline]

27.156.4.2 void gdcm::IOD::Clear ( ) [inline]

27.156.4.3 const IODEntry& gdcm::IOD::GetIODEntry (SizeType *idx*) const [inline]

Examples:

TraverseModules.cxx.

27.156.4.4 SizeType gdcm::IOD::GetNumberOfIODs ( ) const [inline]

Examples:

TraverseModules.cxx.

27.156.4.5 Type gdcm::IOD::GetTypeFromTag (const Defs & *defs,* const Tag & *tag*) const

27.156.5 Friends And Related Function Documentation

27.156.5.1 std::ostream& operator<< (std::ostream & *_os,* const IOD & *_val*) [friend]

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

- gdcmIOD.h

27.157 gdcm::IODEntry Class Reference

Class for representing a IODEntry.

`#include <gdcmIODEntry.h>`

Public Member Functions

- IODEntry (const char *name="", const char *ref="", const char *usag="")
- 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 *usag)

Friends

- std::ostream & operator<< (std::ostream &_os, const IODEntry &_val)

27.157.1 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.

27.157.2 Constructor & Destructor Documentation

27.157.2.1 gdcm::IODEntry::IODEntry (const char * *name* = " ", const char * *ref* = " ", const char * *usag* = " ") [inline]

27.157.3 Member Function Documentation

27.157.3.1 const char* gdcm::IODEntry::GetIE ( ) const [inline]

27.157.3.2 const char* gdcm::IODEntry::GetName ( ) const [inline]

27.157.3.3 const char* gdcm::IODEntry::GetRef ( ) const [inline]

Examples:

TraverseModules.cxx.

27.157.3.4 const char* gdcm::IODEntry::GetUsage ( ) const [inline]

27.157.3.5 Usage::UsageType gdcm::IODEntry::GetUsageType ( ) const

27.157.3.6 void gdcm::IODEntry::SetIE (const char * *ie*) [inline]

27.157.3.7 void gdcm::IODEntry::SetName (const char * *name*) [inline]

27.157.3.8 void gdcm::IODEntry::SetRef (const char * *ref*) [inline]

27.157.3.9 void gdcm::IODEntry::SetUsage (const char * *usag*) [inline]

27.157.4 Friends And Related Function Documentation

27.157.4.1 std::ostream& operator<< (std::ostream & *_os,* const IODEntry & *_val*) [friend]

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

- gdcmIODEntry.h

27.158 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 ()
- 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)

27.158.1 Detailed Description

Class for representing a IODs.

Note

bla

See also

IOD

Examples:

TraverseModules.cxx.

27.158.2 Member Typedef Documentation

27.158.2.1 typedef std::map<IODName, IOD> gdcm::IODs::IODMapType

27.158.2.2 typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator

27.158.2.3 typedef std::string gdcm::IODs::IODName

27.158.3 Constructor & Destructor Documentation

27.158.3.1 gdcm::IODs::IODs ( ) [inline]

27.158.4 Member Function Documentation

27.158.4.1 void gdcm::IODs::AddIOD (const char * *name*, const IOD & *module*) [inline]

27.158.4.2 IODMapTypeConstIterator gdcm::IODs::Begin ( ) const [inline]

Examples:

TraverseModules.cxx.

27.158.4.3 void gdcm::IODs::Clear ( ) [inline]

27.158.4.4 IODMapTypeConstIterator gdcm::IODs::End ( ) const [inline]

Examples:

TraverseModules.cxx.

27.158.4.5 const IOD& gdcm::IODs::GetIOD (const char * *name*) const [inline]

27.158.5 Friends And Related Function Documentation

27.158.5.1 std::ostream& operator<< (std::ostream & *_os,* const IODs & *_val*) [friend]

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

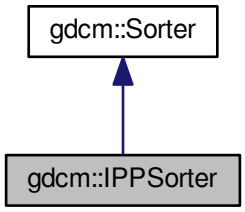
- gdcmIODs.h

27.159 gdcm::IPPSorter Class Reference

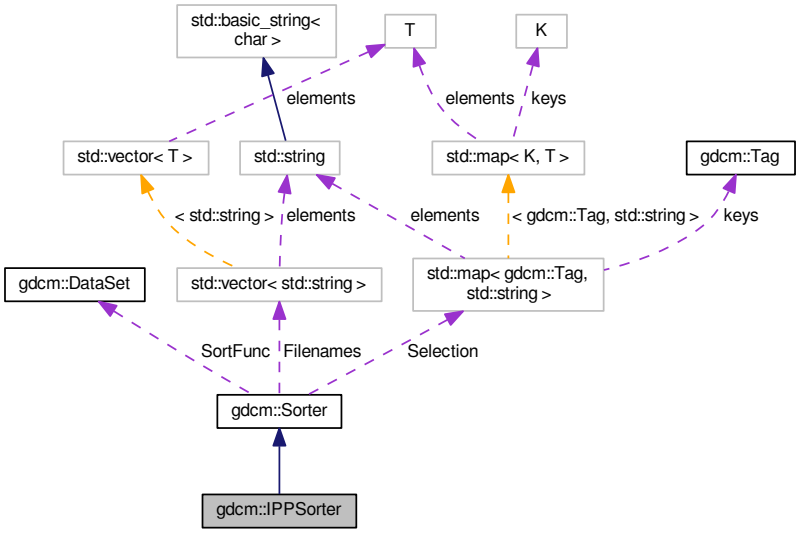
IPPSorter Implement a simple Image Position (Patient) sorter, along the Image Orientation (Patient) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

`#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)
- virtual bool Sort (std::vector< std::string > const &filenames)

Protected Attributes

- bool ComputeZSpacing
- double DirCosTolerance
- bool DropDuplicatePositions
- double ZSpacing
- double ZTolerance

Additional Inherited Members

27.159.1 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

Examples:

Compute3DSpacing.cxx, gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

27.159.2 Constructor & Destructor Documentation

27.159.2.1 gdcm::IPPSorter::IPPSorter ( )

27.159.3 Member Function Documentation

27.159.3.1 double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const [inline]

27.159.3.2 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.

27.159.3.3 double gdcm::IPPSorter::GetZSpacingTolerance ( ) const [inline]

27.159.3.4 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, gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

27.159.3.5 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 the distance in between 1. 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).

27.159.3.6 void gdcm::IPPSorter::SetDropDuplicatePositions (bool *b*) [inline]

Makes the IPPSorter ignore multiple images located at the same position. Only the first occurence will be kept. Drop↩
DuplicatePositions defaults to false.

27.159.3.7 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 serie, 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.

27.159.3.8 virtual bool gdcm::IPPSorter::Sort (std::vector< std::string > const & *filenames*) [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 achived. 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, gdcmorthoplanes.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

27.159.4 Member Data Documentation

27.159.4.1 bool gdcm::IPPSorter::ComputeZSpacing [protected]

27.159.4.2 double gdcm::IPPSorter::DirCosTolerance [protected]

27.159.4.3 bool gdcm::IPPSorter::DropDuplicatePositions [protected]

27.159.4.4 double gdcm::IPPSorter::ZSpacing [protected]

27.159.4.5 double gdcm::IPPSorter::ZTolerance [protected]

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

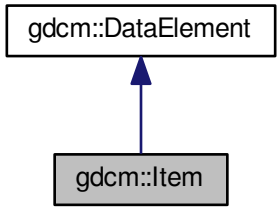
- gdcmIPPSorter.h

27.160 gdcm::Item Class Reference

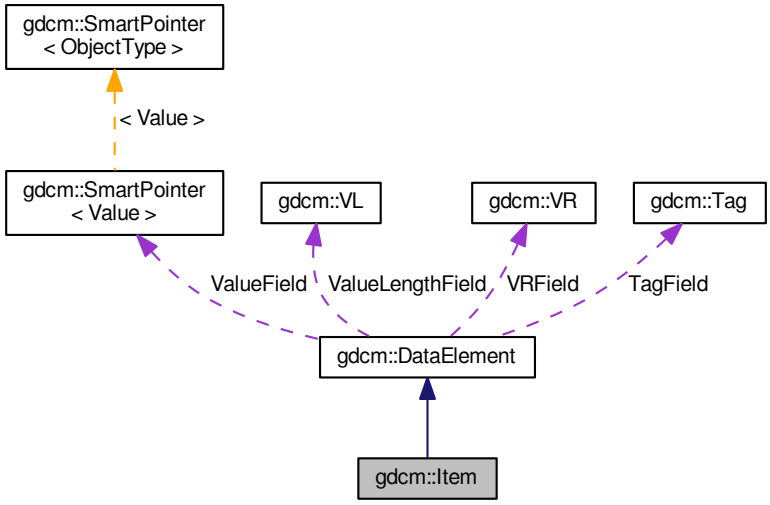
Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set . See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standart Data Element with a specific Data Element 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.

```
#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
- const DataSet & GetNestedDataSet () const
- DataSet & GetNestedDataSet ()
- 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

Friends

- std::ostream & operator<< (std::ostream &os, const Item &val)

Additional Inherited Members

27.160.1 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 Standart 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, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, and LargeVRDSExplicit.cxx.

27.160.2 Constructor & Destructor Documentation

27.160.2.1 gdcm::Item::Item ( ) [inline]

27.160.2.2 gdcm::Item::Item (Item const & *val*) [inline]

27.160.3 Member Function Documentation

27.160.3.1 void gdcm::Item::Clear ( ) [inline]

References gdcm::DataElement::Clear().

Referenced by gdcm::SequenceOfItems::Read().

27.160.3.2 bool gdcm::Item::FindDataElement (const Tag & *t*) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

27.160.3.3 const DataElement& gdcm::Item::GetDataElement (const Tag & *t*) const [inline]

Examples:

ReadAndDumpDICOMDIR.cxx.

27.160.3.4 template<typename TDE > VL gdcm::Item::GetLength ( ) const

27.160.3.5 const DataSet& gdcm::Item::GetNestedDataSet ( ) const [inline]

Examples:

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.↩cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.↩

cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, and LargeVRDSExplicit.cxx.

Referenced by gdcm::SequenceOfItems::Read().

27.160.3.6 DataSet& gdcm::Item::GetNestedDataSet ( ) [inline]

27.160.3.7 void gdcm::Item::InsertDataElement (const DataElement & *de*) [inline]

27.160.3.8 template<typename TDE , typename TSwap > std::istream& gdcm::Item::Read (std::istream & *is*) [inline]

References gdcm::DataSet::Clear(), gdcmDebugMacro, gdcmErrorMacro, gdcmWarningMacro, gdcm::DataSet::IsEmpty(), and gdcm::SwapperDoOp::Swap().

Referenced by gdcm::SequenceOfItems::Read().

27.160.3.9 void gdcm::Item::SetNestedDataSet (const DataSet & *nested*) [inline]

27.160.3.10 template<typename TDE , typename TSwap > const std::ostream& gdcm::Item::Write (std::ostream & *os*) const [inline]

References gdcmWarningMacro, gdcm::VL::GetLength(), gdcm::VL::Write(), and gdcm::Tag::Write().

27.160.4 Friends And Related Function Documentation

27.160.4.1 std::ostream& operator<< (std::ostream & *os,* const Item & *val*) [friend]

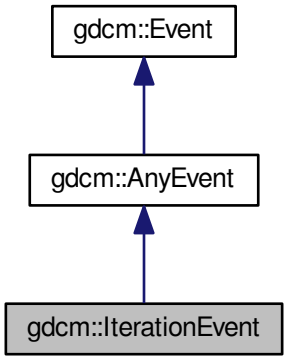
The documentation for this class was generated from the following file:

- gdcmItem.h

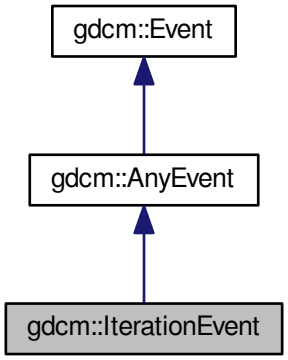
27.161 gdcm::IterationEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



Additional Inherited Members

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

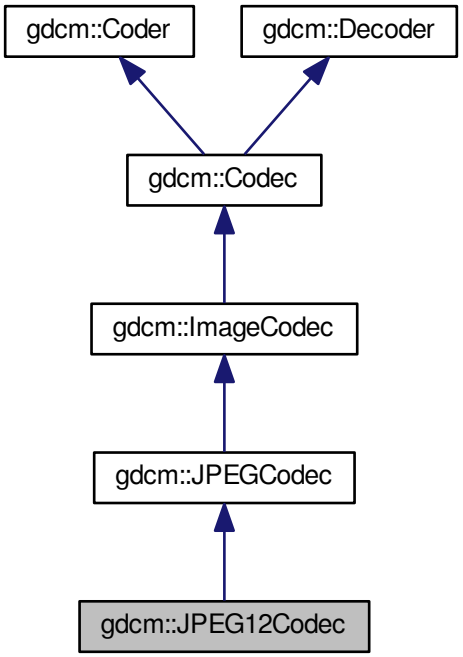
- gdcmEvent.h

27.162 gdcm::JPEG12Codec Class Reference

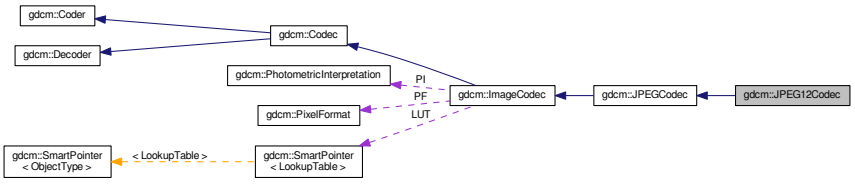
Class to do JPEG 12bits (lossy & lossless)

`#include <gdcmJPEG12Codec.h>`

Inheritance diagram for gdcm::JPEG12Codec:



Collaboration diagram for gdcm::JPEG12Codec:



Public Member Functions

- JPEG12Codec ()
- ~JPEG12Codec ()

- bool DecodeByStreams (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char *input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen)
- bool IsStateSuspension () const

Additional Inherited Members

27.162.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

Note

internal class

27.162.2 Constructor & Destructor Documentation

27.162.2.1 gdcm::JPEG12Codec::JPEG12Codec ( )

27.162.2.2 gdcm::JPEG12Codec::~JPEG12Codec ( )

27.162.3 Member Function Documentation

27.162.3.1 bool gdcm::JPEG12Codec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::ImageCodec.

27.162.3.2 virtual bool gdcm::JPEG12Codec::EncodeBuffer (std::ostream & *os*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

27.162.3.3 bool gdcm::JPEG12Codec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

27.162.3.4 bool gdcm::JPEG12Codec::InternalCode (const char * *input*, unsigned long *len*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

27.162.3.5 bool gdcm::JPEG12Codec::IsStateSuspension ( ) const [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

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

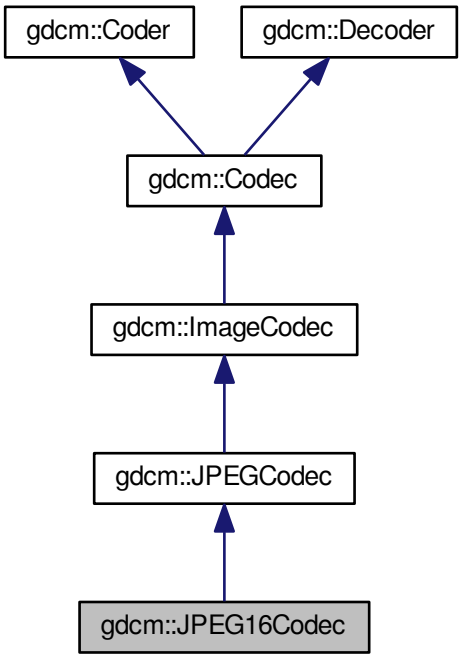
- gdcmJPEG12Codec.h

27.163 gdcm::JPEG16Codec Class Reference

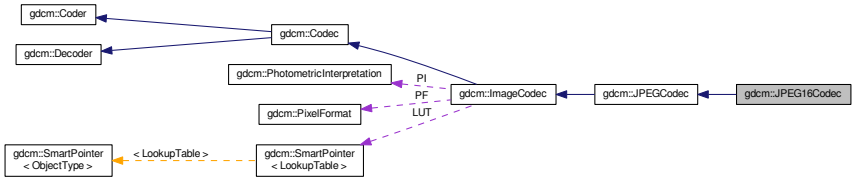
Class to do JPEG 16bits (lossless)

`#include <gdcmJPEG16Codec.h>`

Inheritance diagram for gdcm::JPEG16Codec:



Collaboration diagram for gdcm::JPEG16Codec:



Public Member Functions

- JPEG16Codec ()
- ~JPEG16Codec ()
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char *input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen)
- bool IsStateSuspension () const

Additional Inherited Members

27.163.1 Detailed Description

Class to do JPEG 16bits (lossless)

Note

internal class

27.163.2 Constructor & Destructor Documentation

27.163.2.1 gdcm::JPEG16Codec::JPEG16Codec ( )

27.163.2.2 gdcm::JPEG16Codec::~JPEG16Codec ( )

27.163.3 Member Function Documentation

27.163.3.1 bool gdcm::JPEG16Codec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::ImageCodec.

27.163.3.2 virtual bool gdcm::JPEG16Codec::EncodeBuffer (std::ostream & *os*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

27.163.3.3 bool gdcm::JPEG16Codec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

27.163.3.4 bool gdcm::JPEG16Codec::InternalCode (const char * *input*, unsigned long *len*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

27.163.3.5 bool gdcm::JPEG16Codec::IsStateSuspension ( ) const [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

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

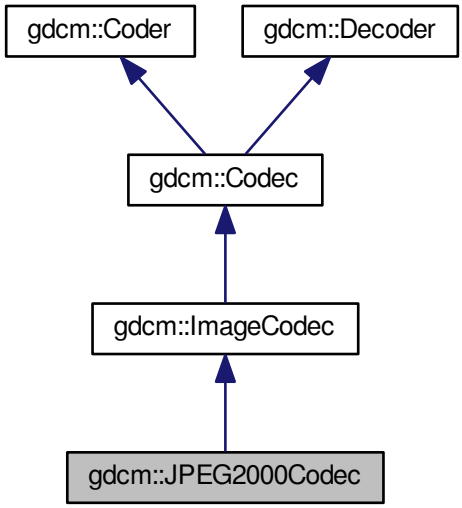
- gdcmJPEG16Codec.h

27.164 gdcm::JPEG2000Codec Class Reference

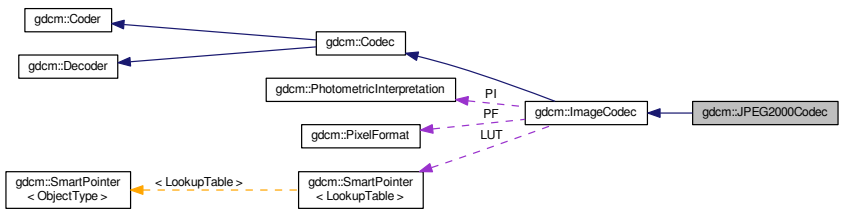
Class to do JPEG 2000.

`#include <gdcmJPEG2000Codec.h>`

Inheritance diagram for gdcm::JPEG2000Codec:



Collaboration diagram for gdcm::JPEG2000Codec:



Public Member Functions

- JPEG2000Codec ()
- ~JPEG2000Codec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- virtual bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- double GetQuality (unsigned int idx=0) const
- double GetRate (unsigned int idx=0) const
- void SetNumberOfResolutions (unsigned int nres)
- 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)

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- 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 ()
- bool IsRowEncoder ()
- bool StartEncode (std::ostream &)
- bool StopEncode (std::ostream &)

Friends

- class Bitmap
- class ImageRegionReader

Additional Inherited Members

27.164.1 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

27.164.2 Constructor & Destructor Documentation

27.164.2.1 gdcm::JPEG2000Codec::JPEG2000Codec ( )

27.164.2.2 gdcm::JPEG2000Codec::~JPEG2000Codec ( )

27.164.3 Member Function Documentation

27.164.3.1 bool gdcm::JPEG2000Codec::AppendFrameEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.164.3.2 bool gdcm::JPEG2000Codec::AppendRowEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.164.3.3 bool gdcm::JPEG2000Codec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.164.3.4 bool gdcm::JPEG2000Codec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.164.3.5 virtual ImageCodec* gdcm::JPEG2000Codec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.164.3.6 bool gdcm::JPEG2000Codec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

27.164.3.7 bool gdcm::JPEG2000Codec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

27.164.3.8 bool gdcm::JPEG2000Codec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.164.3.9 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]

27.164.3.10 virtual bool gdcm::JPEG2000Codec::GetHeaderInfo (std::istream & *is,* **TransferSyntax** & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

27.164.3.11 double gdcm::JPEG2000Codec::GetQuality (unsigned int *idx* = 0) const

27.164.3.12 double gdcm::JPEG2000Codec::GetRate (unsigned int *idx* = 0) const

27.164.3.13 bool gdcm::JPEG2000Codec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.164.3.14 bool gdcm::JPEG2000Codec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.164.3.15 void gdcm::JPEG2000Codec::SetNumberOfResolutions (unsigned int *nres*)

27.164.3.16 void gdcm::JPEG2000Codec::SetQuality (unsigned int *idx,* double *q*)

27.164.3.17 void gdcm::JPEG2000Codec::SetRate (unsigned int *idx,* double *rate*)

27.164.3.18 void gdcm::JPEG2000Codec::SetReversible (bool *res*)

27.164.3.19 void gdcm::JPEG2000Codec::SetTileSize (unsigned int *tx,* unsigned int *ty*)

27.164.3.20 bool gdcm::JPEG2000Codec::StartEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.164.3.21 bool gdcm::JPEG2000Codec::StopEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.164.4 Friends And Related Function Documentation

27.164.4.1 friend class **Bitmap** [friend]

27.164.4.2 friend class **ImageRegionReader** [friend]

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

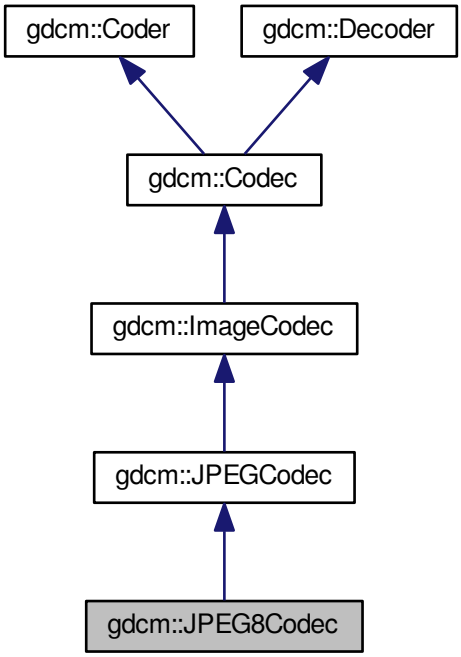
- gdcmJPEG2000Codec.h

27.165 gdcm::JPEG8Codec Class Reference

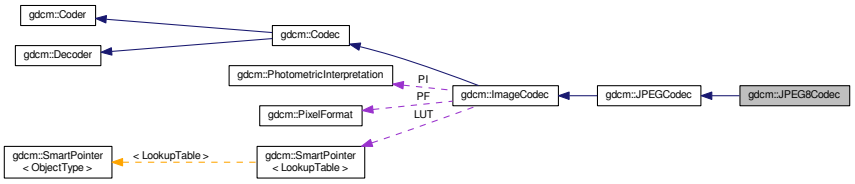
Class to do JPEG 8bits (lossy & lossless)

`#include <gdcmJPEG8Codec.h>`

Inheritance diagram for gdcm::JPEG8Codec:



Collaboration diagram for gdcm::JPEG8Codec:



Public Member Functions

- JPEG8Codec ()
- ~JPEG8Codec ()

- bool DecodeByStreams (std::istream &is, std::ostream &os)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool InternalCode (const char *input, unsigned long len, std::ostream &os)

Protected Member Functions

- virtual bool EncodeBuffer (std::ostream &os, const char *data, size_t datalen)
- bool IsStateSuspension () const

Additional Inherited Members

27.165.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

Note

internal class

27.165.2 Constructor & Destructor Documentation

27.165.2.1 gdcm::JPEG8Codec::JPEG8Codec ( )

27.165.2.2 gdcm::JPEG8Codec::~JPEG8Codec ( )

27.165.3 Member Function Documentation

27.165.3.1 bool gdcm::JPEG8Codec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::ImageCodec.

27.165.3.2 virtual bool gdcm::JPEG8Codec::EncodeBuffer (std::ostream & *os*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

27.165.3.3 bool gdcm::JPEG8Codec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::JPEGCodec.

27.165.3.4 bool gdcm::JPEG8Codec::InternalCode (const char * *input*, unsigned long *len*, std::ostream & *os*) [virtual]

Reimplemented from gdcm::Coder.

27.165.3.5 bool gdcm::JPEG8Codec::IsStateSuspension ( ) const [protected], [virtual]

Reimplemented from gdcm::JPEGCodec.

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

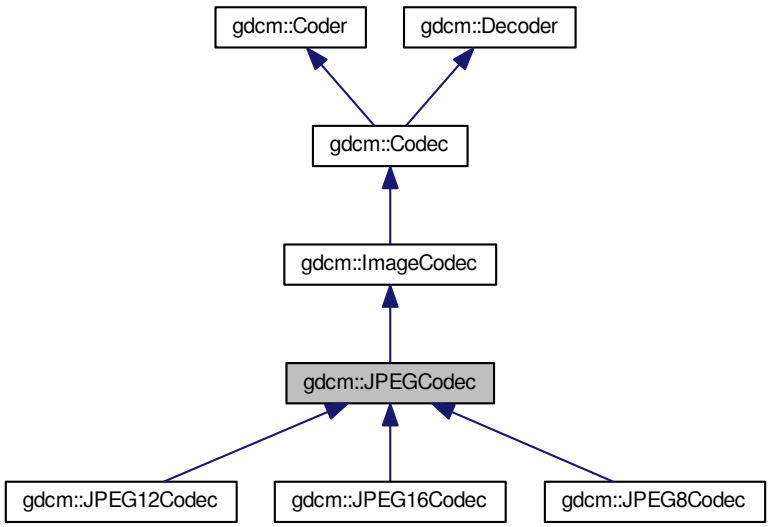
- gdcmJPEG8Codec.h

27.166 gdcm::JPEGCodec Class Reference

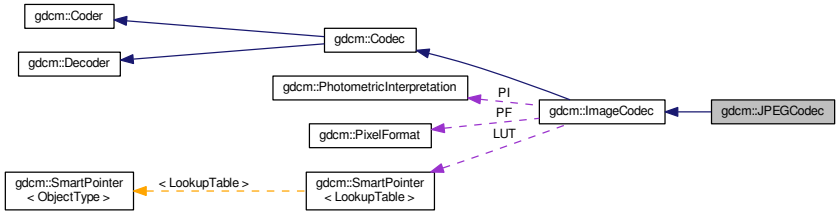
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: JPEG8Codec, JPEG12Codec & JPEG16Codec It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case.

`#include <gdcmJPEGCodec.h>`

Inheritance diagram for gdcm::JPEGCodec:



Collaboration diagram for gdcm::JPEGCodec:



Public Member Functions

- JPEGCodec ()

- ~JPEGCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Compress into JPEG.
- void ComputeOffsetTable (bool b)

  Compute the offset table:
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- virtual bool EncodeBuffer (std::ostream &out, const char *inbuffer, size_t inlen)
- virtual bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool GetLossless () const
- double GetQuality () const
- void SetLossless (bool l)
- void SetPixelFormat (PixelFormat const &pf)
- void SetQuality (double q)

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- 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 ()
- bool IsRowEncoder ()
- virtual bool IsStateSuspension () const
- bool IsValid (PhotometricInterpretation const &pi)
- void SetBitSample (int bit)
- bool StartEncode (std::ostream &)
- bool StopEncode (std::ostream &)

Protected Attributes

- int BitSample
- int Quality

Friends

- class ImageRegionReader

Additional Inherited Members

27.166.1 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/6b93af410f8c92

Examples:

GetJPEGSamplePrecision.cxx.

27.166.2 Constructor & Destructor Documentation

27.166.2.1 gdcm::JPEGCodec::JPEGCodec ( )

27.166.2.2 gdcm::JPEGCodec::~JPEGCodec ( )

27.166.3 Member Function Documentation

27.166.3.1 bool gdcm::JPEGCodec::AppendFrameEncode (std::ostream & *out,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.166.3.2 bool gdcm::JPEGCodec::AppendRowEncode (std::ostream & *out,* const char * *data,* size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.166.3.3 bool gdcm::JPEGCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.166.3.4 bool gdcm::JPEGCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.166.3.5 virtual ImageCodec* gdcm::JPEGCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.166.3.6 bool gdcm::JPEGCodec::Code (DataElement const & *in*, DataElement & *out*) [virtual]

Compress into JPEG.

Reimplemented from gdcm::Coder.

27.166.3.7 void gdcm::JPEGCodec::ComputeOffsetTable (bool *b*)

Compute the offset table:

27.166.3.8 bool gdcm::JPEGCodec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

27.166.3.9 bool gdcm::JPEGCodec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.166.3.10 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]

27.166.3.11 virtual bool gdcm::JPEGCodec::EncodeBuffer (std::ostream & *out*, const char * *inbuffer*, size_t *inlen*) [virtual]

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

27.166.3.12 virtual bool gdcm::JPEGCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

Examples:

GetJPEGSamplePrecision.cxx.

27.166.3.13 bool gdcm::JPEGCodec::GetLossless ( ) const

27.166.3.14 double gdcm::JPEGCodec::GetQuality ( ) const

27.166.3.15 bool gdcm::JPEGCodec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.166.3.16 bool gdcm::JPEGCodec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.166.3.17 virtual bool gdcm::JPEGCodec::IsStateSuspension ( ) const [protected], [virtual]

Reimplemented in gdcm::JPEG12Codec, gdcm::JPEG16Codec, and gdcm::JPEG8Codec.

27.166.3.18 bool gdcm::JPEGCodec::IsValid (PhotometricInterpretation const & *pi*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.166.3.19 void gdcm::JPEGCodec::SetBitSample (int *bit*) [protected]

27.166.3.20 void gdcm::JPEGCodec::SetLossless (bool *l*)

27.166.3.21 void gdcm::JPEGCodec::SetPixelFormat (PixelFormat const & *pf*) [virtual]

Reimplemented from gdcm::ImageCodec.

Examples:

GetJPEGSamplePrecision.cxx.

27.166.3.22 void gdcm::JPEGCodec::SetQuality (double *q*)

27.166.3.23 bool gdcm::JPEGCodec::StartEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.166.3.24 bool gdcm::JPEGCodec::StopEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.166.4 Friends And Related Function Documentation

27.166.4.1 friend class ImageRegionReader [friend]

27.166.5 Member Data Documentation

27.166.5.1 int gdcm::JPEGCodec::BitSample [protected]

27.166.5.2 int gdcm::JPEGCodec::Quality [protected]

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

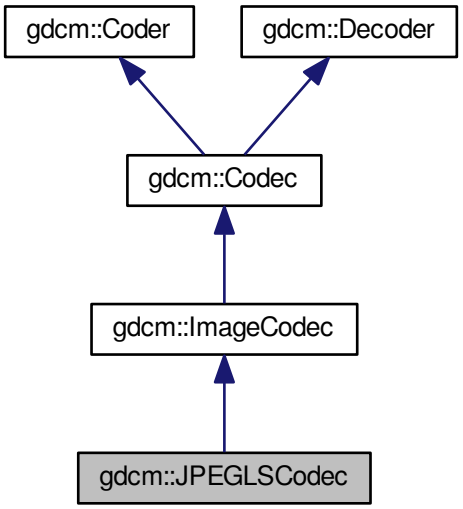
- gdcmJPEGCodec.h

27.167 gdcm::JPEGLSCodec Class Reference

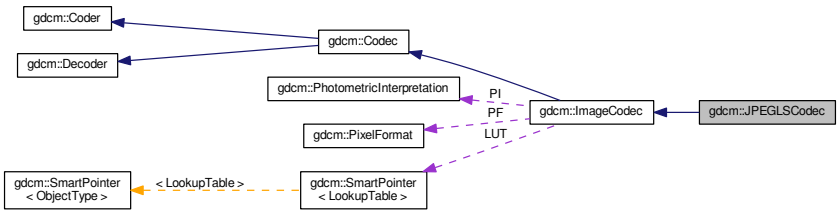
JPEG-LS.

`#include <gdcmJPEGLSCodec.h>`

Inheritance diagram for gdcm::JPEGLSCodec:



Collaboration diagram for gdcm::JPEGLSCodec:



Public Member Functions

- JPEGLSCodec ()
- ~JPEGLSCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

Return whether this decoder support this transfer syntax (can decode it)

- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- 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)
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool GetLossless () const
- void SetBufferLength (unsigned long l)
- void SetLossless (bool l)
- void SetLossyError (int error)

  [0-3] generally

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- 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 ()
- bool IsRowEncoder ()
- bool StartEncode (std::ostream &)
- bool StopEncode (std::ostream &)

Friends

- class ImageRegionReader

Additional Inherited Members

27.167.1 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 `http://charls.codeplex.com`

27.167.2 Constructor & Destructor Documentation

27.167.2.1 gdcm::JPEGLSCodec::JPEGLSCodec ( )

27.167.2.2 gdcm::JPEGLSCodec::~JPEGLSCodec ( )

27.167.3 Member Function Documentation

27.167.3.1 bool gdcm::JPEGLSCodec::AppendFrameEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.167.3.2 bool gdcm::JPEGLSCodec::AppendRowEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.167.3.3 bool gdcm::JPEGLSCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.167.3.4 bool gdcm::JPEGLSCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.167.3.5 virtual ImageCodec* gdcm::JPEGLSCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.167.3.6 bool gdcm::JPEGLSCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

27.167.3.7 bool gdcm::JPEGLSCodec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

27.167.3.8 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*)

27.167.3.9 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]

27.167.3.10 unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const [inline]

27.167.3.11 bool gdcm::JPEGLSCodec::GetHeaderInfo (std::istream & *is,* TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

27.167.3.12 bool gdcm::JPEGLSCodec::GetLossless ( ) const

27.167.3.13 bool gdcm::JPEGLSCodec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.167.3.14 bool gdcm::JPEGLSCodec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.167.3.15 void gdcm::JPEGLSCodec::SetBufferLength (unsigned long *l*) [inline]

27.167.3.16 void gdcm::JPEGLSCodec::SetLossless (bool *l*)

27.167.3.17 void gdcm::JPEGLSCodec::SetLossyError (int *error*)

[0-3] generally

27.167.3.18 bool gdcm::JPEGLSCodec::StartEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.167.3.19 bool gdcm::JPEGLSCodec::StopEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.167.4 Friends And Related Function Documentation

27.167.4.1 friend class ImageRegionReader [friend]

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

- gdcmJPEGLSCodec.h

27.168 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)

27.168.1 Detailed Description

Examples:

QIDO-RS.cxx.

27.168.2 Constructor & Destructor Documentation

27.168.2.1 gdcm::JSON::JSON ( )

27.168.2.2 gdcm::JSON::~JSON ( )

27.168.3 Member Function Documentation

27.168.3.1 bool gdcm::JSON::Code (DataSet const & *in*, std::ostream & *os*)

Examples:

QIDO-RS.cxx.

27.168.3.2 bool gdcm::JSON::Decode (std::istream & *is*, DataSet & *out*)

Examples:

QIDO-RS.cxx.

27.168.3.3 bool gdcm::JSON::GetPrettyPrint ( ) const

27.168.3.4 void gdcm::JSON::PrettyPrintOff ( )

27.168.3.5 void gdcm::JSON::PrettyPrintOn ( )

Examples:

QIDO-RS.cxx.

27.168.3.6 void gdcm::JSON::SetPrettyPrint (bool *onoff*)

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

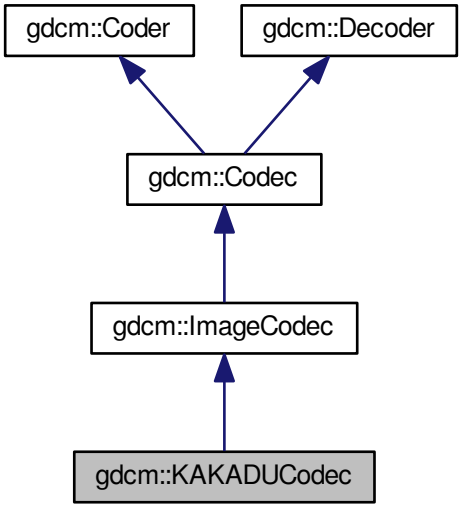
- gdcmJSON.h

27.169 gdcm::KAKADUCodec Class Reference

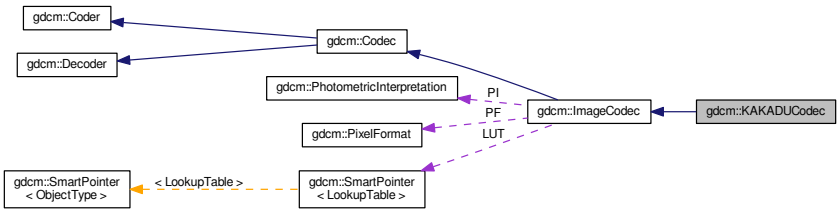
KAKADUCodec.

`#include <gdcmKAKADUCodec.h>`

Inheritance diagram for gdcm::KAKADUCodec:



Collaboration diagram for gdcm::KAKADUCodec:



Public Member Functions

- KAKADUCodec ()
- ~KAKADUCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Additional Inherited Members

27.169.1 Detailed Description

KAKADUCodec.

27.169.2 Constructor & Destructor Documentation

27.169.2.1 gdcm::KAKADUCodec::KAKADUCodec ( )

27.169.2.2 gdcm::KAKADUCodec::~KAKADUCodec ( )

27.169.3 Member Function Documentation

27.169.3.1 bool gdcm::KAKADUCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.169.3.2 bool gdcm::KAKADUCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.169.3.3 virtual ImageCodec* gdcm::KAKADUCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.169.3.4 bool gdcm::KAKADUCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

27.169.3.5 bool gdcm::KAKADUCodec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

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

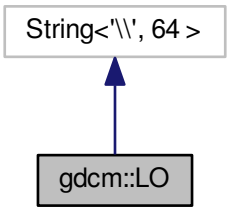
- gdcmKAKADUCodec.h

27.170 gdcm::LO Class Reference

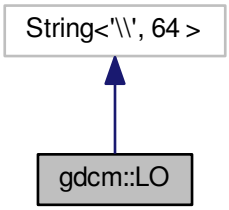
LO.

`#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 value_type *s)
- LO (const value_type *s, size_type n)
- LO (const Superclass &s, size_type pos=0, size_type n=npos)
- bool IsValid () const

27.170.1 Detailed Description

LO.

Note

TODO

27.170.2 Member Typedef Documentation

27.170.2.1 typedef Superclass::const_iterator gdcm::LO::const_iterator

27.170.2.2 typedef Superclass::const_reference gdcm::LO::const_reference

27.170.2.3 typedef Superclass::const_reverse_iterator gdcm::LO::const_reverse_iterator

27.170.2.4 typedef Superclass::difference_type gdcm::LO::difference_type

27.170.2.5 typedef Superclass::iterator gdcm::LO::iterator

27.170.2.6 typedef Superclass::pointer gdcm::LO::pointer

27.170.2.7 typedef Superclass::reference gdcm::LO::reference

27.170.2.8 typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator

27.170.2.9 typedef Superclass::size_type gdcm::LO::size_type

27.170.2.10 typedef String<'\\',64> gdcm::LO::Superclass

27.170.2.11 typedef Superclass::value_type gdcm::LO::value_type

27.170.3 Constructor & Destructor Documentation

27.170.3.1 gdcm::LO::LO ( ) [inline]

27.170.3.2 gdcm::LO::LO (const value_type * *s*) [inline]

27.170.3.3 gdcm::LO::LO (const value_type * *s*, size_type *n*) [inline]

27.170.3.4 gdcm::LO::LO (const Superclass & *s*, size_type *pos* = 0, size_type *n* = npos) [inline]

27.170.4 Member Function Documentation

27.170.4.1 bool gdcm::LO::IsValid ( ) const [inline]

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

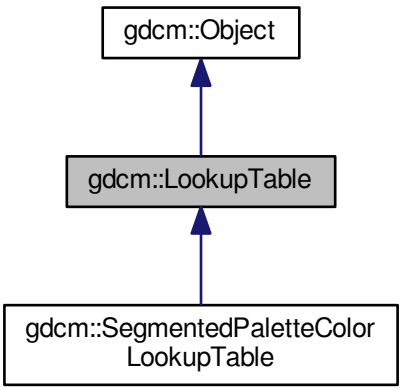
- gdcmLO.h

27.171 gdcm::LookupTable Class Reference

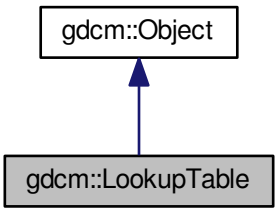
LookupTable class.

`#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 ()
- void Allocate (unsigned short bitsample=8)

  Allocate the LUT.
- void Clear ()

  Clear the LUT.
- void Decode (std::istream &is, std::ostream &os) const

  Decode the LUT.
- bool Decode (char *outputbuffer, size_t outlen, const char *inputbuffer, size_t inlen) const
- 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:
- void Print (std::ostream &) const
- 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.

Protected Attributes

- unsigned short BitSample
- bool IncompleteLUT:1
- LookupTableInternal * Internal

Additional Inherited Members

27.171.1 Detailed Description

LookupTable class.

27.171.2 Member Enumeration Documentation

27.171.2.1 enum gdcm::LookupTable::LookupTableType

Enumerator

RED

GREEN

BLUE

GRAY

UNKNOWN

27.171.3 Constructor & Destructor Documentation

27.171.3.1 gdcm::LookupTable::LookupTable ( )

27.171.3.2 gdcm::LookupTable::~LookupTable ( )

27.171.3.3 gdcm::LookupTable::LookupTable (LookupTable const & *lut*) [inline]

27.171.4 Member Function Documentation

27.171.4.1 void gdcm::LookupTable::Allocate (unsigned short *bitsample* = 8)

Allocate the LUT.

27.171.4.2 void gdcm::LookupTable::Clear ( )

Clear the LUT.

27.171.4.3 void gdcm::LookupTable::Decode (std::istream & *is,* std::ostream & *os*) const

Decode the LUT.

27.171.4.4 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

27.171.4.5 unsigned short gdcm::LookupTable::GetBitSample ( ) const [inline]

return the bit sample

27.171.4.6 bool gdcm::LookupTable::GetBufferAsRGBA (unsigned char * *rgba*) const

return the LUT as RGBA buffer

27.171.4.7 void gdcm::LookupTable::GetLUT (LookupTableType *type,* unsigned char * *array,* unsigned int & *length*) const

Examples:

ExtractImageRegionWithLUT.cs.

27.171.4.8 void gdcm::LookupTable::GetLUTDescriptor (LookupTableType *type,* unsigned short & *length,* unsigned short & *subscript,* unsigned short & *bitsize*) const

27.171.4.9 unsigned int gdcm::LookupTable::GetLUTLength (LookupTableType *type*) const

27.171.4.10 const unsigned char* gdcm::LookupTable::GetPointer ( ) const

return a raw pointer to the LUT

27.171.4.11 void gdcm::LookupTable::InitializeBlueLUT (unsigned short *length,* unsigned short *subscript,* unsigned short *bitsize*)

27.171.4.12 bool gdcm::LookupTable::Initialized ( ) const

return whether the LUT has been initialized

27.171.4.13 void gdcm::LookupTable::InitializeGreenLUT (unsigned short *length,* unsigned short *subscript,* unsigned short *bitsize*)

27.171.4.14 void gdcm::LookupTable::InitializeLUT (LookupTableType *type,* unsigned short *length,* unsigned short *subscript,* unsigned short *bitsize*)

Generic interface:

27.171.4.15 void gdcm::LookupTable::InitializeRedLUT (unsigned short *length,* unsigned short *subscript,* unsigned short *bitsize*)

RED / GREEN / BLUE specific:

27.171.4.16 void gdcm::LookupTable::Print (std::ostream &) const [inline], [virtual]

Reimplemented from gdcm::Object.

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

27.171.4.17 void gdcm::LookupTable::SetBlueLUT (const unsigned char * *blue,* unsigned int *length*)

27.171.4.18 void gdcm::LookupTable::SetGreenLUT (const unsigned char * *green,* unsigned int *length*)

27.171.4.19 virtual void gdcm::LookupTable::SetLUT (LookupTableType *type,* const unsigned char * *array,* unsigned int *length*) [virtual]

Reimplemented in gdcm::SegmentedPaletteColorLookupTable.

27.171.4.20 void gdcm::LookupTable::SetRedLUT (const unsigned char * *red,* unsigned int *length*)

27.171.4.21 bool gdcm::LookupTable::WriteBufferAsRGBA (const unsigned char * *rgba*)

Write the LUT as RGBA.

27.171.5 Member Data Documentation

27.171.5.1 unsigned short gdcm::LookupTable::BitSample [protected]

27.171.5.2 bool gdcm::LookupTable::IncompleteLUT [protected]

27.171.5.3 LookupTableInternal* gdcm::LookupTable::Internal [protected]

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

- gdcmLookupTable.h

27.172 gdcm::Scanner::Itstr Struct Reference

`#include <gdcmScanner.h>`

Public Member Functions

- bool operator() (const char *s1, const char *s2) const

27.172.1 Member Function Documentation

27.172.1.1 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

27.173 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 ()
- void AddMacroEntry (const Tag &tag, const MacroEntry &module)

  Will add a ModuleEntry direcly 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)

27.173.1 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

27.173.2 Member Typedef Documentation

27.173.2.1 typedef std::vector<std::string> gdcm::Macro::ArrayIncludeMacrosType

27.173.2.2 typedef std::map<Tag, MacroEntry> gdcm::Macro::MapModuleEntry

27.173.3 Constructor & Destructor Documentation

27.173.3.1 gdcm::Macro::Macro ( ) [inline]

27.173.4 Member Function Documentation

27.173.4.1 void gdcm::Macro::AddMacroEntry (const Tag & *tag,* const MacroEntry & *module*) [inline]

Will add a ModuleEntry direcly at root-level. See Macro for nested-included level.

27.173.4.2 void gdcm::Macro::Clear ( ) [inline]

27.173.4.3 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.

27.173.4.4 const MacroEntry& gdcm::Macro::GetMacroEntry (const Tag & *tag*) const

27.173.4.5 const char* gdcm::Macro::GetName ( ) const [inline]

27.173.4.6 void gdcm::Macro::SetName (const char * *name*) [inline]

27.173.4.7 bool gdcm::Macro::Verify (const DataSet & *ds,* Usage const & *usage*) const

27.173.5 Friends And Related Function Documentation

27.173.5.1 std::ostream& operator<< (std::ostream & *_os,* const Macro & *_val*) [friend]

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

- gdcmMacro.h

27.174 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 ()
- 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)

27.174.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples:

TraverseModules.cxx.

27.174.2 Member Typedef Documentation

27.174.2.1 typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType

27.174.3 Constructor & Destructor Documentation

27.174.3.1 gdcm::Macros::Macros ( ) [inline]

27.174.4 Member Function Documentation

27.174.4.1 void gdcm::Macros::AddMacro (const char * *ref,* const Macro & *module*) [inline]

27.174.4.2 void gdcm::Macros::Clear ( ) [inline]

27.174.4.3 const Macro& gdcm::Macros::GetMacro (const char * *name*) const [inline]

27.174.4.4 bool gdcm::Macros::IsEmpty ( ) const [inline]

27.174.5 Friends And Related Function Documentation

27.174.5.1 std::ostream& operator<< (std::ostream & *_os,* const Macros & *_val*) [friend]

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

- gdcmMacros.h

27.175 gdcm::network::MaximumLengthSub Class Reference

MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

`#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

27.175.1 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)

27.175.2 Constructor & Destructor Documentation

27.175.2.1 gdcm::network::MaximumLengthSub::MaximumLengthSub ( )

27.175.3 Member Function Documentation

27.175.3.1 uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength ( ) const [inline]

27.175.3.2 void gdcm::network::MaximumLengthSub::Print (std::ostream & *os*) const

27.175.3.3 std::istream& gdcm::network::MaximumLengthSub::Read (std::istream & *is*)

27.175.3.4 void gdcm::network::MaximumLengthSub::SetMaximumLength (uint32_t *maximumlength*)

27.175.3.5 size_t gdcm::network::MaximumLengthSub::Size ( ) const

27.175.3.6 const std::ostream& gdcm::network::MaximumLengthSub::Write (std::ostream & *os*) const

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

- gdcmMaximumLengthSub.h

27.176 gdcm::MD5 Class Reference

Class for MD5.

`#include <gdcmMD5.h>`

Public Member Functions

- MD5 ()
- ~MD5 ()

Static Public Member Functions

- static bool Compute (const char *buffer, unsigned long buf_len, char digest_str[33])
- static bool ComputeFile (const char *filename, char digest_str[33])

27.176.1 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

27.176.2 Constructor & Destructor Documentation

27.176.2.1 gdcm::MD5::MD5 ( )

27.176.2.2 gdcm::MD5::~MD5 ( )

27.176.3 Member Function Documentation

27.176.3.1 static bool gdcm::MD5::Compute (const char * *buffer*, unsigned long *buf_len*, char *digest_str[33]*) [static]

27.176.3.2 static bool gdcm::MD5::ComputeFile (const char * *filename*, char *digest_str[33]*) [static]

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

- gdcmMD5.h

27.177 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,

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)

27.177.1 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.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenerateStandardSOPClasses.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, ReadAndDumpDICOMDIR.cxx, Stream↩ ImageReaderTest.cxx, and TestReader.cxx.

27.177.2 Member Enumeration Documentation

27.177.2.1 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

MS_END

Examples:

GenerateStandardSOPClasses.cxx.

27.177.2.2 enum gdcm::MediaStorage::ObjectType

Enumerator

NoObject

Video

Waveform

Audio

PDF

URI

Segmentation

ObjectEnd

27.177.3 Constructor & Destructor Documentation

27.177.3.1 gdcm::MediaStorage::MediaStorage (MSType *type* = MS_END) [inline]

27.177.4 Member Function Documentation

27.177.4.1 const char* gdcm::MediaStorage::GetModality ( ) const

27.177.4.2 unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const

27.177.4.3 static const char* gdcm::MediaStorage::GetMSString (MSType *ts*) [static]

Return the Media String associated. Will return NULL for MS_END.

Examples:

GenerateStandardSOPClasses.cxx.

Referenced by gdcm::operator<<().

27.177.4.4 static MSType gdcm::MediaStorage::GetMSType (const char * *str*) [static]

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

27.177.4.5 static unsigned int gdcm::MediaStorage::GetNumberOfModality ( ) [static]

27.177.4.6 static unsigned int gdcm::MediaStorage::GetNumberOfMSString ( ) [static]

27.177.4.7 static unsigned int gdcm::MediaStorage::GetNumberOfMSType ( ) [static]

27.177.4.8 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, iU22tomultisc.cxx, and StreamImageReaderTest.cxx.

27.177.4.9 void gdcm::MediaStorage::GuessFromModality (const char * *modality,* unsigned int *dimension* = 2)

27.177.4.10 static 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.

27.177.4.11 bool gdcm::MediaStorage::IsUndefined ( ) const [inline]

Examples:

TestReader.cxx.

27.177.4.12 gdcm::MediaStorage::operator MSType ( ) const [inline]

27.177.4.13 bool gdcm::MediaStorage::SetFromDataSet (DataSet const & *ds*)

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

27.177.4.14 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:

gdcmrtionplan.cxx, gdcmrtplan.cxx, ReadAndDumpDICOMDIR.cxx, and TestReader.cxx.

27.177.4.15 bool gdcm::MediaStorage::SetFromHeader (FileMetaInformation const & *fmi*)

27.177.4.16 bool gdcm::MediaStorage::SetFromModality (DataSet const & *ds*)

27.177.4.17 void gdcm::MediaStorage::SetFromSourceImageSequence (DataSet const & *ds*) [protected]

27.177.5 Friends And Related Function Documentation

27.177.5.1 std::ostream& operator<< (std::ostream & *os,* const MediaStorage & *ms*) [friend]

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

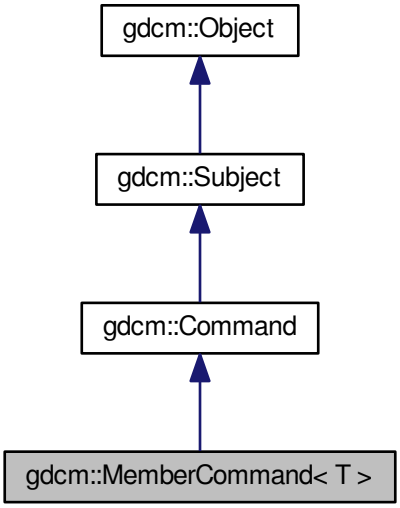
- gdcmMediaStorage.h

27.178 gdcm::MemberCommand< T > Class Template Reference

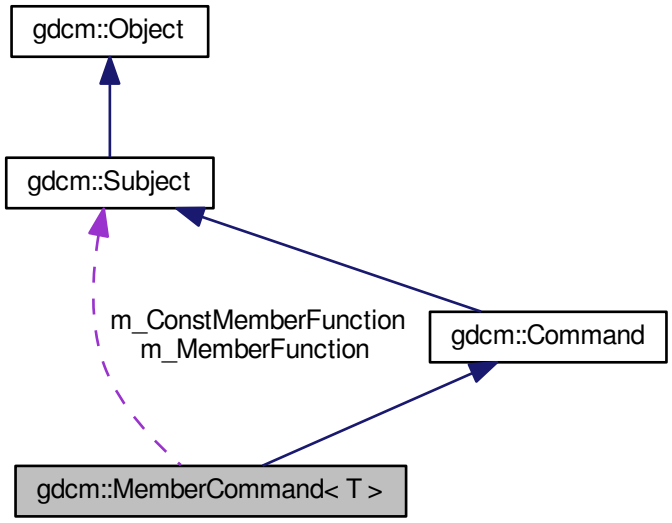
Command subclass that calls a pointer to a member function.

`#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

- virtual void Execute (Subject *caller, const Event &event)
- virtual void Execute (const Subject *caller, const Event &event)
- void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)
- void SetCallbackFunction (T *object, TConstMemberFunctionPointer memberFunction)

Static Public Member Functions

- static SmartPointer < MemberCommand > New ()

Protected Member Functions

- MemberCommand ()
- virtual ~MemberCommand ()

Protected Attributes

- TConstMemberFunctionPointer m_ConstMemberFunction
- TMemberFunctionPointer m_MemberFunction
- T * m_This

27.178.1 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.

27.178.2 Member Typedef Documentation

27.178.2.1 template<class T > typedef MemberCommand gdcm::MemberCommand< T >::Self

Standard class typedefs.

27.178.2.2 template<class T > typedef void(T::* gdcm::MemberCommand< T >::TConstMemberFunctionPointer)(const Subject *, const Event &)

27.178.2.3 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

27.178.3 Constructor & Destructor Documentation

27.178.3.1 template<class T > gdcm::MemberCommand< T >::MemberCommand ( ) [inline], [protected]

Referenced by gdcm::MemberCommand< T >::New().

27.178.3.2 template<class T > virtual gdcm::MemberCommand< T >::~MemberCommand ( ) [inline], [protected], [virtual]

27.178.4 Member Function Documentation

27.178.4.1 template<class T > virtual void gdcm::MemberCommand< T >::Execute (Subject * *caller,* const Event & *event*) [inline], [virtual]

Invoke the member function.

Implements gdcm::Command.

References gdcm::MemberCommand< T >::m_MemberFunction.

27.178.4.2 template<class T > virtual void gdcm::MemberCommand< T >::Execute (const Subject * *caller,* const Event & *event*) [inline], [virtual]

Invoke the member function with a const object.

Implements gdcm::Command.

References gdcm::MemberCommand< T >::m_ConstMemberFunction.

27.178.4.3 template<class T > static SmartPointer<MemberCommand> gdcm::MemberCommand< T >::New ( ) [inline], [static]

Method for creation through the object factory.

References gdcm::MemberCommand< T >::MemberCommand().

27.178.4.4 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.

References gdcm::MemberCommand< T >::m_MemberFunction, and gdcm::MemberCommand< T >::m_This.

27.178.4.5 template<class T > void gdcm::MemberCommand< T >::SetCallbackFunction (T * *object*, TConstMemberFunctionPointer *memberFunction*) [inline]

References gdcm::MemberCommand< T >::m_ConstMemberFunction, and gdcm::MemberCommand< T >::m_This.

27.178.5 Member Data Documentation

27.178.5.1 template<class T > TConstMemberFunctionPointer gdcm::MemberCommand< T >::m_ConstMemberFunction [protected]

Referenced by gdcm::MemberCommand< T >::Execute(), and gdcm::MemberCommand< T >::SetCallback↩ Function().

27.178.5.2 template<class T > TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction [protected]

Referenced by gdcm::MemberCommand< T >::Execute(), and gdcm::MemberCommand< T >::SetCallback↩ Function().

27.178.5.3 template<class T > T* gdcm::MemberCommand< T >::m_This [protected]

Referenced by gdcm::MemberCommand< T >::SetCallbackFunction().

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

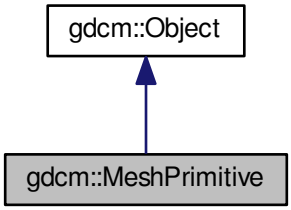
- gdcmCommand.h

27.179 gdcm::MeshPrimitive Class Reference

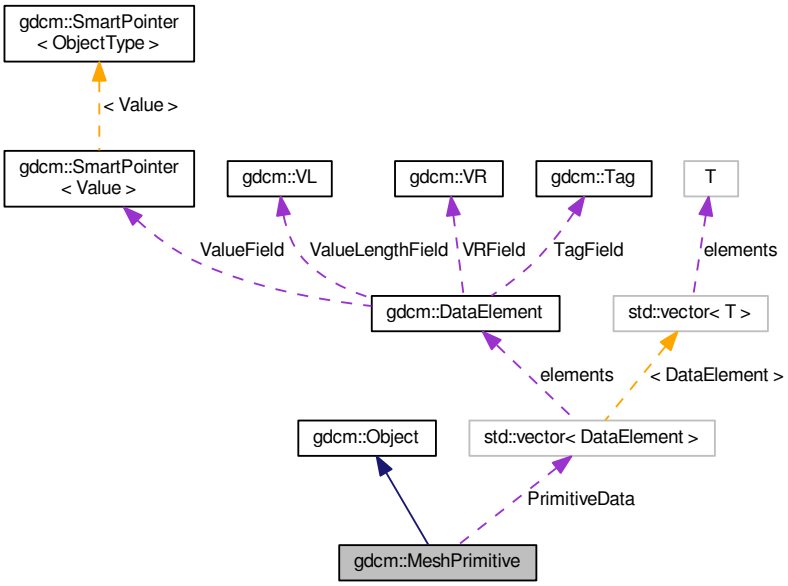
This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

`#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 ()
- virtual ~MeshPrimitive ()
- void AddPrimitiveData (DataElement const &de)
- unsigned int GetNumberOfPrimitivesData () const
- 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 ()
- MPType GetPrimitiveType () const
- void SetPrimitiveData (DataElement const &de)
- void SetPrimitiveData (const unsigned int idx, DataElement const &de)
- void SetPrimitivesData (PrimitivesData const &DEs)
- void SetPrimitiveType (const MPType type)

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

27.179.1 Detailed Description

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

27.179.2 Member Typedef Documentation

27.179.2.1 typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData

27.179.3 Member Enumeration Documentation

27.179.3.1 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

27.179.4 Constructor & Destructor Documentation

27.179.4.1 gdcm::MeshPrimitive::MeshPrimitive ( )

27.179.4.2 virtual gdcm::MeshPrimitive::~MeshPrimitive ( ) [virtual]

27.179.5 Member Function Documentation

27.179.5.1 void gdcm::MeshPrimitive::AddPrimitiveData (DataElement const & *de*)

27.179.5.2 static MPType gdcm::MeshPrimitive::GetMPType (const char * *type*) [static]

27.179.5.3 static const char* gdcm::MeshPrimitive::GetMPTypeString (const MPType *type*) [static]

27.179.5.4 unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ( ) const

27.179.5.5 const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( ) const

27.179.5.6 DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( )

27.179.5.7 const DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int *idx*) const

27.179.5.8 DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int *idx*)

27.179.5.9 const PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( ) const

27.179.5.10 PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( )

27.179.5.11 MPType gdcm::MeshPrimitive::GetPrimitiveType ( ) const

27.179.5.12 void gdcm::MeshPrimitive::SetPrimitiveData (DataElement const & *de*)

27.179.5.13 void gdcm::MeshPrimitive::SetPrimitiveData (const unsigned int *idx*, DataElement const & *de*)

27.179.5.14 void gdcm::MeshPrimitive::SetPrimitivesData (PrimitivesData const & *DEs*)

27.179.5.15 void gdcm::MeshPrimitive::SetPrimitiveType (const MPType *type*)

27.179.6 Member Data Documentation

27.179.6.1 PrimitivesData gdcm::MeshPrimitive::PrimitiveData [protected]

27.179.6.2 MPType gdcm::MeshPrimitive::PrimitiveType [protected]

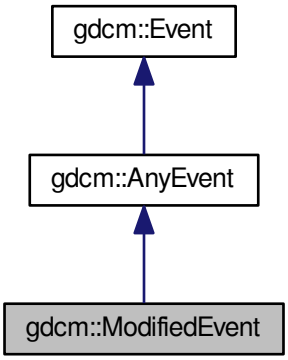
The documentation for this class was generated from the following file:

- gdcmMeshPrimitive.h

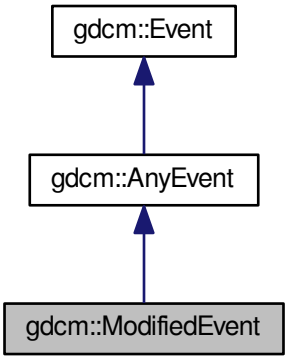
27.180 gdcm::ModifiedEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::ModifiedEvent:



Collaboration diagram for gdcm::ModifiedEvent:



Additional Inherited Members

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

- gdcmEvent.h

27.181 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 ()
- void AddMacro (const char *include)
- void AddModuleEntry (const Tag &tag, const ModuleEntry &module)

  Will add a ModuleEntry direcly 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)

27.181.1 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.

27.181.2 Member Typedef Documentation

27.181.2.1 typedef std::vector<std::string> gdcm::Module::ArrayIncludeMacrosType

27.181.2.2 typedef std::map<Tag, ModuleEntry> gdcm::Module::MapModuleEntry

27.181.3 Constructor & Destructor Documentation

27.181.3.1 gdcm::Module::Module ( ) [inline]

27.181.4 Member Function Documentation

27.181.4.1 void gdcm::Module::AddMacro (const char * *include*) [inline]

27.181.4.2 void gdcm::Module::AddModuleEntry (const Tag & *tag,* const ModuleEntry & *module*) [inline]

Will add a ModuleEntry direcly at root-level. See Macro for nested-included level.

27.181.4.3 void gdcm::Module::Clear ( ) [inline]

27.181.4.4 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.

27.181.4.5 const ModuleEntry& gdcm::Module::GetModuleEntryInMacros (Macros const & *macros,* const Tag & *tag*) const

Examples:

TraverseModules.cxx.

27.181.4.6 const char* gdcm::Module::GetName ( ) const [inline]

27.181.4.7 void gdcm::Module::SetName (const char * *name*) [inline]

27.181.4.8 bool gdcm::Module::Verify (const DataSet & *ds,* Usage const & *usage*) const

27.181.5 Friends And Related Function Documentation

27.181.5.1 std::ostream& operator<< (std::ostream & *_os,* const Module & *_val*) [friend]

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

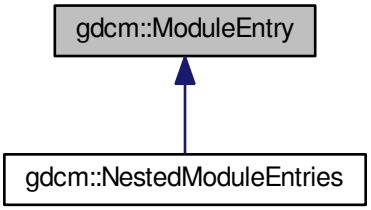
- gdcmModule.h

27.182 gdcm::ModuleEntry Class Reference

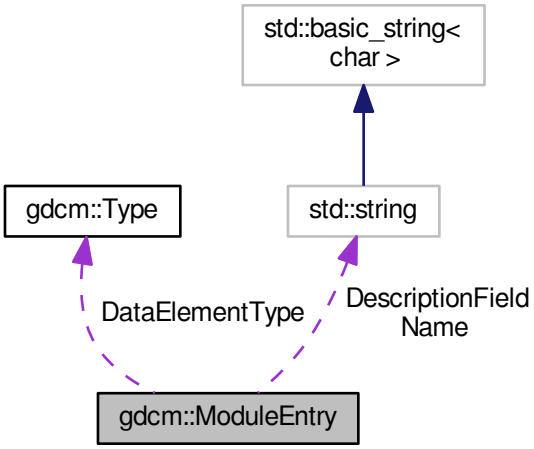
Class for representing a ModuleEntry.

`#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 ()
- 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)

27.182.1 Detailed Description

Class for representing a ModuleEntry.

Note

bla

See also

DictEntry

Examples:

TraverseModules.cxx.

27.182.2 Member Typedef Documentation

27.182.2.1 typedef std::string gdcm::ModuleEntry::Description

27.182.3 Constructor & Destructor Documentation

27.182.3.1 gdcm::ModuleEntry::ModuleEntry (const char * *name* = " ", const char * *type* = "3", const char * *description* = " ") [inline]

References gdcm::Type::GetTypeType().

27.182.3.2 virtual gdcm::ModuleEntry::~ModuleEntry ( ) [inline], [virtual]

27.182.4 Member Function Documentation

27.182.4.1 const Description& gdcm::ModuleEntry::GetDescription ( ) const [inline]

27.182.4.2 const char* gdcm::ModuleEntry::GetName ( ) const [inline]

27.182.4.3 const Type& gdcm::ModuleEntry::GetType ( ) const [inline]

Examples:

TraverseModules.cxx.

27.182.4.4 void gdcm::ModuleEntry::SetDescription (const char * *d*) [inline]

27.182.4.5 void gdcm::ModuleEntry::SetName (const char * *name*) [inline]

27.182.4.6 void gdcm::ModuleEntry::SetType (const Type & *type*) [inline]

27.182.5 Friends And Related Function Documentation

27.182.5.1 std::ostream& operator<< (std::ostream & *_os,* const ModuleEntry & *_val*) [friend]

27.182.6 Member Data Documentation

27.182.6.1 Type gdcm::ModuleEntry::DataElementType [protected]

Referenced by gdcm::operator<<().

27.182.6.2 Description gdcm::ModuleEntry::DescriptionField [protected]

Referenced by gdcm::operator<<().

27.182.6.3 std::string gdcm::ModuleEntry::Name [protected]

Referenced by gdcm::operator<<().

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

- gdcmModuleEntry.h

27.183 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 ()
- 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)

27.183.1 Detailed Description

Class for representing a Modules.

Note

bla

See also

Module

Examples:

TraverseModules.cxx.

27.183.2 Member Typedef Documentation

27.183.2.1 typedef std::map<std::string, Module> gdcm::Modules::ModuleMapType

27.183.3 Constructor & Destructor Documentation

27.183.3.1 gdcm::Modules::Modules ( ) [inline]

27.183.4 Member Function Documentation

27.183.4.1 void gdcm::Modules::AddModule (const char * *ref*, const Module & *module*) [inline]

27.183.4.2 void gdcm::Modules::Clear ( ) [inline]

27.183.4.3 const Module& gdcm::Modules::GetModule (const char * *name*) const [inline]

Examples:

TraverseModules.cxx.

27.183.4.4 bool gdcm::Modules::IsEmpty ( ) const [inline]

27.183.5 Friends And Related Function Documentation

27.183.5.1 std::ostream& operator<< (std::ostream & *_os*, const Modules & *_val*) [friend]

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

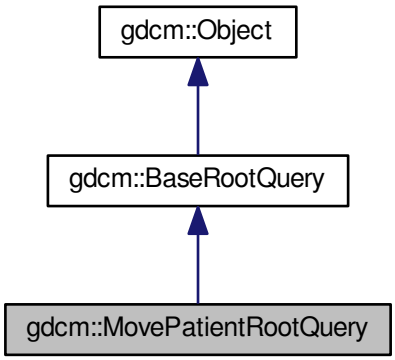
- gdcmModules.h

27.184 gdcm::MovePatientRootQuery Class Reference

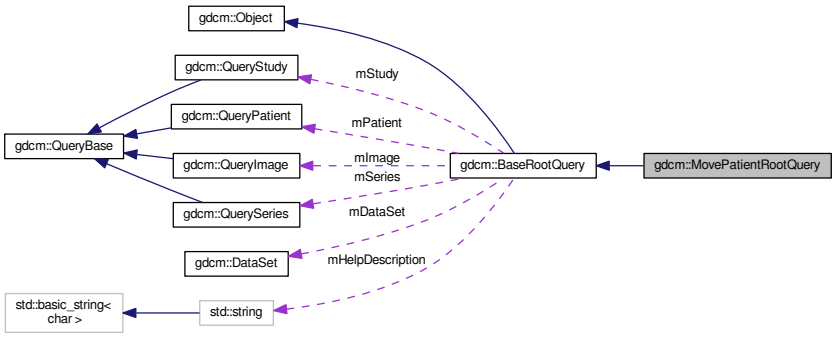
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

`#include <gdcmMovePatientRootQuery.h>`

Inheritance diagram for gdcm::MovePatientRootQuery:



Collaboration diagram for gdcm::MovePatientRootQuery:



Public Member Functions

- MovePatientRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

27.184.1 Detailed Description

MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

27.184.2 Constructor & Destructor Documentation

27.184.2.1 gdcm::MovePatientRootQuery::MovePatientRootQuery ( )

27.184.3 Member Function Documentation

27.184.3.1 UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseRootQuery.

27.184.3.2 std::vector<Tag> gdcm::MovePatientRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [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.

27.184.3.3 void gdcm::MovePatientRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [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.

27.184.3.4 bool gdcm::MovePatientRootQuery::ValidateQuery (bool *inStrict* = true) const [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.

27.184.4 Friends And Related Function Documentation

27.184.4.1 friend class QueryFactory [friend]

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

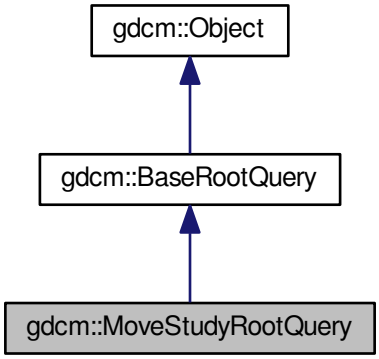
- gdcmMovePatientRootQuery.h

27.185 gdcm::MoveStudyRootQuery Class Reference

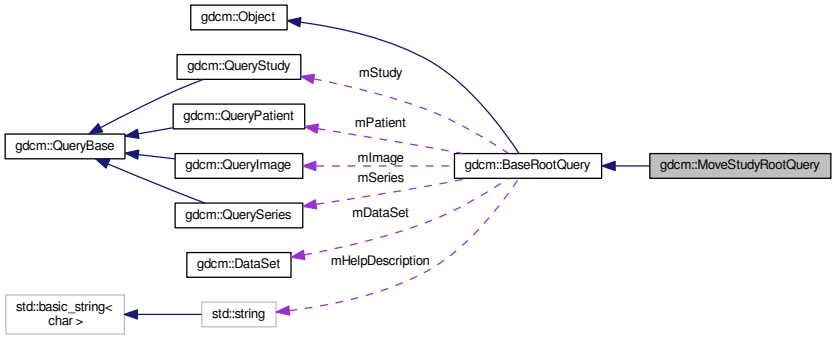
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

`#include <gdcmMoveStudyRootQuery.h>`

Inheritance diagram for gdcm::MoveStudyRootQuery:



Collaboration diagram for gdcm::MoveStudyRootQuery:



Public Member Functions

- MoveStudyRootQuery ()
- UIDs::TSName GetAbstractSyntaxUID () const
- std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)
- void InitializeDataSet (const EQueryLevel &inQueryLevel)
- bool ValidateQuery (bool inStrict=true) const

Friends

- class QueryFactory

Additional Inherited Members

27.185.1 Detailed Description

MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

27.185.2 Constructor & Destructor Documentation

27.185.2.1 gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )

27.185.3 Member Function Documentation

27.185.3.1 UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements gdcm::BaseRootQuery.

27.185.3.2 std::vector<Tag> gdcm::MoveStudyRootQuery::GetTagListByLevel (const EQueryLevel & *inQueryLevel*) [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.

27.185.3.3 void gdcm::MoveStudyRootQuery::InitializeDataSet (const EQueryLevel & *inQueryLevel*) [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.

27.185.3.4 bool gdcm::MoveStudyRootQuery::ValidateQuery (bool *inStrict* = true) const [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.

27.185.4 Friends And Related Function Documentation

27.185.4.1 friend class QueryFactory [friend]

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

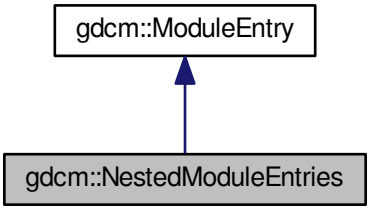
- gdcmMoveStudyRootQuery.h

27.186 gdcm::NestedModuleEntries Class Reference

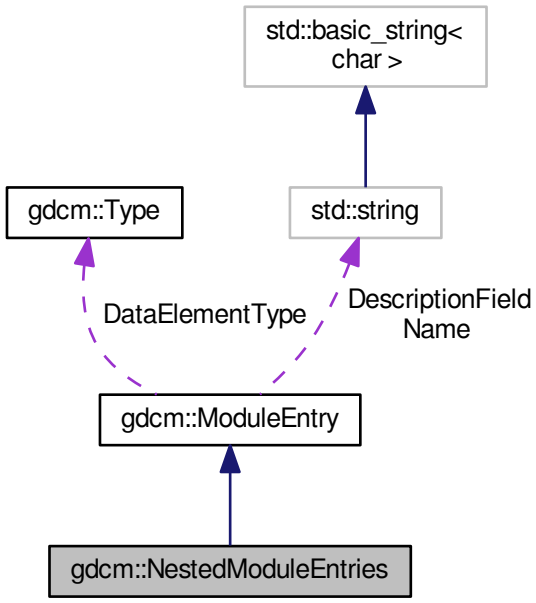
Class for representing a NestedModuleEntries.

`#include <gdcmNestedModuleEntries.h>`

Inheritance diagram for gdcm::NestedModuleEntries:



Collaboration diagram for gdcm::NestedModuleEntries:



Public Types

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

Public Member Functions

- NestedModuleEntries (const char *name="", const char *type="3", const char *description="")
- void AddModuleEntry (const ModuleEntry &me)
- const ModuleEntry & GetModuleEntry (SizeType idx) const
- ModuleEntry & GetModuleEntry (SizeType idx)
- SizeType GetNumberOfModuleEntries ()

Friends

- std::ostream & operator<< (std::ostream &_os, const NestedModuleEntries &_val)

Additional Inherited Members

27.186.1 Detailed Description

Class for representing a NestedModuleEntries.

Note

bla

See also

ModuleEntry

27.186.2 Member Typedef Documentation

27.186.2.1 typedef std::vector<ModuleEntry>::size_type gdcm::NestedModuleEntries::SizeType

27.186.3 Constructor & Destructor Documentation

27.186.3.1 gdcm::NestedModuleEntries::NestedModuleEntries (const char * *name* = "", const char * *type* = "3", const char * *description* = "") [inline]

27.186.4 Member Function Documentation

27.186.4.1 void gdcm::NestedModuleEntries::AddModuleEntry (const ModuleEntry & *me*) [inline]

27.186.4.2 const ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (SizeType *idx*) const [inline]

27.186.4.3 ModuleEntry& gdcm::NestedModuleEntries::GetModuleEntry (SizeType *idx*) [inline]

27.186.4.4 SizeType gdcm::NestedModuleEntries::GetNumberOfModuleEntries ( ) [inline]

27.186.5 Friends And Related Function Documentation

27.186.5.1 std::ostream& operator<< (std::ostream & *_os,* const NestedModuleEntries & *_val*) [friend]

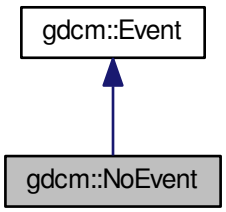
The documentation for this class was generated from the following file:

- gdcmNestedModuleEntries.h

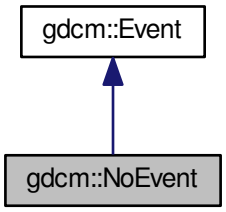
27.187 gdcm::NoEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::NoEvent:



Collaboration diagram for gdcm::NoEvent:



Additional Inherited Members

27.187.1 Detailed Description

Define some common GDCM events

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

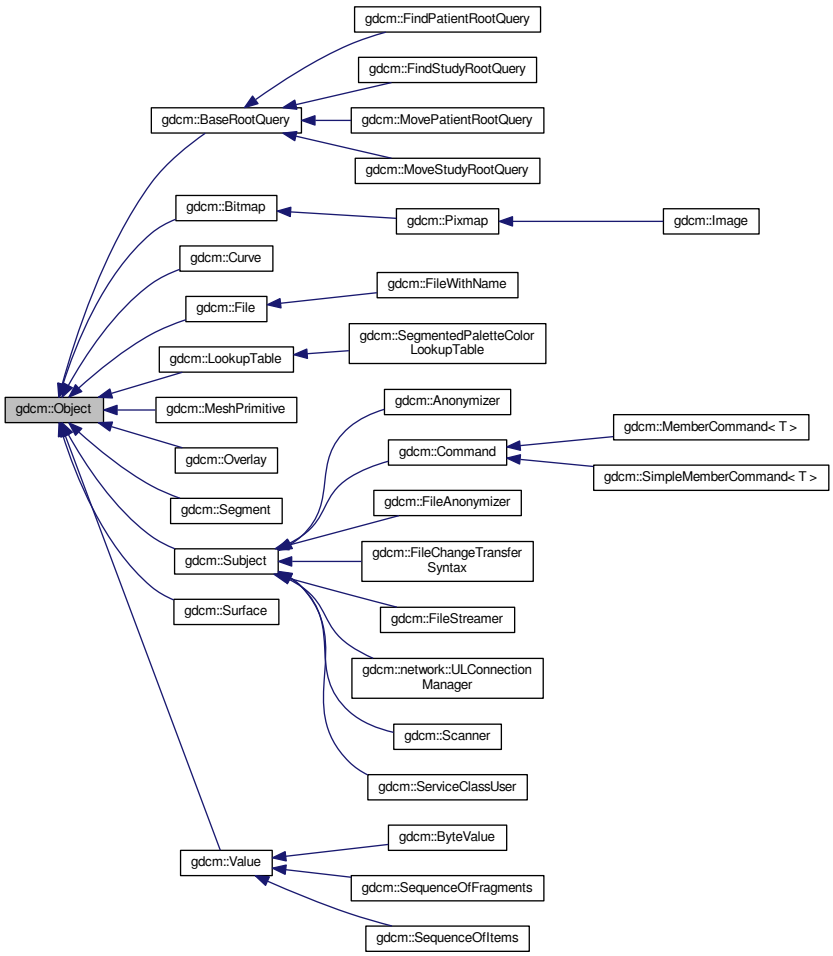
- gdcmEvent.h

27.188 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

27.188.1 Detailed Description

Object.

Note

main superclass for object that want to use SmartPointer invasive ref counting system

See also

SmartPointer

27.188.2 Constructor & Destructor Documentation

27.188.2.1 gdcm::Object::Object ( ) [inline]

27.188.2.2 virtual gdcm::Object::~Object ( ) [inline], [virtual]

27.188.2.3 gdcm::Object::Object (const Object &) [inline]

Special requirement for copy/cstor, assignment operator.

27.188.3 Member Function Documentation

27.188.3.1 void gdcm::Object::operator= (const Object &) [inline]

27.188.3.2 virtual void gdcm::Object::Print (std::ostream &) const [inline], [virtual]

Reimplemented in gdcm::SequenceOfFragments, gdcm::ByteValue, gdcm::SequenceOfItems, gdcm::BaseRootQuery, gdcm::Scanner, gdcm::Image, gdcm::Curve, gdcm::Overlay, gdcm::Bitmap, gdcm::LookupTable, gdcm::Pixmap, and gdcm::SegmentedPaletteColorLookupTable.

Examples:

ReadAndDumpDICOMDIR.cxx.

Referenced by gdcm::operator<<().

27.188.3.3 void gdcm::Object::Register ( ) [inline], [protected]

27.188.3.4 void gdcm::Object::UnRegister ( ) [inline], [protected]

27.188.4 Friends And Related Function Documentation

27.188.4.1 std::ostream& operator<< (std::ostream & *os,* const Object & *obj*) [friend]

27.188.4.2 template<class ObjectType > friend class SmartPointer [friend]

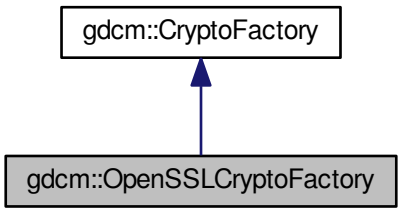
The documentation for this class was generated from the following file:

- gdcmObject.h

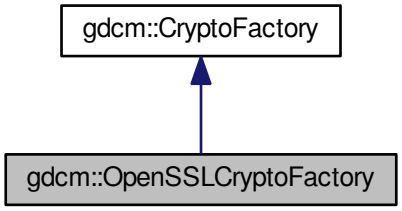
27.189 gdcm::OpenSSLCryptoFactory Class Reference

`#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 ()

Additional Inherited Members

27.189.1 Constructor & Destructor Documentation

27.189.1.1 gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory (CryptoLib *id*) [inline]

References gdcmDebugMacro.

27.189.2 Member Function Documentation

27.189.2.1 CryptographicMessageSyntax* gdcm::OpenSSLCryptoFactory::CreateCMSProvider ( ) [inline], [virtual]

Implements gdcm::CryptoFactory.

27.189.2.2 void gdcm::OpenSSLCryptoFactory::InitOpenSSL ( ) [protected]

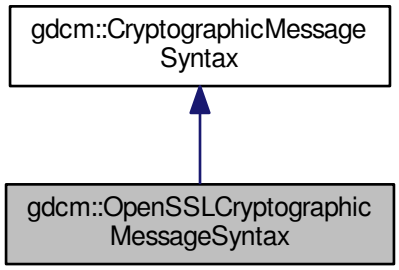
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptoFactory.h

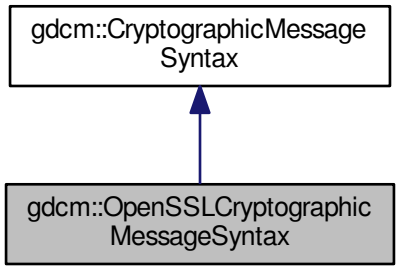
27.190 gdcm::OpenSSLCryptographicMessageSyntax Class Reference

```
#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)

Additional Inherited Members

27.190.1 Constructor & Destructor Documentation

27.190.1.1 gdcm::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ( )

27.190.1.2 gdcm::OpenSSLCryptographicMessageSyntax::~OpenSSLCryptographicMessageSyntax ( )

27.190.2 Member Function Documentation

27.190.2.1 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.

27.190.2.2 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.

27.190.2.3 CipherTypes gdcm::OpenSSLCryptographicMessageSyntax::GetCipherType ( ) const [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.190.2.4 bool gdcm::OpenSSLCryptographicMessageSyntax::ParseCertificateFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.190.2.5 bool gdcm::OpenSSLCryptographicMessageSyntax::ParseKeyFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.190.2.6 void gdcm::OpenSSLCryptographicMessageSyntax::SetCipherType (CipherTypes *type*) [virtual]

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

27.190.2.7 bool gdcm::OpenSSLCryptographicMessageSyntax::SetPassword (const char * *pass*, size_t *passLen*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

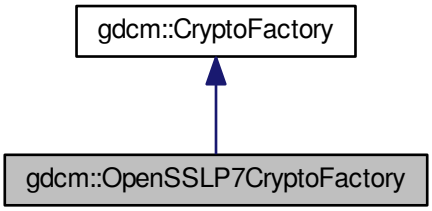
The documentation for this class was generated from the following file:

- gdcmOpenSSLCryptographicMessageSyntax.h

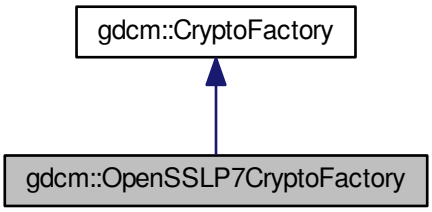
27.191 gdcm::OpenSSLP7CryptoFactory Class Reference

`#include <gdcmOpenSSLP7CryptoFactory.h>`

Inheritance diagram for gdcm::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcm::OpenSSLP7CryptoFactory:



Public Member Functions

- OpenSSLP7CryptoFactory (CryptoLib id)
- CryptographicMessageSyntax * CreateCMSProvider ()

Additional Inherited Members

27.191.1 Constructor & Destructor Documentation

27.191.1.1 gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory (CryptoLib *id*) [inline]

References gdcmDebugMacro.

27.191.2 Member Function Documentation

27.191.2.1 CryptographicMessageSyntax* gdcm::OpenSSLP7CryptoFactory::CreateCMSProvider ( ) [inline], [virtual]

Implements gdcm::CryptoFactory.

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

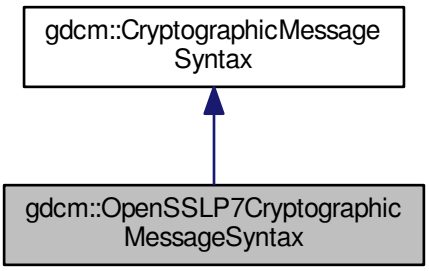
- gdcmOpenSSLP7CryptoFactory.h

27.192 gdcm::OpenSSLP7CryptographicMessageSyntax Class Reference

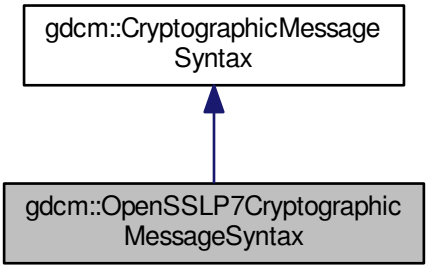
Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

`#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)

Additional Inherited Members

27.192.1 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`

27.192.2 Constructor & Destructor Documentation

27.192.2.1 gdcm::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ( )

27.192.2.2 gdcm::OpenSSLP7CryptographicMessageSyntax::~OpenSSLP7CryptographicMessageSyntax ( )

27.192.3 Member Function Documentation

27.192.3.1 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.

27.192.3.2 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.

27.192.3.3 CipherTypes gdcm::OpenSSLP7CryptographicMessageSyntax::GetCipherType ( ) const [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.192.3.4 bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseCertificateFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.192.3.5 bool gdcm::OpenSSLP7CryptographicMessageSyntax::ParseKeyFile (const char * *filename*) [virtual]

Implements gdcm::CryptographicMessageSyntax.

27.192.3.6 void gdcm::OpenSSLP7CryptographicMessageSyntax::SetCipherType (CipherTypes *type*) [virtual]

Set Cipher Type. Default is: AES256_CIPHER

Implements gdcm::CryptographicMessageSyntax.

27.192.3.7 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

27.193 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 ()
- 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)

27.193.1 Detailed Description

class to handle Orientation

27.193.2 Member Enumeration Documentation

27.193.2.1 enum gdcm::Orientation::OrientationType

Enumerator

UNKNOWN

AXIAL

CORONAL

SAGITTAL

OBLIQUE

27.193.3 Constructor & Destructor Documentation

27.193.3.1 gdcm::Orientation::Orientation ( )

27.193.3.2 gdcm::Orientation::~Orientation ( )

27.193.4 Member Function Documentation

27.193.4.1 static const char* gdcm::Orientation::GetLabel (OrientationType *type*) [static]

Return the label of an Orientation.

27.193.4.2 static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (double *x*, double *y*, double *z*) [static], [protected]

27.193.4.3 static double gdcm::Orientation::GetObliquityThresholdCosineValue ( ) [static]

27.193.4.4 static 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

27.193.4.5 void gdcm::Orientation::Print (std::ostream &) const

Print.

Referenced by gdcm::operator<<().

27.193.4.6 static void gdcm::Orientation::SetObliquityThresholdCosineValue (double *val*) [static]

ObliquityThresholdCosineValue stuff.

27.193.5 Friends And Related Function Documentation

27.193.5.1 std::ostream& operator<< (std::ostream & *_os*, const Orientation & *o*) [friend]

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

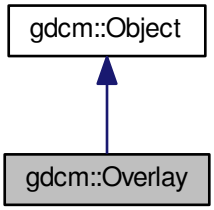
- gdcmOrientation.h

27.194 gdcm::Overlay Class Reference

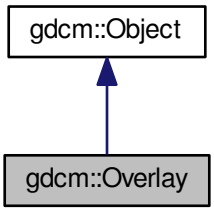
Overlay class.

`#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 ()
- 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 wether or no the OverlayData is in the Pixel Data:
- bool IsZero () const

  return true if all bits are set to 0
- void Print (std::ostream &) const

  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 numberoffframes)

  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:

Static Public Member Functions

- static const char * GetOverlayTypeAsString (OverlayType ot)
- static OverlayType GetOverlayTypeFromString (const char *)

Additional Inherited Members

27.194.1 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 differenciate them (other than the group tag).

Example:

27.194.2 Member Enumeration Documentation

27.194.2.1 enum gdcm::Overlay::OverlayType

Enumerator

Invalid

Graphics

ROI

27.194.3 Constructor & Destructor Documentation

27.194.3.1 gdcm::Overlay::Overlay ( )

27.194.3.2 gdcm::Overlay::~Overlay ( )

27.194.3.3 gdcm::Overlay::Overlay (Overlay const & *ov*)

27.194.4 Member Function Documentation

27.194.4.1 void gdcm::Overlay::Decompress (std::ostream & *os*) const

Decode the internal OverlayData (packed bits) into unpacked representation.

27.194.4.2 unsigned short gdcm::Overlay::GetBitPosition ( ) const

return bit position

27.194.4.3 unsigned short gdcm::Overlay::GetBitsAllocated ( ) const

return bits allocated

27.194.4.4 unsigned short gdcm::Overlay::GetColumns ( ) const

get columns

27.194.4.5 const char* gdcm::Overlay::GetDescription ( ) const

get description

27.194.4.6 unsigned short gdcm::Overlay::GetGroup ( ) const

Get Group number.

27.194.4.7 const signed short* gdcm::Overlay::GetOrigin ( ) const

get origin

27.194.4.8 const ByteValue& gdcm::Overlay::GetOverlayData ( ) const

Return the Overlay Data as ByteValue: Not thread safe

27.194.4.9 static const char* gdcm::Overlay::GetOverlayTypeAsString (OverlayType *ot*) [static]

27.194.4.10 static OverlayType gdcm::Overlay::GetOverlayTypeFromString (const char *) [static]

27.194.4.11 unsigned short gdcm::Overlay::GetRows ( ) const

get rows

27.194.4.12 const char* gdcm::Overlay::GetType ( ) const

get type

27.194.4.13 OverlayType gdcm::Overlay::GetTypeAsEnum ( ) const

27.194.4.14 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()

27.194.4.15 size_t gdcm::Overlay::GetUnpackBufferLength ( ) const

Retrieve the size of the buffer needed to hold the Overlay as specified by Col & Row parameters

27.194.4.16 bool gdcm::Overlay::GrabOverlayFromPixelData (DataSet const & *ds*)

27.194.4.17 bool gdcm::Overlay::IsEmpty ( ) const

Return whether or not the Overlay is empty:

27.194.4.18 bool gdcm::Overlay::IsInPixelData ( ) const

return if the Overlay is stored in the pixel data or not

27.194.4.19 void gdcm::Overlay::IsInPixelData (bool *b*)

Set wether or no the OverlayData is in the Pixel Data:

27.194.4.20 bool gdcm::Overlay::IsZero ( ) const

return true if all bits are set to 0

27.194.4.21 void gdcm::Overlay::Print (std::ostream &) const [virtual]

Print.

Reimplemented from gdcm::Object.

27.194.4.22 void gdcm::Overlay::SetBitPosition (unsigned short *bitposition*)

set bit position

27.194.4.23 void gdcm::Overlay::SetBitsAllocated (unsigned short *bitsallocated*)

set bits allocated

27.194.4.24 void gdcm::Overlay::SetColumns (unsigned short *columns*)

set columns

27.194.4.25 void gdcm::Overlay::SetDescription (const char * *description*)

set description

27.194.4.26 void gdcm::Overlay::SetFrameOrigin (unsigned short *frameorigin*)

set frame origin

27.194.4.27 void gdcm::Overlay::SetGroup (unsigned short *group*)

Set Group number.

27.194.4.28 void gdcm::Overlay::SetNumberOfFrames (unsigned int *numberofframes*)

set number of frames

27.194.4.29 void gdcm::Overlay::SetOrigin (const signed short *origin[2]*)

set origin

27.194.4.30 void gdcm::Overlay::SetOverlay (const char * *array,* size_t *length*)

set overlay from byte array + length

27.194.4.31 void gdcm::Overlay::SetRows (unsigned short *rows*)

set rows

27.194.4.32 void gdcm::Overlay::SetType (const char * *type*)

set type

27.194.4.33 void gdcm::Overlay::Update (const DataElement & *de*)

Update overlay from data element de:

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

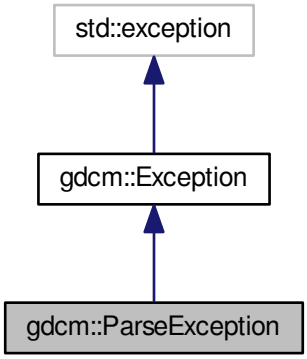
- gdcmOverlay.h

27.195 gdcm::ParseException Class Reference

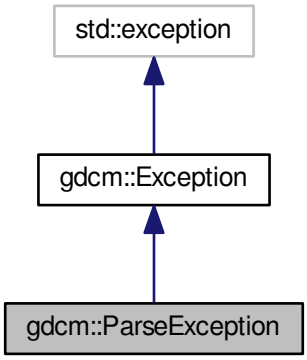
ParseException Standard exception handling object.

`#include <gdcmParseException.h>`

Inheritance diagram for gdcm::ParseException:



Collaboration diagram for gdcm::ParseException:



Public Member Functions

- ParseException ()
- virtual ~ParseException () throw ()
- const DataElement & GetLastElement () const
- ParseException & operator= (const ParseException &orig)
- void SetLastElement (DataElement &de)

27.195.1 Detailed Description

ParseException Standard exception handling object.

27.195.2 Constructor & Destructor Documentation

27.195.2.1 gdcm::ParseException::ParseException ( ) [inline]

27.195.2.2 virtual gdcm::ParseException::~ParseException ( ) throw) [inline], [virtual]

27.195.3 Member Function Documentation

27.195.3.1 const DataElement& gdcm::ParseException::GetLastElement ( ) const [inline]

27.195.3.2 ParseException& gdcm::ParseException::operator= (const ParseException & *orig*) [inline]

Assignment operator.

27.195.3.3 void gdcm::ParseException::SetLastElement (DataElement & *de*) [inline]

Equivalence operator.

Referenced by gdcm::Fragment::ReadBacktrack(), and gdcm::Fragment::ReadValue().

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

- gdcmParseException.h

27.196 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 ()

27.196.1 Detailed Description

Parser ala XML_Parser from expat (SAX)

Detailled description here

Note

Simple API for DICOM

27.196.2 Member Typedef Documentation

27.196.2.1 typedef void(* gdcm::Parser::EndElementHandler)(void *userData, const Tag &name)

27.196.2.2 typedef void(* gdcm::Parser::StartElementHandler)(void *userData, const Tag &tag, const char *atts[])

27.196.3 Member Enumeration Documentation

27.196.3.1 enum gdcm::Parser::ErrorType

Enumerator

NoError

NoMemoryError

SyntaxError

NoElementsError

TagMismatchError

DuplicateAttributeError

JunkAfterDocElementError

UndefinedEntityError

UnexpectedStateError

27.196.4 Constructor & Destructor Documentation

27.196.4.1 gdcm::Parser::Parser ( ) [inline]

27.196.4.2 gdcm::Parser::~Parser ( ) [inline]

27.196.5 Member Function Documentation

27.196.5.1 char* gdcm::Parser::GetBuffer (int *len*) [protected]

27.196.5.2 unsigned long gdcm::Parser::GetCurrentByteIndex ( ) const

27.196.5.3 ErrorType gdcm::Parser::GetErrorCode ( ) const

27.196.5.4 static const char* gdcm::Parser::GetErrorString (ErrorType const & *err*) [static]

27.196.5.5 void* gdcm::Parser::GetUserData ( ) const

27.196.5.6 bool gdcm::Parser::Parse (const char * *s*, int *len*, bool *isFinal*)

27.196.5.7 bool gdcm::Parser::ParseBuffer (int *len*, bool *isFinal*) [protected]

27.196.5.8 ErrorType gdcm::Parser::Process ( ) [protected]

27.196.5.9 void gdcm::Parser::SetElementHandler (StartElementHandler *start*, EndElementHandler *end*)

27.196.5.10 void gdcm::Parser::SetUserData (void * *userData*)

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

- gdcmParser.h

27.197 gdcm::Patient Class Reference

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

`#include <gdcmPatient.h>`

Public Member Functions

- Patient ()

27.197.1 Detailed Description

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

27.197.2 Constructor & Destructor Documentation

27.197.2.1 gdcm::Patient::Patient ( ) [inline]

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

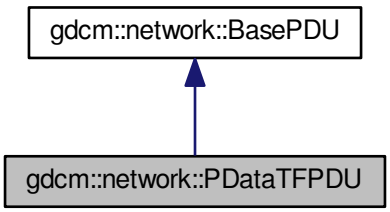
- gdcmPatient.h

27.198 gdcm::network::PDataTFPDU Class Reference

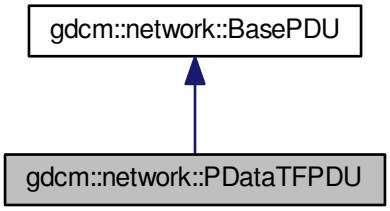
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

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

Protected Member Functions

- std::istream & ReadInto (std::istream &is, std::ostream &os)

27.198.1 Detailed Description

PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.

27.198.2 Member Typedef Documentation

27.198.2.1 typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType

27.198.3 Constructor & Destructor Documentation

27.198.3.1 gdcm::network::PDataTFPDU::PDataTFPDU ( )

27.198.4 Member Function Documentation

27.198.4.1 void gdcm::network::PDataTFPDU::AddPresentationDataValue (PresentationDataValue const & *pdv*) [inline]

27.198.4.2 SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const [inline]

27.198.4.3 PresentationDataValue const& gdcm::network::PDataTFPDU::GetPresentationDataValue (SizeType *i*) const [inline]

27.198.4.4 bool gdcm::network::PDataTFPDU::IsLastFragment ( ) const [virtual]

Implements gdcm::network::BasePDU.

27.198.4.5 void gdcm::network::PDataTFPDU::Print (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

27.198.4.6 std::istream& gdcm::network::PDataTFPDU::Read (std::istream & *is*) [virtual]

Implements gdcm::network::BasePDU.

27.198.4.7 std::istream& gdcm::network::PDataTFPDU::ReadInto (std::istream & *is*, std::ostream & *os*) [protected]

27.198.4.8 size_t gdcm::network::PDataTFPDU::Size ( ) const [virtual]

Implements gdcm::network::BasePDU.

27.198.4.9 const std::ostream& gdcm::network::PDataTFPDU::Write (std::ostream & *os*) const [virtual]

Implements gdcm::network::BasePDU.

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

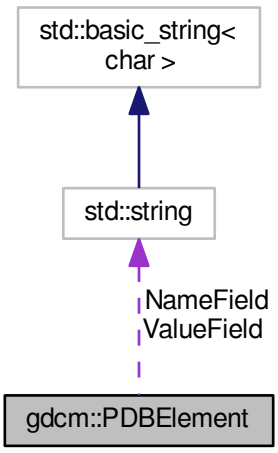
- gdcmPDataTFPDU.h

27.199 gdcm::PDBElement Class Reference

Class to represent a PDB Element.

`#include <gdcmPDBElement.h>`

Collaboration diagram for gdcm::PDBElement:



Public Member Functions

- PDBElement ()
- 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)

27.199.1 Detailed Description

Class to represent a PDB Element.

See also

PDBHeader

27.199.2 Constructor & Destructor Documentation

27.199.2.1 gdcm::PDBElement::PDBElement ( ) [inline]

27.199.3 Member Function Documentation

27.199.3.1 const char* gdcm::PDBElement::GetName ( ) const [inline]

Set/Get Name.

27.199.3.2 const char* gdcm::PDBElement::GetValue ( ) const [inline]

Set/Get Value.

27.199.3.3 bool gdcm::PDBElement::operator== (const PDBElement & *de*) const [inline]

References NameField, and ValueField.

27.199.3.4 void gdcm::PDBElement::SetName (const char * *name*) [inline]

27.199.3.5 void gdcm::PDBElement::SetValue (const char * *value*) [inline]

27.199.4 Friends And Related Function Documentation

27.199.4.1 std::ostream& operator<< (std::ostream & *os,* const PDBElement & *val*) [friend]

27.199.5 Member Data Documentation

27.199.5.1 std::string gdcm::PDBElement::NameField [protected]

Referenced by gdcm::operator<<(), and operator==().

27.199.5.2 std::string gdcm::PDBElement::ValueField [protected]

Referenced by gdcm::operator<<(), and operator==().

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

- gdcmPDBElement.h

27.200 gdcm::PDBHeader Class Reference

Class for PDBHeader.

`#include <gdcmPDBHeader.h>`

Public Member Functions

- PDBHeader ()
- ~PDBHeader ()
- 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)

27.200.1 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.

See also

CSAHeader

27.200.2 Constructor & Destructor Documentation

27.200.2.1 gdcm::PDBHeader::PDBHeader ( ) [inline]

27.200.2.2 gdcm::PDBHeader::~PDBHeader ( ) [inline]

27.200.3 Member Function Documentation

27.200.3.1 bool gdcm::PDBHeader::FindPDBElementByName (const char * *name*)

Return true if the PDB element matching name is found or not.

27.200.3.2 const PDBElement& gdcm::PDBHeader::GetPDBEEnd ( ) const [protected]

27.200.3.3 const PDBElement& gdcm::PDBHeader::GetPDBElementByName (const char * *name*)

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

27.200.3.4 static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag ( ) [static]

Return the Private Tag where the PDB header is stored within a DICOM DataSet.

27.200.3.5 bool gdcm::PDBHeader::LoadFromDataElement (DataElement const & *de*)

Load the PDB Header from a DataElement of a DataSet.

27.200.3.6 void gdcm::PDBHeader::Print (std::ostream & *os*) const

Print.

Referenced by gdcm::operator<<().

27.200.4 Friends And Related Function Documentation

27.200.4.1 std::ostream& operator<< (std::ostream & *_os,* const PDBHeader & *d*) [friend]

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

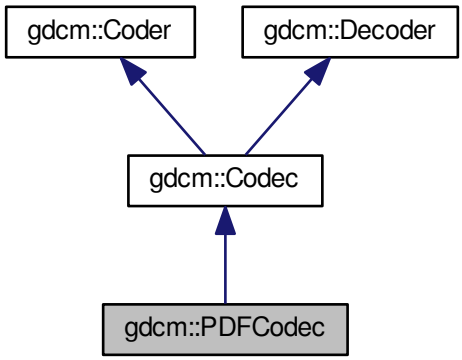
- gdcmPDBHeader.h

27.201 gdcm::PDFCodec Class Reference

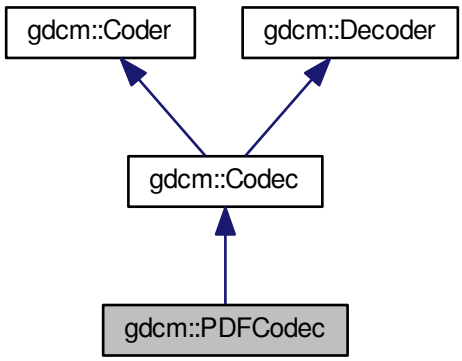
PDFCodec class.

`#include <gdcmPDFCodec.h>`

Inheritance diagram for gdcm::PDFCodec:



Collaboration diagram for gdcm::PDFCodec:



Public Member Functions

- PDFCodec ()
- ~PDFCodec ()
- bool CanCode (TransferSyntax const &) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &) const

  Return whether this decoder support this transfer syntax (can decode it)
- bool Decode (DataElement const &is, DataElement &os)

  Decode.

Additional Inherited Members

27.201.1 Detailed Description

PDFCodec class.

27.201.2 Constructor & Destructor Documentation

27.201.2.1 gdcm::PDFCodec::PDFCodec ( )

27.201.2.2 gdcm::PDFCodec::~PDFCodec ( )

27.201.3 Member Function Documentation

27.201.3.1 bool gdcm::PDFCodec::CanCode (TransferSyntax const &) const [inline], [virtual]

Return whether this coder support this transfer syntax (can code it)

Implements gdcm::Coder.

27.201.3.2 bool gdcm::PDFCodec::CanDecode (TransferSyntax const &) const [inline], [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implements gdcm::Decoder.

27.201.3.3 bool gdcm::PDFCodec::Decode (DataElement const & , DataElement &) [virtual]

Decode.

Reimplemented from gdcm::Decoder.

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

- gdcmPDFCodec.h

27.202 gdcm::network::PDUFactory Class Reference

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.

`#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)
- static std::vector< BasePDU * > CreateCStoreRSPPDU (const DataSet *inDataSet, const BasePDU *inPC)
- static EEventID DetermineEventByPDU (const BasePDU *inPDU)
- static std::vector < PresentationDataValue > GetPDVs (const std::vector< BasePDU * > &inDataPDUs)

27.202.1 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.

27.202.2 Member Function Documentation

27.202.2.1 static BasePDU* gdcm::network::PDUFactory::ConstructAbortPDU ( ) [static]

27.202.2.2 static BasePDU* gdcm::network::PDUFactory::ConstructPDU (uint8_t *itemtype*) [static]

27.202.2.3 static BasePDU* gdcm::network::PDUFactory::ConstructReleasePDU ( ) [static]

27.202.2.4 static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCEchoPDU (const ULConnection & *inConnection*) [static]

27.202.2.5 static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCFindPDU (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [static]

27.202.2.6 static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCMovePDU (const ULConnection & *inConnection,* const BaseRootQuery * *inRootQuery*) [static]

27.202.2.7 static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCStoreRQPDU (const ULConnection & *inConnection,* const File & *file*) [static]

27.202.2.8 static std::vector<BasePDU*> gdcm::network::PDUFactory::CreateCStoreRSPPDU (const DataSet * *inDataSet,* const BasePDU * *inPC*) [static]

27.202.2.9 static EEventID gdcm::network::PDUFactory::DetermineEventByPDU (const BasePDU * *inPDU*) [static]

27.202.2.10 static 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

27.203 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 = '^'

27.203.1 Detailed Description

PersonName class.

27.203.2 Member Function Documentation

27.203.2.1 unsigned int gdcm::PersonName::GetMaxLength ( ) const `[inline]`

27.203.2.2 unsigned int gdcm::PersonName::GetNumberOfComponents ( ) const `[inline]`

27.203.2.3 void gdcm::PersonName::Print (std::ostream & *os*) const `[inline]`

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

27.203.2.5 void gdcm::PersonName::SetComponents (const char * *comp1* = "", const char * *comp2* = "", const char * *comp3* = "", const char * *comp4* = "", const char * *comp5* = "") `[inline]`

27.203.2.6 void gdcm::PersonName::SetComponents (const char * *components[]*) `[inline]`

27.203.3 Member Data Documentation

27.203.3.1 char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]

27.203.3.2 const unsigned int gdcm::PersonName::MaxLength = 64 `[static]`

27.203.3.3 const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5 `[static]`

27.203.3.4 const char gdcm::PersonName::Padding = ' ' `[static]`

27.203.3.5 const char gdcm::PersonName::Separator = '^' `[static]`

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

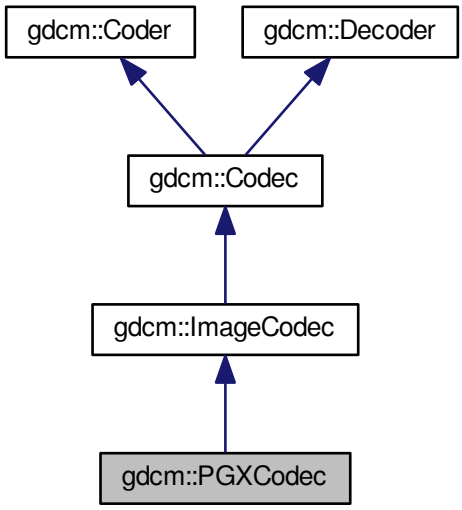
- gdcmPersonName.h

27.204 gdcm::PGXCodec Class Reference

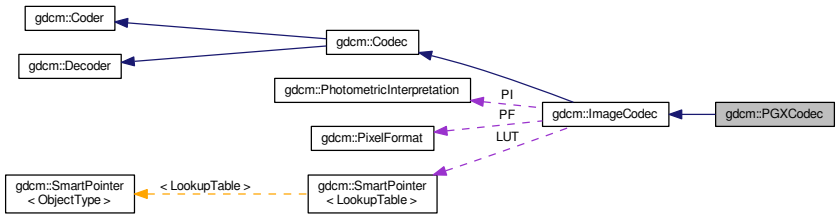
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

`#include <gdcmPGXCodec.h>`

Inheritance diagram for gdcm::PGXCodec:



Collaboration diagram for gdcm::PGXCodec:



Public Member Functions

- PGXCodec ()
- ~PGXCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool Read (const char *filename, DataElement &out) const
- bool Write (const char *filename, const DataElement &out) const

Additional Inherited Members

27.204.1 Detailed Description

Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

27.204.2 Constructor & Destructor Documentation

27.204.2.1 gdcm::PGXCodec::PGXCodec ( )

27.204.2.2 gdcm::PGXCodec::~PGXCodec ( )

27.204.3 Member Function Documentation

27.204.3.1 bool gdcm::PGXCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.204.3.2 bool gdcm::PGXCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.204.3.3 virtual ImageCodec* gdcm::PGXCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.204.3.4 bool gdcm::PGXCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

27.204.3.5 bool gdcm::PGXCodec::Read (const char * *filename*, DataElement & *out*) const

27.204.3.6 bool gdcm::PGXCodec::Write (const char * *filename*, const DataElement & *out*) const

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

- gdcmPGXCodec.h

27.205 gdcm::PhotometricInterpretation Class Reference

Class to represent an PhotometricInterpretation.

`#include <gdcmPhotometricInterpretation.h>`

Public Types

- enum PIType {
  UNKNOW = 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=UNKNOW)
- 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)

27.205.1 Detailed Description

Class to represent an PhotometricInterpretation.

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, csa2img.cxx, HelloVizWorld.cxx, and iU22tomultisc.cxx.

27.205.2 Member Enumeration Documentation

27.205.2.1 enum gdcm::PhotometricInterpretation::PIType

Enumerator

UNKNOW

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

27.205.3 Constructor & Destructor Documentation

27.205.3.1 gdcm::PhotometricInterpretation::PhotometricInterpretation (PIType *pi* = UNKNOW) [inline]

27.205.4 Member Function Documentation

27.205.4.1 static const char* gdcm::PhotometricInterpretation::GetPIString (PIType *pi*) [static]

Referenced by gdcm::operator<<().

27.205.4.2 static PIType gdcm::PhotometricInterpretation::GetPIType (const char * *pi*) [static]

27.205.4.3 unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel ( ) const

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

27.205.4.4 const char* gdcm::PhotometricInterpretation::GetString ( ) const

27.205.4.5 PIType gdcm::PhotometricInterpretation::GetType ( ) const [inline]

27.205.4.6 bool gdcm::PhotometricInterpretation::IsLossless ( ) const

27.205.4.7 bool gdcm::PhotometricInterpretation::IsLossy ( ) const

27.205.4.8 static bool gdcm::PhotometricInterpretation::IsRetired (PIType *pi*) [static]

27.205.4.9 bool gdcm::PhotometricInterpretation::IsSameColorSpace (PhotometricInterpretation const & *pi*) const

27.205.4.10 gdcm::PhotometricInterpretation::operator PIType ( ) const [inline]

27.205.5 Friends And Related Function Documentation

27.205.5.1 std::ostream& operator<< (std::ostream & *os,* const PhotometricInterpretation & *pi*) [friend]

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

- gdcmPhotometricInterpretation.h

27.206 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 (unsigned short samplesperpixel=1, unsigned short bitsallocated=8, unsigned short bitsstored=8, unsigned short highbit=7, unsigned short pixelrepresentation=0)
- PixelFormat (ScalarType st)
- 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!= (ScalarType st) const
- bool operator!= (const PixelFormat &pf) const
- bool operator== (ScalarType st) const
- bool operator== (const PixelFormat &pf) 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)

27.206.1 Detailed Description

PixelFormat.

Note

By default the Pixel Type will be instanciated 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, csa2img.cxx, FixJAIBugJPEGLS.cxx, GetJPEGSample↩ Precision.cxx, iU22tomultisc.cxx, and threadgdcm.cxx.

27.206.2 Member Enumeration Documentation

27.206.2.1 enum gdcm::PixelFormat::ScalarType

Enumerator

UINT8

INT8

UINT12

INT12

UINT16

INT16

UINT32

INT32

UINT64

INT64

FLOAT16

FLOAT32

FLOAT64

SINGLEBIT

UNKNOWN

27.206.3 Constructor & Destructor Documentation

27.206.3.1 gdcm::PixelFormat::PixelFormat (unsigned short *samplesperpixel* = `1`, unsigned short *bitsallocated* = `8`, unsigned short *bitsstored* = `8`, unsigned short *highbit* = `7`, unsigned short *pixelrepresentation* = `0`) `[inline],[explicit]`

27.206.3.2 gdcm::PixelFormat::PixelFormat (ScalarType *st*)

27.206.4 Member Function Documentation

27.206.4.1 unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const `[inline]`

BitsAllocated see Tag (0028,0100) US Bits Allocated.

Examples:

GetJPEGSamplePrecision.cxx.

27.206.4.2 unsigned short gdcm::PixelFormat::GetBitsStored ( ) const [inline]

BitsStored see Tag (0028,0101) US Bits Stored.

Examples:

GetJPEGSamplePrecision.cxx.

27.206.4.3 unsigned short gdcm::PixelFormat::GetHighBit ( ) const [inline]

HighBit see Tag (0028,0102) US High Bit.

27.206.4.4 int64_t gdcm::PixelFormat::GetMax ( ) const

return the max possible of the pixel

27.206.4.5 int64_t gdcm::PixelFormat::GetMin ( ) const

return the min possible of the pixel

27.206.4.6 unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const [inline]

PixelRepresentation: 0 or 1, see Tag (0028,0103) US Pixel Representation.

27.206.4.7 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:

threadgdcm.cxx.

27.206.4.8 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.

27.206.4.9 ScalarType gdcm::PixelFormat::GetScalarType ( ) const

ScalarType does not take into account the sample per pixel.

27.206.4.10 const char* gdcm::PixelFormat::GetScalarTypeAsString ( ) const

27.206.4.11 bool gdcm::PixelFormat::IsCompatible (const TransferSyntax & *ts*) const

27.206.4.12 bool gdcm::PixelFormat::IsValid ( ) const

return IsValid

27.206.4.13 gdcm::PixelFormat::operator ScalarType ( ) const [inline]

27.206.4.14 bool gdcm::PixelFormat::operator!= (ScalarType *st*) const [inline]

27.206.4.15 bool gdcm::PixelFormat::operator!= (const PixelFormat & *pf*) const [inline]

27.206.4.16 bool gdcm::PixelFormat::operator== (ScalarType *st*) const [inline]

27.206.4.17 bool gdcm::PixelFormat::operator== (const PixelFormat & *pf*) const [inline]

27.206.4.18 void gdcm::PixelFormat::Print (std::ostream & *os*) const

Print.

Referenced by gdcm::operator<<().

27.206.4.19 void gdcm::PixelFormat::SetBitsAllocated (unsigned short *ba*) [inline]

27.206.4.20 void gdcm::PixelFormat::SetBitsStored (unsigned short *bs*) [inline]

27.206.4.21 void gdcm::PixelFormat::SetHighBit (unsigned short *hb*) [inline]

27.206.4.22 void gdcm::PixelFormat::SetPixelRepresentation (unsigned short *pr*) [inline]

27.206.4.23 void gdcm::PixelFormat::SetSamplesPerPixel (unsigned short *spp*) [inline]

Examples:

CreateARGBImage.cxx, CreateCMYKImage.cxx, and GenFakeImage.cxx.

References gdcmAssertMacro.

27.206.4.24 void gdcm::PixelFormat::SetScalarType (ScalarType *st*)

Set PixelFormat based only on the ScalarType

Warning

: You need to call SetScalarType *before* SetSamplesPerPixel

27.206.4.25 bool gdcm::PixelFormat::Validate ( ) [protected]

When image with 24/24/23 was read, need to validate.

Referenced by gdcm::Bitmap::SetPixelFormat().

27.206.5 Friends And Related Function Documentation

27.206.5.1 friend class Bitmap [friend]

27.206.5.2 std::ostream& operator<< (std::ostream & *_os*, const PixelFormat & *pf*) [friend]

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

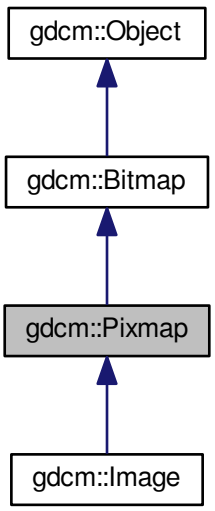
- gdcmPixelFormat.h

27.207 gdcm::Pixmap Class Reference

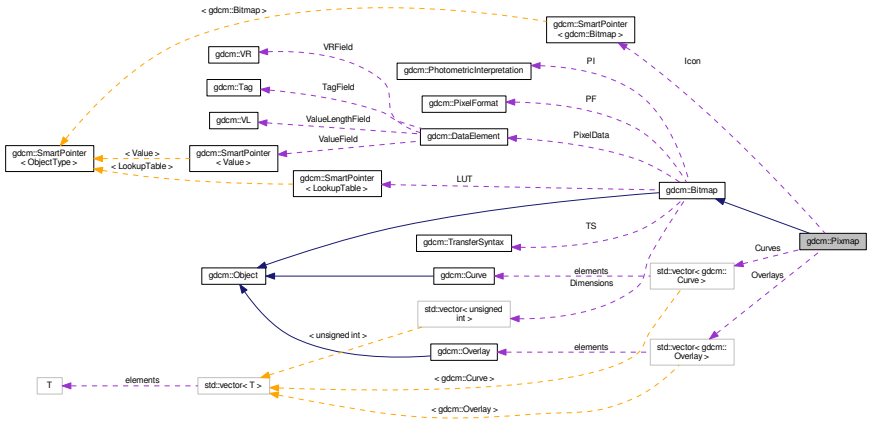
Pixmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data Image It does not contains any World Space information (IPP, IOP)

`#include <gdcmPixmap.h>`

Inheritance diagram for gdcm::Pixmap:



Collaboration diagram for gdcm::Pixmap:



Public Member Functions

- Pixmap ()
- ~Pixmap ()
- bool AreOverlaysInPixelData () const

  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
- const IconImage & GetIconImage () const

  Set/Get Icon Image.
- IconImage & GetIconImage ()
- 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
- void RemoveOverlay (size_t i)
- void SetIconImage (IconImage const &ii)
- void SetNumberOfCurves (size_t n)
- void SetNumberOfOverlays (size_t n)

Protected Attributes

- std::vector< Curve > Curves
- SmartPointer< IconImage > Icon
- std::vector< Overlay > Overlays

Additional Inherited Members

27.207.1 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

27.207.2 Constructor & Destructor Documentation

27.207.2.1 gdcm::Pixmap::Pixmap ( )

27.207.2.2 gdcm::Pixmap::~Pixmap ( )

27.207.3 Member Function Documentation

27.207.3.1 bool gdcm::Pixmap::AreOverlaysInPixelData ( ) const [virtual]

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from gdcm::Bitmap.

27.207.3.2 Curve& gdcm::Pixmap::GetCurve (size_t *i* = 0) [inline]

Curve: group 50xx.

27.207.3.3 const Curve& gdcm::Pixmap::GetCurve (size_t *i* = 0) const [inline]

27.207.3.4 const IconImage& gdcm::Pixmap::GetIconImage ( ) const [inline]

Set/Get Icon Image.

27.207.3.5 IconImage& gdcm::Pixmap::GetIconImage ( ) [inline]

27.207.3.6 size_t gdcm::Pixmap::GetNumberOfCurves ( ) const [inline]

27.207.3.7 size_t gdcm::Pixmap::GetNumberOfOverlays ( ) const [inline]

27.207.3.8 Overlay& gdcm::Pixmap::GetOverlay (size_t *i* = 0) [inline]

Overlay: group 60xx.

27.207.3.9 const Overlay& gdcm::Pixmap::GetOverlay (size_t *i* = 0) const [inline]

27.207.3.10 void gdcm::Pixmap::Print (std::ostream &) const [virtual]

Reimplemented from gdcm::Bitmap.

27.207.3.11 void gdcm::Pixmap::RemoveOverlay (size_t *i*) [inline]

27.207.3.12 void gdcm::Pixmap::SetIconImage (IconImage const & *ii*) [inline]

27.207.3.13 void gdcm::Pixmap::SetNumberOfCurves (size_t *n*) [inline]

27.207.3.14 void gdcm::Pixmap::SetNumberOfOverlays (size_t *n*) [inline]

27.207.4 Member Data Documentation

27.207.4.1 std::vector<Curve> gdcm::Pixmap::Curves [protected]

27.207.4.2 SmartPointer<IconImage> gdcm::Pixmap::Icon [protected]

27.207.4.3 std::vector<Overlay> gdcm::Pixmap::Overlays [protected]

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

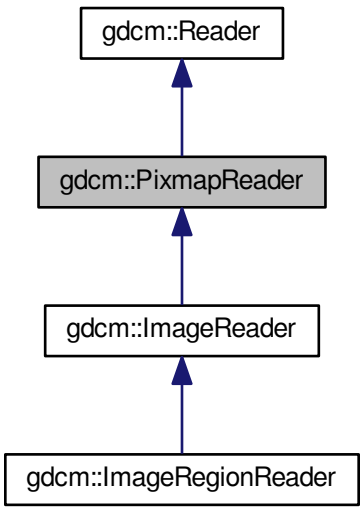
- gdcmPixmap.h

27.208 gdcm::PixmapReader Class Reference

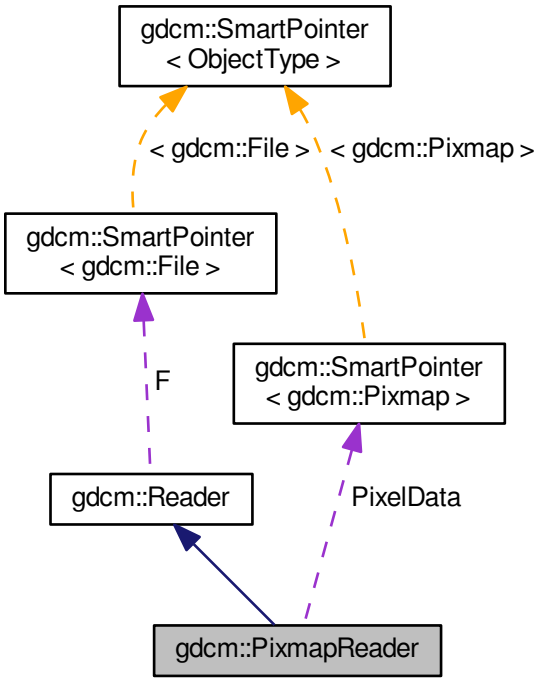
PixmapReader.

`#include <gdcmPixmapReader.h>`

Inheritance diagram for gdcm::PixmapReader:



Collaboration diagram for gdcm::PixmapReader:



Public Member Functions

- PixmapReader ()
- virtual ~PixmapReader ()
- const Pixmap & GetPixmap () const

  Return the read image (need to call Read() first)
- Pixmap & GetPixmap ()
- virtual bool Read ()

Protected Member Functions

- virtual bool ReadACRNEMAImage ()
- virtual bool ReadImage (MediaStorage const &ms)
- bool ReadImageInternal (MediaStorage const &ms, bool handlepixeldata=true)

Protected Attributes

- SmartPointer< Pixmap > PixelData

27.208.1 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

27.208.2 Constructor & Destructor Documentation

27.208.2.1 gdcm::PixmapReader::PixmapReader ( )

27.208.2.2 virtual gdcm::PixmapReader::~PixmapReader ( ) [virtual]

27.208.3 Member Function Documentation

27.208.3.1 const Pixmap& gdcm::PixmapReader::GetPixmap ( ) const

Return the read image (need to call Read() first)

27.208.3.2 Pixmap& gdcm::PixmapReader::GetPixmap ( )

27.208.3.3 virtual bool gdcm::PixmapReader::Read ( ) [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.

Reimplemented in gdcm::ImageRegionReader, and gdcm::ImageReader.

27.208.3.4 virtual bool gdcm::PixmapReader::ReadACRNEMAImage ( ) [protected], [virtual]

Reimplemented in gdcm::ImageReader.

27.208.3.5 virtual bool gdcm::PixmapReader::ReadImage (MediaStorage const & *ms*) [protected], [virtual]

Reimplemented in gdcm::ImageReader.

27.208.3.6 bool gdcm::PixmapReader::ReadImageInternal (MediaStorage const & *ms*, bool *handlepixeldata* = `true`) [protected]

27.208.4 Member Data Documentation

27.208.4.1 SmartPointer<Pixmap> gdcm::PixmapReader::PixelData [protected]

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

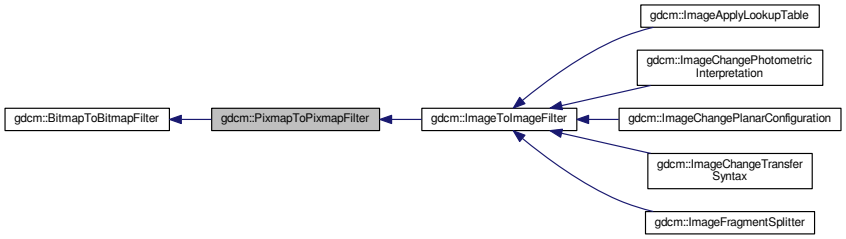
- gdcmPixmapReader.h

27.209 gdcm::PixmapToPixmapFilter Class Reference

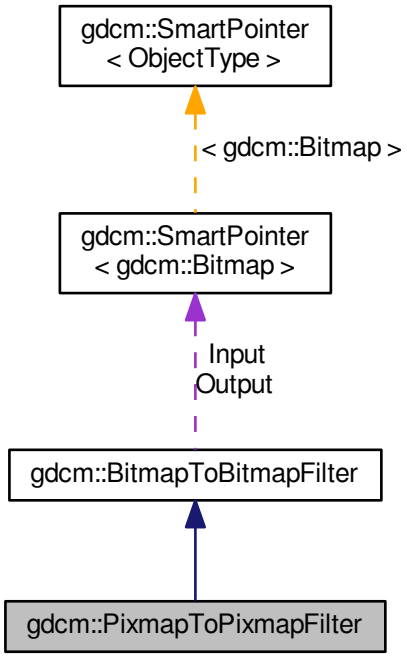
PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.

`#include <gdcmPixmapToPixmapFilter.h>`

Inheritance diagram for gdcm::PixmapToPixmapFilter:



Collaboration diagram for gdcm::PixmapToPixmapFilter:



Public Member Functions

- PixmapToPixmapFilter ()
- ~PixmapToPixmapFilter ()
- Pixmap & GetInput ()
- const Pixmap & GetOutput () const

  Get Output image.
- const Pixmap & GetOutputAsPixmap () const

Additional Inherited Members

27.209.1 Detailed Description

PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.

27.209.2 Constructor & Destructor Documentation

27.209.2.1 gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )

27.209.2.2 gdcm::PixmapToPixmapFilter::~PixmapToPixmapFilter ( ) [inline]

27.209.3 Member Function Documentation

27.209.3.1 Pixmap& gdcm::PixmapToPixmapFilter::GetInput ( )

27.209.3.2 const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput ( ) const

Get Output image.

27.209.3.3 const Pixmap& gdcm::PixmapToPixmapFilter::GetOutputAsPixmap ( ) const

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

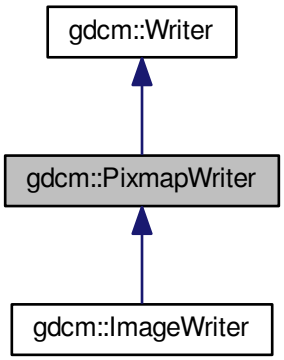
- gdcmPixmapToPixmapFilter.h

27.210 gdcm::PixmapWriter Class Reference

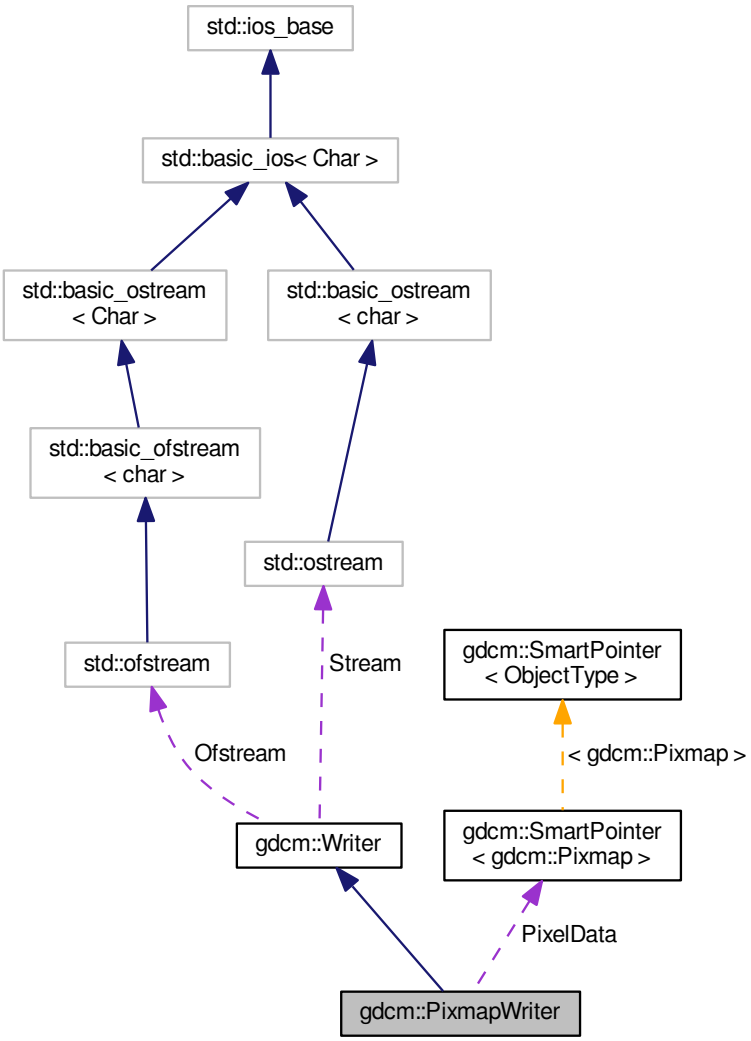
PixmapWriter This class will takes two inputs:

```
#include <gdcmPixmapWriter.h>
```

Inheritance diagram for gdcm::PixmapWriter:



Collaboration diagram for gdcm::PixmapWriter:



Public Member Functions

- PixmapWriter ()
- ~PixmapWriter ()
- virtual const Pixmap & GetImage () const
- virtual Pixmap & GetImage ()
- const Pixmap & GetPixmap () const
- Pixmap & GetPixmap ()
- virtual void SetImage (Pixmap const &img)
- void SetPixmap (Pixmap const &img)
- bool Write ()

Write.

Protected Member Functions

- void DoIconImage (DataSet &ds, Pixmap const &image)
- bool PrepareWrite ()

Protected Attributes

- SmartPointer< Pixmap > PixelData

27.210.1 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 implicitely lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

27.210.2 Constructor & Destructor Documentation

27.210.2.1 gdcm::PixmapWriter::PixmapWriter ( )

27.210.2.2 gdcm::PixmapWriter::~PixmapWriter ( )

27.210.3 Member Function Documentation

27.210.3.1 void gdcm::PixmapWriter::DoIconImage (DataSet & *ds*, Pixmap const & *image*) [protected]

27.210.3.2 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.

27.210.3.3 virtual Pixmap& gdcm::PixmapWriter::GetImage ( ) [inline], [virtual]

Reimplemented in gdcm::ImageWriter.

27.210.3.4 const Pixmap& gdcm::PixmapWriter::GetPixmap ( ) const [inline]

27.210.3.5 Pixmap& gdcm::PixmapWriter::GetPixmap ( ) [inline]

27.210.3.6 bool gdcm::PixmapWriter::PrepareWrite ( ) [protected]

27.210.3.7 virtual void gdcm::PixmapWriter::SetImage (Pixmap const & *img*) [virtual]

Examples:

CompressImage.cxx, GenFakeImage.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, and MergeTwoFiles.cxx.

27.210.3.8 void gdcm::PixmapWriter::SetPixmap (Pixmap const & *img*)

27.210.3.9 bool gdcm::PixmapWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::Writer.

27.210.4 Member Data Documentation

27.210.4.1 SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData [protected]

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

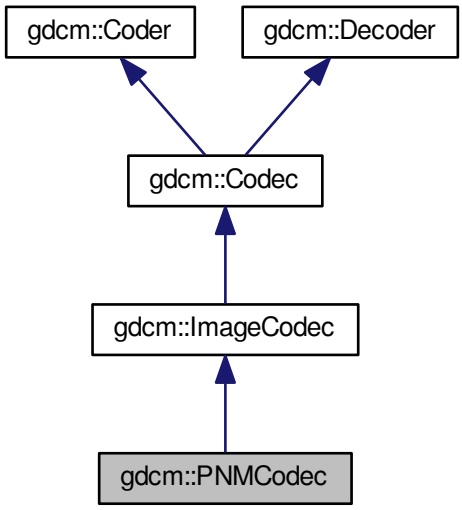
- gdcmPixmapWriter.h

27.211 gdcm::PNMCodec Class Reference

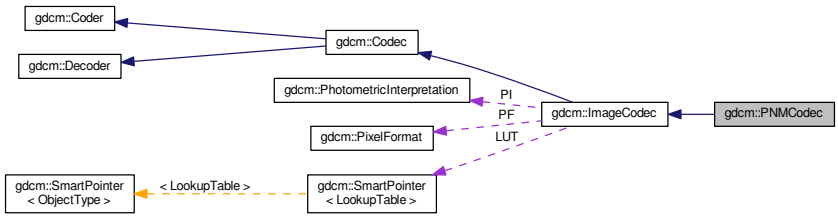
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: http://netpbm.↩ sourceforge.net/.

`#include <gdcmPNMCodec.h>`

Inheritance diagram for gdcm::PNMCodec:



Collaboration diagram for gdcm::PNMCodec:



Public Member Functions

- PNMCodec ()
- ~PNMCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- bool Read (const char *filename, DataElement &out) const

- void SetBufferLength (unsigned long l)
- bool Write (const char *filename, const DataElement &out) const

Additional Inherited Members

27.211.1 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.

27.211.2 Constructor & Destructor Documentation

27.211.2.1 gdcm::PNMCodec::PNMCodec ( )

27.211.2.2 gdcm::PNMCodec::~PNMCodec ( )

27.211.3 Member Function Documentation

27.211.3.1 bool gdcm::PNMCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.211.3.2 bool gdcm::PNMCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.211.3.3 virtual ImageCodec* gdcm::PNMCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.211.3.4 unsigned long gdcm::PNMCodec::GetBufferLength ( ) const [inline]

27.211.3.5 bool gdcm::PNMCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

27.211.3.6 bool gdcm::PNMCodec::Read (const char * *filename,* DataElement & *out*) const

27.211.3.7 void gdcm::PNMCodec::SetBufferLength (unsigned long *l*) [inline]

27.211.3.8 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

27.212 gdcm::Preamble Class Reference

DICOM Preamble (Part 10)

`#include <gdcmPreamble.h>`

Public Member Functions

- Preamble ()
- Preamble (Preamble const &)
- ~Preamble ()
- void Clear ()
- void Create ()
- const char * GetInternal () const
- VL GetLength () const
- bool IsEmpty () const
- Preamble & operator= (Preamble const &)
- void Print (std::ostream &os) const
- std::istream & Read (std::istream &is)
- void Remove ()
- void Valid ()
- std::ostream const & Write (std::ostream &os) const

Protected Member Functions

- bool IsValid () const

Friends

- std::ostream & operator<< (std::ostream &_os, const Preamble &_val)

27.212.1 Detailed Description

DICOM Preamble (Part 10)

27.212.2 Constructor & Destructor Documentation

27.212.2.1 gdcm::Preamble::Preamble ( )

27.212.2.2 gdcm::Preamble::~Preamble ( )

27.212.2.3 gdcm::Preamble::Preamble (Preamble const &) [inline]

27.212.3 Member Function Documentation

27.212.3.1 void gdcm::Preamble::Clear ( )

27.212.3.2 void gdcm::Preamble::Create ( )

27.212.3.3 const char* gdcm::Preamble::GetInternal ( ) const [inline]

27.212.3.4 VL gdcm::Preamble::GetLength ( ) const [inline]

27.212.3.5 bool gdcm::Preamble::IsEmpty ( ) const [inline]

27.212.3.6 bool gdcm::Preamble::IsValid ( ) const [inline], [protected]

27.212.3.7 Preamble& gdcm::Preamble::operator= (Preamble const &) [inline]

27.212.3.8 void gdcm::Preamble::Print (std::ostream & *os*) const

27.212.3.9 std::istream& gdcm::Preamble::Read (std::istream & *is*)

27.212.3.10 void gdcm::Preamble::Remove ( )

27.212.3.11 void gdcm::Preamble::Valid ( )

27.212.3.12 std::ostream const& gdcm::Preamble::Write (std::ostream & *os*) const

27.212.4 Friends And Related Function Documentation

27.212.4.1 std::ostream& operator<< (std::ostream & *_os,* const Preamble & *_val*) [friend]

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

- gdcmPreamble.h

27.213 gdcm::PresentationContext Class Reference

PresentationContext.

```
#include <gdcmPresentationContext.h>
```

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::ImplicitVRLittleEndianDefault↩ TransferSyntaxforDICOM)
- 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 *as)
- void SetPresentationContextID (uint8_t id)

27.213.1 Detailed Description

PresentationContext.

See also

PresentationContextAC PresentationContextRQ

27.213.2 Member Typedef Documentation

27.213.2.1 typedef TransferSyntaxArrayType::size_type gdcm::PresentationContext::SizeType

27.213.2.2 typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType

27.213.3 Constructor & Destructor Documentation

27.213.3.1 gdcm::PresentationContext::PresentationContext ( )

27.213.3.2 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).

27.213.4 Member Function Documentation

27.213.4.1 void gdcm::PresentationContext::AddTransferSyntax (const char * *tsstr*)

27.213.4.2 const char* gdcm::PresentationContext::GetAbstractSyntax ( ) const [inline]

27.213.4.3 SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const [inline]

27.213.4.4 uint8_t gdcm::PresentationContext::GetPresentationContextID ( ) const

27.213.4.5 const char* gdcm::PresentationContext::GetTransferSyntax (SizeType *i*) const [inline]

27.213.4.6 bool gdcm::PresentationContext::operator== (const PresentationContext & *pc*) const [inline]

27.213.4.7 void gdcm::PresentationContext::Print (std::ostream & *os*) const

27.213.4.8 void gdcm::PresentationContext::SetAbstractSyntax (const char * *as*) [inline]

27.213.4.9 void gdcm::PresentationContext::SetPresentationContextID (uint8_t *id*)

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

- gdcmPresentationContext.h

27.214 gdcm::network::PresentationContextAC Class Reference

PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

`#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

27.214.1 Detailed Description

PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

See also

PresentationContext

27.214.2 Constructor & Destructor Documentation

27.214.2.1 gdcm::network::PresentationContextAC::PresentationContextAC ( )

27.214.3 Member Function Documentation

27.214.3.1 uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID ( ) const [inline]

27.214.3.2 uint8_t gdcm::network::PresentationContextAC::GetReason ( ) const [inline]

27.214.3.3 TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax ( ) const [inline]

27.214.3.4 void gdcm::network::PresentationContextAC::Print (std::ostream & *os*) const

27.214.3.5 std::istream& gdcm::network::PresentationContextAC::Read (std::istream & *is*)

27.214.3.6 void gdcm::network::PresentationContextAC::SetPresentationContextID (uint8_t *id*)

27.214.3.7 void gdcm::network::PresentationContextAC::SetReason (uint8_t *r*) [inline]

27.214.3.8 void gdcm::network::PresentationContextAC::SetTransferSyntax (TransferSyntaxSub const & *ts*)

27.214.3.9 size_t gdcm::network::PresentationContextAC::Size ( ) const

27.214.3.10 const std::ostream& gdcm::network::PresentationContextAC::Write (std::ostream & *os*) const

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

- gdcmPresentationContextAC.h

27.215 gdcm::PresentationContextGenerator Class Reference

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.

`#include <gdcmPresentationContextGenerator.h>`

Public Types

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

Public Member Functions

- PresentationContextGenerator ()
- 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 *as, const char *ts)
- const char * GetDefaultTransferSyntax () const

27.215.1 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.

27.215.2 Member Typedef Documentation

27.215.2.1 typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType

27.215.2.2 typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType

27.215.3 Constructor & Destructor Documentation

27.215.3.1 gdcm::PresentationContextGenerator::PresentationContextGenerator ( )

27.215.4 Member Function Documentation

27.215.4.1 bool gdcm::PresentationContextGenerator::AddPresentationContext (const char * *as*, const char * *ts*) [protected]

27.215.4.2 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.

27.215.4.3 bool gdcm::PresentationContextGenerator::GenerateFromUID (UIDs::TSName *asname*)

Generate the PresentationContext array from a UID (eg. VerificationSOPClass)

27.215.4.4 const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const [protected]

27.215.4.5 PresentationContextArrayType const& gdcm::PresentationContextGenerator::GetPresentationContexts ( ) [inline]

Examples:

CStoreQtProgress.cxx.

27.215.4.6 void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax (const TransferSyntax & *ts*)

Not implemented for now. GDCM internally uses Implicit Little Endian.

27.215.4.7 void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )

27.215.4.8 void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ( )

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

- gdcmPresentationContextGenerator.h

27.216 gdcm::network::PresentationContextRQ Class Reference

PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

`#include <gdcmPresentationContextRQ.h>`

Public Types

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

Public Member Functions

- PresentationContextRQ ()
- PresentationContextRQ (UIDs::TSName asname, UIDs::TSName tsname=UIDs::ImplicitVRLittleEndianDefault↩ TransferSyntaxforDICOM)
- PresentationContextRQ (const PresentationContext &pc)
- void AddTransferSyntax (TransferSyntaxSub const &ts)
- AbstractSyntax const & GetAbstractSyntax () const
- AbstractSyntax & GetAbstractSyntax ()
- SizeType GetNumberOfTransferSyntaxes () const
- uint8_t GetPresentationContextID () const
- TransferSyntaxSub const & GetTransferSyntax (SizeType i) const
- TransferSyntaxSub & GetTransferSyntax (SizeType i)
- 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 &as)
- void SetPresentationContextID (uint8_t id)
- size_t Size () const
- const std::ostream & Write (std::ostream &os) const

27.216.1 Detailed Description

PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

See also

PresentationContextAC

27.216.2 Member Typedef Documentation

27.216.2.1 typedef std::vector<TransferSyntaxSub>::size_type gdcm::network::PresentationContextRQ::SizeType

27.216.3 Constructor & Destructor Documentation

27.216.3.1 gdcm::network::PresentationContextRQ::PresentationContextRQ ( )

27.216.3.2 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).

27.216.3.3 gdcm::network::PresentationContextRQ::PresentationContextRQ (const PresentationContext & *pc*)

27.216.4 Member Function Documentation

27.216.4.1 void gdcm::network::PresentationContextRQ::AddTransferSyntax (TransferSyntaxSub const & *ts*)

27.216.4.2 **AbstractSyntax** const& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) const [inline]

27.216.4.3 **AbstractSyntax**& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) [inline]

27.216.4.4 **SizeType** gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const [inline]

27.216.4.5 uint8_t gdcm::network::PresentationContextRQ::GetPresentationContextID ( ) const

27.216.4.6 **TransferSyntaxSub** const& gdcm::network::PresentationContextRQ::GetTransferSyntax (**SizeType** *i*) const [inline]

27.216.4.7 **TransferSyntaxSub**& gdcm::network::PresentationContextRQ::GetTransferSyntax (**SizeType** *i*) [inline]

27.216.4.8 std::vector<**TransferSyntaxSub**> const& gdcm::network::PresentationContextRQ::GetTransferSyntaxes ( ) const [inline]

27.216.4.9 bool gdcm::network::PresentationContextRQ::operator== (const **PresentationContextRQ** & *pc*) const [inline]

27.216.4.10 void gdcm::network::PresentationContextRQ::Print (std::ostream & *os*) const

27.216.4.11 std::istream& gdcm::network::PresentationContextRQ::Read (std::istream & *is*)

27.216.4.12 void gdcm::network::PresentationContextRQ::SetAbstractSyntax (**AbstractSyntax** const & *as*)

27.216.4.13 void gdcm::network::PresentationContextRQ::SetPresentationContextID (uint8_t *id*)

27.216.4.14 size_t gdcm::network::PresentationContextRQ::Size ( ) const

27.216.4.15 const std::ostream& gdcm::network::PresentationContextRQ::Write (std::ostream & *os*) const

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

- gdcmPresentationContextRQ.h

27.217 gdcm::network::PresentationDataValue Class Reference

PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

`#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)

27.217.1 Detailed Description

PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

27.217.2 Constructor & Destructor Documentation

27.217.2.1 gdcm::network::PresentationDataValue::PresentationDataValue ( )

27.217.3 Member Function Documentation

27.217.3.1 static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (const std::vector< PresentationDataValue > & *inPDVs*) [static]

Warning

DataSet will be read as Implicit Little Endian TS

27.217.3.2 static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit (const std::vector< PresentationDataValue > & *inPDVs*) [static]

27.217.3.3 const std::string& gdcm::network::PresentationDataValue::GetBlob ( ) const

27.217.3.4 bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const

27.217.3.5 bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const

27.217.3.6 uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const [inline]

27.217.3.7 uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const [inline]

27.217.3.8 void gdcm::network::PresentationDataValue::Print (std::ostream & *os*) const

27.217.3.9 std::istream& gdcm::network::PresentationDataValue::Read (std::istream & *is*)

27.217.3.10 std::istream& gdcm::network::PresentationDataValue::ReadInto (std::istream & *is*, std::ostream & *os*)

27.217.3.11 void gdcm::network::PresentationDataValue::SetBlob (const std::string & *partialblob*)

27.217.3.12 void gdcm::network::PresentationDataValue::SetCommand (bool *inCommand*)

27.217.3.13 void gdcm::network::PresentationDataValue::SetDataSet (const DataSet & *ds*)

Set DataSet. Write DataSet in implicit.

Warning

size of dataset should be below maxpdusize

27.217.3.14 void gdcm::network::PresentationDataValue::SetLastFragment (bool *inLast*)

27.217.3.15 void gdcm::network::PresentationDataValue::SetMessageHeader (uint8_t *messageheader*) [inline]

27.217.3.16 void gdcm::network::PresentationDataValue::SetPresentationContextID (uint8_t *id*) [inline]

27.217.3.17 size_t gdcm::network::PresentationDataValue::Size ( ) const

27.217.3.18 const std::ostream& gdcm::network::PresentationDataValue::Write (std::ostream & *os*) const

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

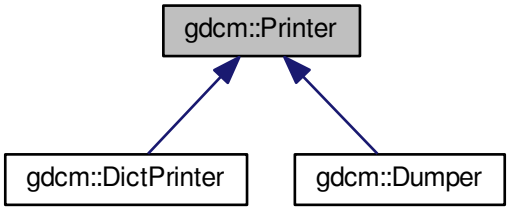
- gdcmPresentationDataValue.h

27.218 gdcm::Printer Class Reference

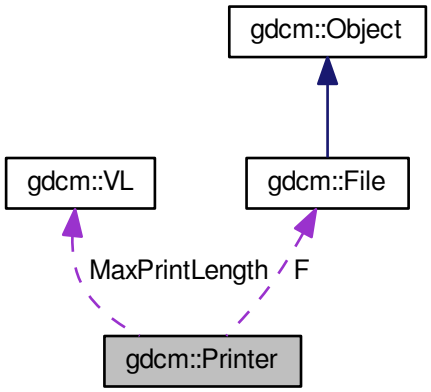
Printer class.

`#include <gdcmPrinter.h>`

Inheritance diagram for gdcm::Printer:



Collaboration diagram for gdcm::Printer:



Public Types

- enum PrintStyles {
  VERBOSE_STYLE = 0,
  CONDENSED_STYLE,
  XML }

Public Member Functions

- Printer ()
- ~Printer ()
- 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

27.218.1 Detailed Description

Printer class.

27.218.2 Member Enumeration Documentation

27.218.2.1 enum gdcm::Printer::PrintStyles

Enumerator

VERBOSE_STYLE

CONDENSED_STYLE

XML

27.218.3 Constructor & Destructor Documentation

27.218.3.1 gdcm::Printer::Printer ( )

27.218.3.2 gdcm::Printer::~Printer ( )

27.218.4 Member Function Documentation

27.218.4.1 PrintStyles gdcm::Printer::GetPrintStyle ( ) const [inline]

Get PrintStyle value.

27.218.4.2 void gdcm::Printer::Print (std::ostream & *os*)

Print.

27.218.4.3 VR gdcm::Printer::PrintDataElement (std::ostringstream & *os*, const Dicts & *dicts*, const DataSet & *ds*, const DataElement & *de*, std::ostream & *out*, std::string const & *indent*) [protected]

27.218.4.4 void gdcm::Printer::PrintDataSet (const DataSet & *ds*, std::ostream & *os*, const std::string & *s* = " ")

Print an individual dataset.

27.218.4.5 void gdcm::Printer::PrintSQ (const SequenceOfItems * *sqi,* std::ostream & *os,* std::string const & *indent*) [protected]

27.218.4.6 void gdcm::Printer::SetColor (bool *c*)

Set color mode or not.

27.218.4.7 void gdcm::Printer::SetFile (File const & *f*) [inline]

Set file.

27.218.4.8 void gdcm::Printer::SetStyle (PrintStyles *ps*) [inline]

Set PrintStyle value.

27.218.5 Member Data Documentation

27.218.5.1 const File* gdcm::Printer::F [protected]

27.218.5.2 VL gdcm::Printer::MaxPrintLength [protected]

27.218.5.3 PrintStyles gdcm::Printer::PrintStyle [protected]

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

- gdcmPrinter.h

27.219 gdcm::PrivateDict Class Reference

Private Dict.

`#include <gdcmDict.h>`

Public Member Functions

- PrivateDict ()
- ~PrivateDict ()
- 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)

27.219.1 Detailed Description

Private Dict.

27.219.2 Constructor & Destructor Documentation

27.219.2.1 gdcm::PrivateDict::PrivateDict ( ) [inline]

27.219.2.2 gdcm::PrivateDict::~PrivateDict ( ) [inline]

27.219.3 Member Function Documentation

27.219.3.1 void gdcm::PrivateDict::AddDictEntry (const PrivateTag & *tag,* const DictEntry & *de*) [inline]

References gdcm::DictEntry::GetVM(), gdcm::DictEntry::GetVR(), gdcm::DictEntry::SetVR(), and gdcm::VR::UN.

27.219.3.2 bool gdcm::PrivateDict::FindDictEntry (const PrivateTag & *tag*) const [inline]

27.219.3.3 const DictEntry& gdcm::PrivateDict::GetDictEntry (const PrivateTag & *tag*) const [inline]

27.219.3.4 bool gdcm::PrivateDict::IsEmpty ( ) const [inline]

27.219.3.5 void gdcm::PrivateDict::LoadDefault ( ) [protected]

27.219.3.6 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().

27.219.3.7 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.

27.219.4 Friends And Related Function Documentation

27.219.4.1 friend class Dicts [friend]

27.219.4.2 std::ostream& operator<< (std::ostream & *os,* const PrivateDict & *val*) [friend]

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

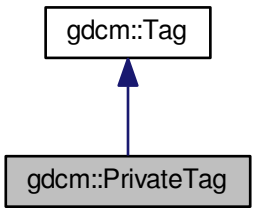
- gdcmDict.h

27.220 gdcm::PrivateTag Class Reference

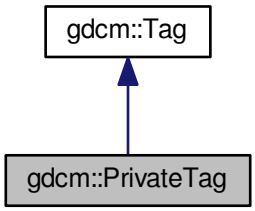
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element, Owner)

`#include <gdcmPrivateTag.h>`

Inheritance diagram for gdcm::PrivateTag:



Collaboration diagram for gdcm::PrivateTag:



Public Member Functions

- PrivateTag (uint16_t group=0, uint16_t element=0, const char *owner="")
- PrivateTag (Tag const &t, const char *owner="")
- DataElement GetAsDataElement () const
- const char * GetOwner () const
- bool operator< (const PrivateTag &_val) const
- bool ReadFromCommaSeparatedString (const char *str)
- void SetOwner (const char *owner)

Friends

- std::ostream & operator<< (std::ostream &_os, const PrivateTag &_val)

27.220.1 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, csa2img.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, ELSCINT1WaveToText.cxx, GetSubSequenceData.cxx, i↩ U22tomultisc.cxx, MrProtocol.cxx, pmsct_rgb1.cxx, PublicDict.cxx, ReadGEMSSDO.cxx, and rle2img.cxx.

27.220.2 Constructor & Destructor Documentation

27.220.2.1 gdcm::PrivateTag::PrivateTag (uint16_t *group* = 0, uint16_t *element* = 0, const char * *owner* = " ") [inline]

27.220.2.2 gdcm::PrivateTag::PrivateTag (Tag const & *t*, const char * *owner* = " ") [inline]

References gdcm::Tag::GetElement().

27.220.3 Member Function Documentation

27.220.3.1 DataElement gdcm::PrivateTag::GetAsDataElement ( ) const

27.220.3.2 const char* gdcm::PrivateTag::GetOwner ( ) const [inline]

Examples:

PublicDict.cxx.

Referenced by gdcm::PrivateDict::PrintXML().

27.220.3.3 bool gdcm::PrivateTag::operator< (const PrivateTag & *_val*) const

27.220.3.4 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")

27.220.3.5 void gdcm::PrivateTag::SetOwner (const char * *owner*) [inline]

27.220.4 Friends And Related Function Documentation

27.220.4.1 std::ostream& operator<< (std::ostream & *_os*, const PrivateTag & *_val*) [friend]

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

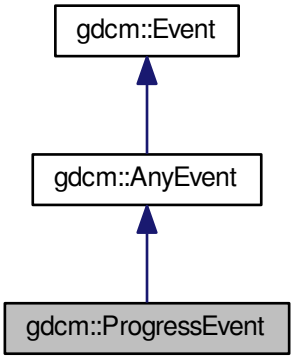
- gdcmPrivateTag.h

27.221 gdcm::ProgressEvent Class Reference

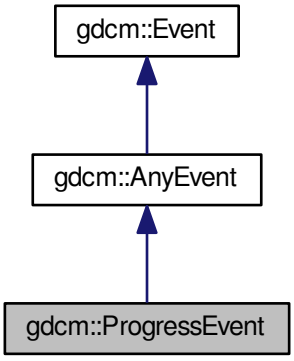
ProgressEvent Special type of event triggered during.

`#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 (double p=0)
- ProgressEvent (const Self &s)
- virtual ~ProgressEvent ()
- virtual bool CheckEvent (const ::gdcm::Event *e) const
- virtual const char * GetEventName () const
- double GetProgress () const
- virtual ::gdcm::Event * MakeObject () const
- void SetProgress (double p)

27.221.1 Detailed Description

ProgressEvent Special type of event triggered during.

See also

AnyEvent

27.221.2 Member Typedef Documentation

27.221.2.1 typedef ProgressEvent gdcm::ProgressEvent::Self

27.221.2.2 typedef AnyEvent gdcm::ProgressEvent::Superclass

27.221.3 Constructor & Destructor Documentation

27.221.3.1 gdcm::ProgressEvent::ProgressEvent (double *p* = 0) [inline]

27.221.3.2 virtual gdcm::ProgressEvent::~ProgressEvent ( ) [inline], [virtual]

27.221.3.3 gdcm::ProgressEvent::ProgressEvent (const Self & *s*) [inline]

27.221.4 Member Function Documentation

27.221.4.1 virtual bool gdcm::ProgressEvent::CheckEvent (const ::gdcm::Event * *e*) const [inline], [virtual]

27.221.4.2 virtual const char* gdcm::ProgressEvent::GetEventName ( ) const [inline], [virtual]

Return the StringName associated with the event.

Implements gdcm::Event.

27.221.4.3 double gdcm::ProgressEvent::GetProgress ( ) const [inline]

27.221.4.4 virtual ::gdcm::Event* gdcm::ProgressEvent::MakeObject ( ) const [inline], [virtual]

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

Implements gdcm::Event.

27.221.4.5 void gdcm::ProgressEvent::SetProgress (double *p*) [inline]

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

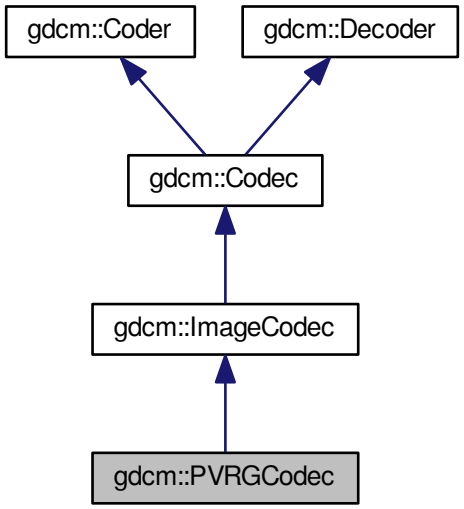
- gdcmProgressEvent.h

27.222 gdcm::PVRGCodec Class Reference

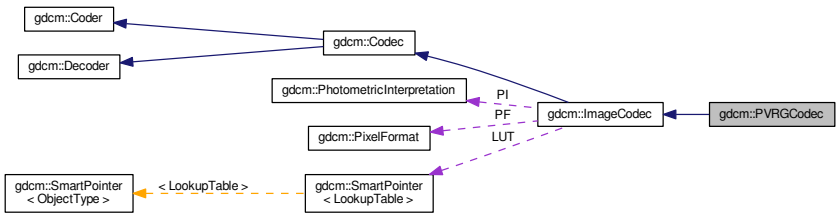
PVRGCodec.

`#include <gdcmPVRGCodec.h>`

Inheritance diagram for gdcm::PVRGCodec:



Collaboration diagram for gdcm::PVRGCodec:



Public Member Functions

- PVRGCodec ()
- ~PVRGCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- void SetLossyFlag (bool l)

Additional Inherited Members

27.222.1 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...

27.222.2 Constructor & Destructor Documentation

27.222.2.1 gdcm::PVRGCodec::PVRGCodec ( )

27.222.2.2 gdcm::PVRGCodec::~PVRGCodec ( )

27.222.3 Member Function Documentation

27.222.3.1 bool gdcm::PVRGCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.222.3.2 bool gdcm::PVRGCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.222.3.3 virtual ImageCodec* gdcm::PVRGCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.222.3.4 bool gdcm::PVRGCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

27.222.3.5 bool gdcm::PVRGCodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

27.222.3.6 void gdcm::PVRGCodec::SetLossyFlag (bool /)

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

- gdcmPVRGCodec.h

27.223 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 use)

27.223.1 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.

27.223.2 Constructor & Destructor Documentation

27.223.2.1 gdcm::PythonFilter::PythonFilter ( )

27.223.2.2 gdcm::PythonFilter::~PythonFilter ( )

27.223.3 Member Function Documentation

27.223.3.1 File& gdcm::PythonFilter::GetFile ( ) [inline]

27.223.3.2 const File& gdcm::PythonFilter::GetFile ( ) const [inline]

27.223.3.3 void gdcm::PythonFilter::SetDicts (const Dicts & *dicts*)

27.223.3.4 void gdcm::PythonFilter::SetFile (const File & *f*) [inline]

27.223.3.5 PyObject* gdcm::PythonFilter::ToPyObject (const Tag & *t*) const

27.223.3.6 void gdcm::PythonFilter::UseDictAlways (bool *use*) [inline]

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

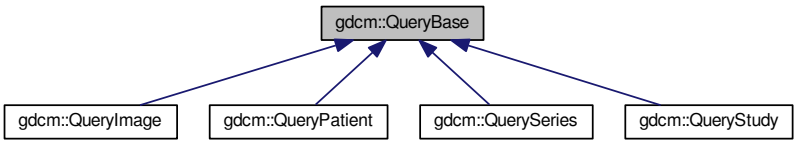
- gdcmPythonFilter.h

27.224 gdcm::QueryBase Class Reference

QueryBase contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

`#include <gdcmQueryBase.h>`

Inheritance diagram for gdcm::QueryBase:



Public Member Functions

- virtual ~QueryBase ()
- 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

27.224.1 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.

27.224.2 Constructor & Destructor Documentation

27.224.2.1 virtual gdcm::QueryBase::~QueryBase ( ) [inline], [virtual]

27.224.3 Member Function Documentation

27.224.3.1 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

27.224.3.2 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

27.224.3.3 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.

27.224.3.4 virtual const char* gdcm::QueryBase::GetName ( ) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

27.224.3.5 virtual std::vector<Tag> gdcm::QueryBase::GetOptionalTags (const ERootType & *inRootType*) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

27.224.3.6 virtual DataElement gdcm::QueryBase::GetQueryLevel ( ) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

27.224.3.7 virtual std::vector<Tag> gdcm::QueryBase::GetRequiredTags (const ERootType & *inRootType*) const [pure virtual]

Implemented in gdcm::QueryImage, gdcm::QueryPatient, gdcm::QuerySeries, and gdcm::QueryStudy.

27.224.3.8 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

27.225 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 BaseRootQuery * ProduceQuery (ERootType inRootType, EQueryType inQueryType, EQueryLevel in↩ QueryLevel)

27.225.1 Detailed Description

QueryFactory.h.

Note

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

27.225.2 Member Function Documentation

27.225.2.1 static 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()).

27.225.2.2 static void gdcm::QueryFactory::ListCharSets (std::ostream & *os*) [static]

List all possible CharSet.

27.225.2.3 static 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

27.225.2.4 static 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:

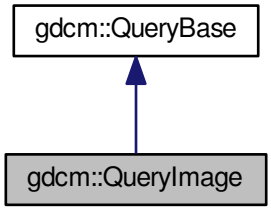
- gdcmQueryFactory.h

27.226 gdcm::QueryImage Class Reference

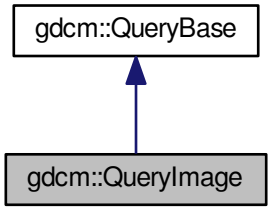
QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

`#include <gdcmQueryImage.h>`

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for gdcm::QueryImage:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const

  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

27.226.1 Detailed Description

QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

27.226.2 Member Function Documentation

27.226.2.1 std::vector<Tag> gdcm::QueryImage::GetHierachicalSearchTags (const ERootType & *inRootType*) const [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

27.226.2.2 const char* gdcm::QueryImage::GetName ( ) const [virtual]

Implements gdcm::QueryBase.

27.226.2.3 std::vector<Tag> gdcm::QueryImage::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

27.226.2.4 DataElement gdcm::QueryImage::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

27.226.2.5 std::vector<Tag> gdcm::QueryImage::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

27.226.2.6 std::vector<Tag> gdcm::QueryImage::GetUniqueTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

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

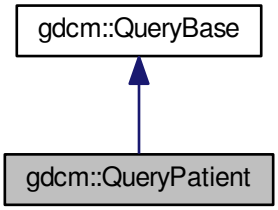
- gdcmQueryImage.h

27.227 gdcm::QueryPatient Class Reference

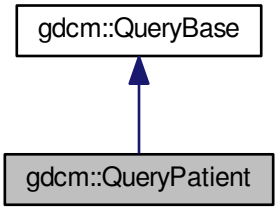
QueryPatient contains: class to construct a patient-based query for c-find and c-move.

```
#include <gdcmQueryPatient.h>
```

Inheritance diagram for gdcm::QueryPatient:



Collaboration diagram for gdcm::QueryPatient:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const

  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

27.227.1 Detailed Description

QueryPatient contains: class to construct a patient-based query for c-find and c-move.

27.227.2 Member Function Documentation

27.227.2.1 std::vector<Tag> gdcm::QueryPatient::GetHierachicalSearchTags (const ERootType & *inRootType*) const [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

27.227.2.2 const char* gdcm::QueryPatient::GetName ( ) const [virtual]

Implements gdcm::QueryBase.

27.227.2.3 std::vector<Tag> gdcm::QueryPatient::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

27.227.2.4 DataElement gdcm::QueryPatient::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

27.227.2.5 std::vector<Tag> gdcm::QueryPatient::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

27.227.2.6 std::vector<Tag> gdcm::QueryPatient::GetUniqueTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

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

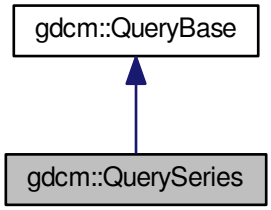
- gdcmQueryPatient.h

27.228 gdcm::QuerySeries Class Reference

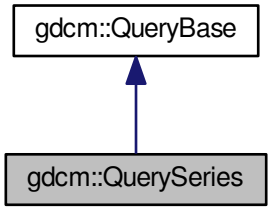
QuerySeries contains: class to construct a series-based query for c-find and c-move.

`#include <gdcmQuerySeries.h>`

Inheritance diagram for gdcm::QuerySeries:



Collaboration diagram for gdcm::QuerySeries:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const

  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

27.228.1 Detailed Description

QuerySeries contains: class to construct a series-based query for c-find and c-move.

27.228.2 Member Function Documentation

27.228.2.1 std::vector<Tag> gdcm::QuerySeries::GetHierachicalSearchTags (const ERootType & *inRootType*) const [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

27.228.2.2 const char* gdcm::QuerySeries::GetName ( ) const [virtual]

Implements gdcm::QueryBase.

27.228.2.3 std::vector<Tag> gdcm::QuerySeries::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

27.228.2.4 DataElement gdcm::QuerySeries::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

27.228.2.5 std::vector<Tag> gdcm::QuerySeries::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

27.228.2.6 std::vector<Tag> gdcm::QuerySeries::GetUniqueTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

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

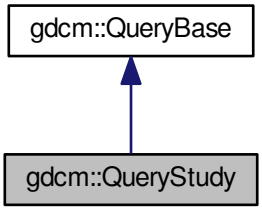
- gdcmQuerySeries.h

27.229 gdcm::QueryStudy Class Reference

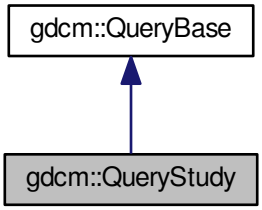
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

`#include <gdcmQueryStudy.h>`

Inheritance diagram for gdcm::QueryStudy:



Collaboration diagram for gdcm::QueryStudy:



Public Member Functions

- std::vector< Tag > GetHierachicalSearchTags (const ERootType &inRootType) const

  Return all Unique Key for a particular Query Root type (from the same level and above).
- const char * GetName () const
- std::vector< Tag > GetOptionalTags (const ERootType &inRootType) const
- DataElement GetQueryLevel () const
- std::vector< Tag > GetRequiredTags (const ERootType &inRootType) const
- std::vector< Tag > GetUniqueTags (const ERootType &inRootType) const

27.229.1 Detailed Description

QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

27.229.2 Member Function Documentation

27.229.2.1 std::vector<Tag> gdcm::QueryStudy::GetHierachicalSearchTags (const ERootType & *inRootType*) const [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements gdcm::QueryBase.

27.229.2.2 const char* gdcm::QueryStudy::GetName ( ) const [virtual]

Implements gdcm::QueryBase.

27.229.2.3 std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

27.229.2.4 DataElement gdcm::QueryStudy::GetQueryLevel ( ) const [virtual]

Implements gdcm::QueryBase.

27.229.2.5 std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

27.229.2.6 std::vector<Tag> gdcm::QueryStudy::GetUniqueTags (const ERootType & *inRootType*) const [virtual]

Implements gdcm::QueryBase.

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

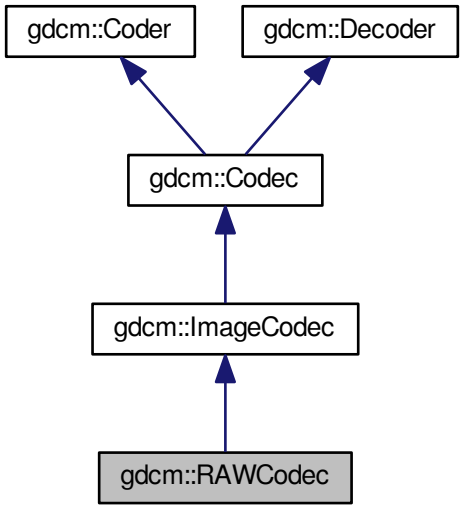
- gdcmQueryStudy.h

27.230 gdcm::RAWCodec Class Reference

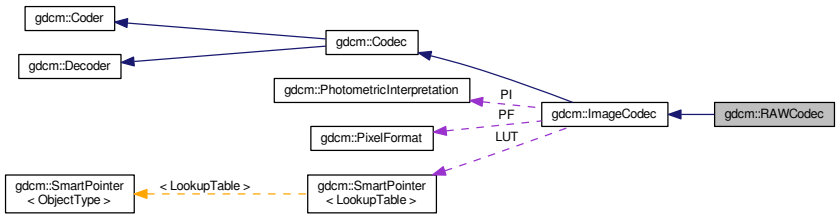
RAWCodec class.

`#include <gdcmRAWCodec.h>`

Inheritance diagram for gdcm::RAWCodec:



Collaboration diagram for gdcm::RAWCodec:



Public Member Functions

- RAWCodec ()
- ~RAWCodec ()
- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

Decode.

- bool DecodeBytes (const char *inBytes, size_t inBufferLength, char *outBytes, size_t inOutBufferLength)
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)

Protected Member Functions

- bool DecodeByStreams (std::istream &is, std::ostream &os)

Additional Inherited Members

27.230.1 Detailed Description

RAWCodec class.

27.230.2 Constructor & Destructor Documentation

27.230.2.1 gdcm::RAWCodec::RAWCodec ( )

27.230.2.2 gdcm::RAWCodec::~RAWCodec ( )

27.230.3 Member Function Documentation

27.230.3.1 bool gdcm::RAWCodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.230.3.2 bool gdcm::RAWCodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.230.3.3 virtual ImageCodec* gdcm::RAWCodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.230.3.4 bool gdcm::RAWCodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

27.230.3.5 bool gdcm::RAWCodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

27.230.3.6 bool gdcm::RAWCodec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.230.3.7 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.

27.230.3.8 bool gdcm::RAWCodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

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

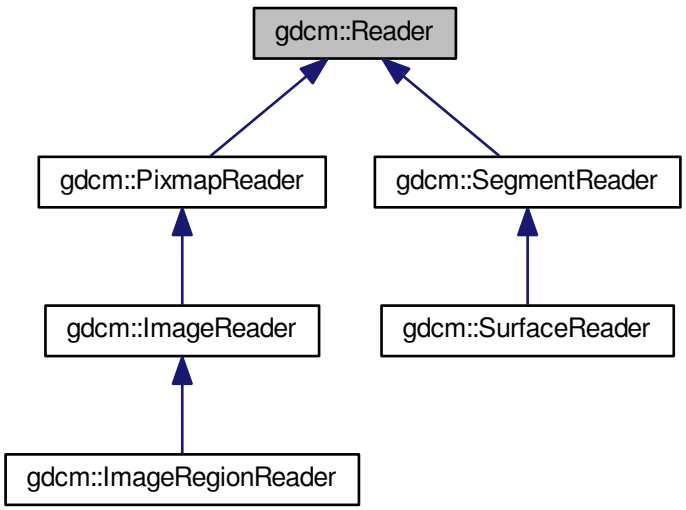
- gdcmRAWCodec.h

27.231 gdcm::Reader Class Reference

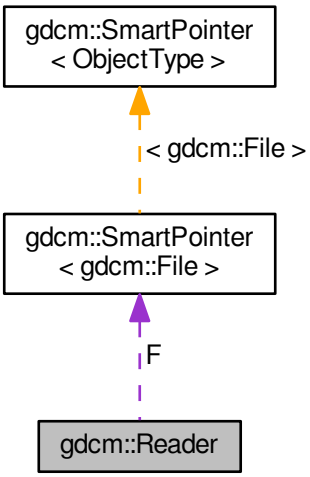
Reader ala DOM (Document Object Model)

`#include <gdcmReader.h>`

Inheritance diagram for gdcm::Reader:



Collaboration diagram for gdcm::Reader:



Public Member Functions

- Reader ()
- virtual ~Reader ()
- bool CanRead () const
- const File & GetFile () const

  Set/Get File.
- File & GetFile ()

  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

27.231.1 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).

Detailled 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, CreateFakeRTDOSE.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImage↩ HeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncrypted↩ Content.cxx, FixBrokenJ2K.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, Get↩ SequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.↩ cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadUTF8QtDir.cxx, rle2img.cxx, and TestReader.cxx.

27.231.2 Constructor & Destructor Documentation

27.231.2.1 gdcm::Reader::Reader ( ) [inline]

27.231.2.2 virtual gdcm::Reader::~Reader ( ) [virtual]

27.231.3 Member Function Documentation

27.231.3.1 bool gdcm::Reader::CanRead ( ) const

Test whether this is a DICOM file

Warning

need to call either SetFileName or SetStream first

Examples:

ReadUTF8QtDir.cxx.

27.231.3.2 const File& gdcm::Reader::GetFile ( ) const [inline]

Set/Get File.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, CreateFakeRTDOSE.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and TestReader.cxx.

27.231.3.3 File& gdcm::Reader::GetFile ( ) [inline]

Set/Get File.

27.231.3.4 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++

27.231.3.5 std::istream* gdcm::Reader::GetStreamPtr ( ) const [inline],[protected]

27.231.3.6 virtual bool gdcm::Reader::Read ( ) [virtual]

Main function to read a file.

Reimplemented in gdcm::ImageRegionReader, gdcm::PixmapReader, gdcm::ImageReader, gdcm::SegmentReader, and gdcm::SurfaceReader.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateFakeRTDOSE.cxx, csa2img.cxx, DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, FixBrokenJ2K.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, rle2img.cxx, and TestReader.cxx.

27.231.3.7 bool gdcm::Reader::ReadDataSet ( ) [protected]

27.231.3.8 bool gdcm::Reader::ReadMetaInformation ( ) [protected]

27.231.3.9 bool gdcm::Reader::ReadPreamble ( ) [protected]

27.231.3.10 bool gdcm::Reader::ReadSelectedPrivateTags (std::set< PrivateTag > const & *ptags*, bool *readvalues* = true)

Will only read the specified selected private tags.

27.231.3.11 bool gdcm::Reader::ReadSelectedTags (std::set< Tag > const & *tags*, bool *readvalues* = true)

Will only read the specified selected tags.

27.231.3.12 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	

27.231.3.13 void gdcm::Reader::SetFile (File & *file*) [inline]

Set/Get File.

27.231.3.14 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:

BasicImageAnonymizer.cs, ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, CheckBigEndianBug.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, ConvertToQImage.cxx, CreateFakeRTDOSE.cxx, csa2img.cxx,

DiffFile.cxx, DumpADAC.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpImageHeaderInfo.cxx, DumpPhilipsECHO.cxx, DuplicatePCDE.cxx, ELSCINT1WaveToText.cxx, ExtractEncryptedContent.cxx, ExtractIconFromFile.cxx, ExtractImageRegion.cs, ExtractImageRegionWithLUT.cs, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, MrProtocol.cxx, NewSequence.cs, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrintAttributes.cxx, ReadExplicitLengthSQIVR.cxx, ReadGEMSSDO.cxx, ReadMultiTimesException.cxx, ReadUTF8QtDir.cxx, rle2img.cxx, SimplePrintPatientName.cs, TestReader.cxx, and threadgdcm.cxx.

27.231.3.15 void gdcm::Reader::SetStream (std::istream & *input_stream*) [inline]

Set the open-ed stream directly.

Examples:

ReadUTF8QtDir.cxx.

27.231.4 Friends And Related Function Documentation

27.231.4.1 friend class StreamImageReader [friend]

27.231.5 Member Data Documentation

27.231.5.1 SmartPointer<File> gdcm::Reader::F [protected]

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

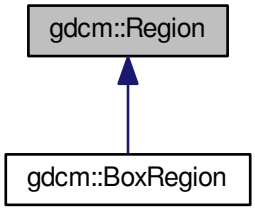
- gdcmReader.h

27.232 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.

27.232.1 Detailed Description

Class for manipulation region.

27.232.2 Constructor & Destructor Documentation

27.232.2.1 gdcm::Region::Region ( )

27.232.2.2 virtual gdcm::Region::~Region ( ) [virtual]

27.232.3 Member Function Documentation

27.232.3.1 virtual size_t gdcm::Region::Area ( ) const [pure virtual]

compute the area

Implemented in gdcm::BoxRegion.

27.232.3.2 virtual Region* gdcm::Region::Clone ( ) const [pure virtual]

Implemented in gdcm::BoxRegion.

27.232.3.3 virtual BoxRegion gdcm::Region::ComputeBoundingBox ( ) [pure virtual]

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in gdcm::BoxRegion.

27.232.3.4 virtual bool gdcm::Region::Empty ( ) const [pure virtual]

return whether this domain is empty:

Implemented in gdcm::BoxRegion.

27.232.3.5 virtual bool gdcm::Region::IsValid ( ) const [pure virtual]

return whether this is valid domain

Implemented in gdcm::BoxRegion.

27.232.3.6 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

27.233 gdcm::Rescaler Class Reference

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.

`#include <gdcmRescaler.h>`

Public Member Functions

- Rescaler ()
- ~Rescaler ()
- 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)

27.233.1 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 ouput, 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

27.233.2 Constructor & Destructor Documentation

27.233.2.1 gdcm::Rescaler::Rescaler ( ) [inline]

27.233.2.2 gdcm::Rescaler::~Rescaler ( ) [inline]

27.233.3 Member Function Documentation

27.233.3.1 PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ( )

Compute the Pixel Format of the output data Used for direct transformation

27.233.3.2 PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )

Compute the Pixel Format of the output data Used for inverse transformation

27.233.3.3 double gdcm::Rescaler::GetIntercept ( ) const [inline]

27.233.3.4 double gdcm::Rescaler::GetSlope ( ) const [inline]

27.233.3.5 bool gdcm::Rescaler::InverseRescale (char * *out*, const char * *in*, size_t *n*)

Inverse transform.

27.233.3.6 template<typename TIn > void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (char * *out*, const TIn * *in*, size_t *n*) [protected]

27.233.3.7 bool gdcm::Rescaler::Rescale (char * *out*, const char * *in*, size_t *n*)

Direct transform.

27.233.3.8 template<typename TIn > void gdcm::Rescaler::RescaleFunctionIntoBestFit (char * *out*, const TIn * *in*, size_t *n*) [protected]

27.233.3.9 void gdcm::Rescaler::SetIntercept (double *i*) [inline]

Set Intercept: used for both direct&inverse transformation.

27.233.3.10 void gdcm::Rescaler::SetMinMaxForPixelType (double *min*, double *max*) [inline]

Set target interval for output data. A best match will be computed (if possible) Used for inverse transformation

27.233.3.11 void gdcm::Rescaler::SetPixelFormat (PixelFormat const & *pf*) [inline]

Set Pixel Format of input data.

27.233.3.12 void gdcm::Rescaler::SetSlope (double *s*) [inline]

Set Slope: user for both direct&inverse transformation.

27.233.3.13 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 specifix Target Pixel Type

27.233.3.14 void gdcm::Rescaler::SetUseTargetPixelType (bool *b*)

Override default behavior of Rescale.

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

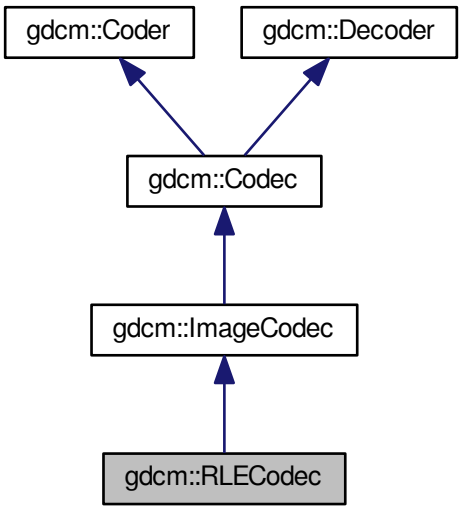
- gdcmRescaler.h

27.234 gdcm::RLECodec Class Reference

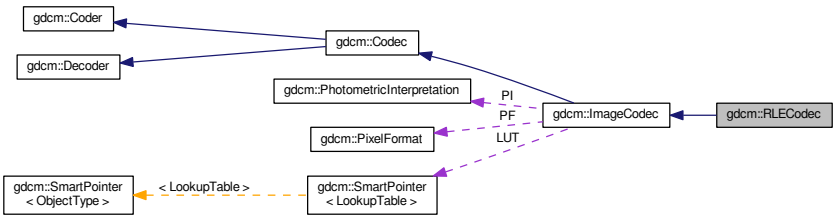
Class to do RLE.

`#include <gdcmRLECodec.h>`

Inheritance diagram for gdcm::RLECodec:



Collaboration diagram for gdcm::RLECodec:



Public Member Functions

- RLECodec ()
- ~RLECodec ()

- bool CanCode (TransferSyntax const &ts) const

  Return whether this coder support this transfer syntax (can code it)
- bool CanDecode (TransferSyntax const &ts) const

  Return whether this decoder support this transfer syntax (can decode it)
- virtual ImageCodec * Clone () const
- bool Code (DataElement const &in, DataElement &out)

  Code.
- bool Decode (DataElement const &is, DataElement &os)

  Decode.
- unsigned long GetBufferLength () const
- bool GetHeaderInfo (std::istream &is, TransferSyntax &ts)
- void SetBufferLength (unsigned long l)
- void SetLength (unsigned long l)

Protected Member Functions

- bool AppendFrameEncode (std::ostream &out, const char *data, size_t datalen)
- bool AppendRowEncode (std::ostream &out, const char *data, size_t datalen)
- bool DecodeByStreams (std::istream &is, std::ostream &os)
- 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 ()
- bool IsRowEncoder ()
- bool StartEncode (std::ostream &)
- bool StopEncode (std::ostream &)

Friends

- class ImageRegionReader

Additional Inherited Members

27.234.1 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).

27.234.2 Constructor & Destructor Documentation

27.234.2.1 gdcm::RLECodec::RLECodec ( )

27.234.2.2 gdcm::RLECodec::~RLECodec ( )

27.234.3 Member Function Documentation

27.234.3.1 bool gdcm::RLECodec::AppendFrameEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.234.3.2 bool gdcm::RLECodec::AppendRowEncode (std::ostream & *out*, const char * *data*, size_t *datalen*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.234.3.3 bool gdcm::RLECodec::CanCode (TransferSyntax const &) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from gdcm::ImageCodec.

27.234.3.4 bool gdcm::RLECodec::CanDecode (TransferSyntax const &) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from gdcm::ImageCodec.

27.234.3.5 virtual ImageCodec* gdcm::RLECodec::Clone ( ) const [virtual]

Implements gdcm::ImageCodec.

27.234.3.6 bool gdcm::RLECodec::Code (DataElement const & *in_*, DataElement & *out_*) [virtual]

Code.

Reimplemented from gdcm::Coder.

27.234.3.7 bool gdcm::RLECodec::Decode (DataElement const &, DataElement &) [virtual]

Decode.

Reimplemented from gdcm::ImageCodec.

27.234.3.8 bool gdcm::RLECodec::DecodeByStreams (std::istream & *is*, std::ostream & *os*) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.234.3.9 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]

27.234.3.10 unsigned long gdcm::RLECodec::GetBufferLength ( ) const [inline]

27.234.3.11 bool gdcm::RLECodec::GetHeaderInfo (std::istream & *is*, TransferSyntax & *ts*) [virtual]

Reimplemented from gdcm::ImageCodec.

27.234.3.12 bool gdcm::RLECodec::IsFrameEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.234.3.13 bool gdcm::RLECodec::IsRowEncoder ( ) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.234.3.14 void gdcm::RLECodec::SetBufferLength (unsigned long *l*) [inline]

27.234.3.15 void gdcm::RLECodec::SetLength (unsigned long *l*) [inline]

27.234.3.16 bool gdcm::RLECodec::StartEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.234.3.17 bool gdcm::RLECodec::StopEncode (std::ostream &) [protected], [virtual]

Reimplemented from gdcm::ImageCodec.

27.234.4 Friends And Related Function Documentation

27.234.4.1 friend class ImageRegionReader [friend]

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

- gdcmRLECodec.h

27.235 gdcm::network::RoleSelectionSub Class Reference

RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

`#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

27.235.1 Detailed Description

RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

27.235.2 Constructor & Destructor Documentation

27.235.2.1 gdcm::network::RoleSelectionSub::RoleSelectionSub ( )

27.235.3 Member Function Documentation

27.235.3.1 void gdcm::network::RoleSelectionSub::Print (std::ostream & *os*) const

27.235.3.2 std::istream& gdcm::network::RoleSelectionSub::Read (std::istream & *is*)

27.235.3.3 void gdcm::network::RoleSelectionSub::SetTuple (const char * *uid*, uint8_t *scurole*, uint8_t *scprole*)

27.235.3.4 size_t gdcm::network::RoleSelectionSub::Size ( ) const

27.235.3.5 const std::ostream& gdcm::network::RoleSelectionSub::Write (std::ostream & *os*) const

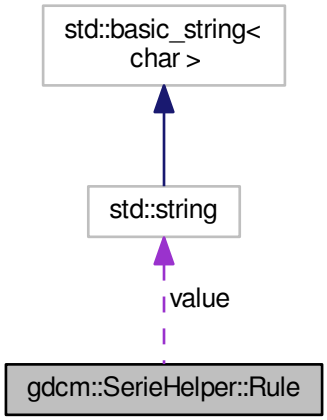
The documentation for this class was generated from the following file:

- gdcmRoleSelectionSub.h

27.236 gdcm::SerieHelper::Rule Struct Reference

`#include <gdcmSerieHelper.h>`

Collaboration diagram for gdcm::SerieHelper::Rule:



Public Attributes

- uint16_t elem
- uint16_t group
- int op
- std::string value

27.236.1 Member Data Documentation

27.236.1.1 uint16_t gdcm::SerieHelper::Rule::elem

27.236.1.2 uint16_t gdcm::SerieHelper::Rule::group

27.236.1.3 int gdcm::SerieHelper::Rule::op

27.236.1.4 std::string gdcm::SerieHelper::Rule::value

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

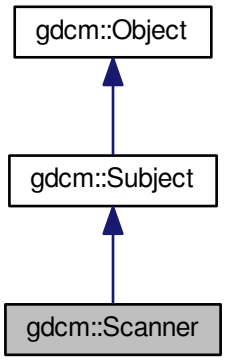
- gdcmSerieHelper.h

27.237 gdcm::Scanner Class Reference

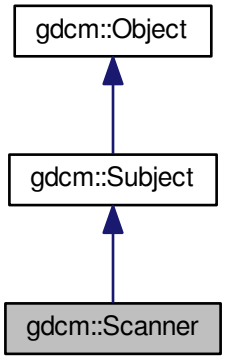
Scanner This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

`#include <gdcmScanner.h>`

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



Classes

- struct Itstr

Public Types

- typedef MappingType::const_iterator ConstIterator
- typedef std::map< const char *, TagToValue, Itstr > MappingType
- typedef std::map< Tag, const char * > TagToValue
- typedef TagToValue::value_type TagToValueValueType
- typedef std::set< std::string > ValuesType

Public Member Functions

- Scanner ()
- ~Scanner ()
- 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

  Print result.
- bool Scan (Directory::FilenamesType const &filenames)

  Start the scan !

Static Public Member Functions

- static SmartPointer< Scanner > New ()

  for wrapped language: instanciate a reference counted object

Protected Member Functions

- void ProcessPublicTag (StringFilter &sf, const char *filename)

Friends

- std::ostream & operator<< (std::ostream &_os, const Scanner &s)

27.237.1 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 mimimum 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, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

27.237.2 Member Typedef Documentation

27.237.2.1 typedef MappingType::const_iterator gdcm::Scanner::ConstIterator

27.237.2.2 typedef std::map<const char *,TagToValue, ltstr> gdcm::Scanner::MappingType

27.237.2.3 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)

27.237.2.4 typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType

27.237.2.5 typedef std::set< std::string > gdcm::Scanner::ValuesType

27.237.3 Constructor & Destructor Documentation

27.237.3.1 gdcm::Scanner::Scanner ( ) [inline]

27.237.3.2 gdcm::Scanner::~Scanner ( )

27.237.4 Member Function Documentation

27.237.4.1 void gdcm::Scanner::AddPrivateTag (PrivateTag const & *t*)

27.237.4.2 void gdcm::Scanner::AddSkipTag (Tag const & *t*)

Add a tag that will need to be skipped. Those are root level skip tags.

27.237.4.3 void gdcm::Scanner::AddTag (Tag const & *t*)

Add a tag that will need to be read. Those are root level skip tags.

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

27.237.4.4 ConstIterator gdcm::Scanner::Begin ( ) const [inline]

27.237.4.5 void gdcm::Scanner::ClearSkipTags ( )

27.237.4.6 void gdcm::Scanner::ClearTags ( )

27.237.4.7 ConstIterator gdcm::Scanner::End ( ) const [inline]

27.237.4.8 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'

27.237.4.9 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'

27.237.4.10 Directory::FilenamesType const& gdcm::Scanner::GetFilenames ( ) const [inline]

27.237.4.11 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.

27.237.4.12 TagToValue const& gdcm::Scanner::GetMapping (const char * *filename*) const

Get the std::map mapping filenames to value for file 'filename'.

Examples:

DumpToSQLITE3.cxx, and SimpleScanner.cxx.

27.237.4.13 TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (Tag const & *t*, const char * *value*) const

See GetFilenameFromTagToValue(). This is simply GetFilenameFromTagToValue followed.

27.237.4.14 MappingType const& gdcm::Scanner::GetMappings ( ) const [inline]

Mappings are the mapping from a particular tag to the map, mapping filename to value:

27.237.4.15 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)

27.237.4.16 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 !

27.237.4.17 ValuesType const& gdcm::Scanner::GetValues ( ) const [inline]

Get all the values found (in lexicographic order)

Examples:

SortImage.cxx, and VolumeSorter.cxx.

27.237.4.18 ValuesType gdcm::Scanner::GetValues (Tag const & *t*) const

Get all the values found (in lexicographic order) associated with Tag 't'.

27.237.4.19 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, and SimpleScanner.cxx.

27.237.4.20 static SmartPointer<Scanner> gdcm::Scanner::New ( ) [inline], [static]

for wrapped language: instanciate a reference counted object

27.237.4.21 void gdcm::Scanner::Print (std::ostream & *os*) const [virtual]

Print result.

Reimplemented from gdcm::Object.

Referenced by gdcm::operator<<().

27.237.4.22 void gdcm::Scanner::ProcessPublicTag (StringFilter & *sf,* const char * *filename*) [protected]

27.237.4.23 bool gdcm::Scanner::Scan (Directory::FilenamesType const & *filenames*)

Start the scan !

Examples:

DiscriminateVolume.cxx, DumpToSQLITE3.cxx, SimpleScanner.cxx, SortImage.cxx, and VolumeSorter.cxx.

27.237.5 Friends And Related Function Documentation

27.237.5.1 std::ostream& operator<< (std::ostream & *_os,* const Scanner & *s*) [friend]

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

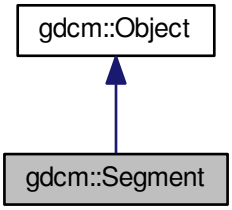
- gdcmScanner.h

27.238 gdcm::Segment Class Reference

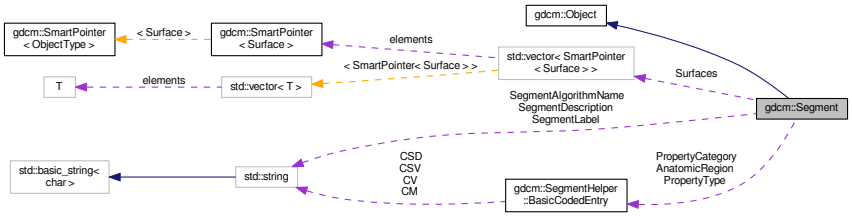
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

`#include <gdcmSegment.h>`

Inheritance diagram for gdcm::Segment:



Collaboration diagram for gdcm::Segment:



Public Types

- enum ALGOType {
  MANUAL = 0,
  AUTOMATIC,
  ALGOType_END }

- typedef std::vector
  < SmartPointer< Surface > > SurfaceVector

Public Member Functions

- Segment ()
- virtual ~Segment ()
- void AddSurface (SmartPointer< Surface > surface)
- SegmentHelper::BasicCodedEntry
  const & GetAnatomicRegion () const
- SegmentHelper::BasicCodedEntry & GetAnatomicRegion ()
- SegmentHelper::BasicCodedEntry
  const & GetPropertyCategory () const
- SegmentHelper::BasicCodedEntry & GetPropertyCategory ()
- SegmentHelper::BasicCodedEntry
  const & GetPropertyType () const
- SegmentHelper::BasicCodedEntry & GetPropertyType ()
- 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 const & GetSurfaces () const
- SurfaceVector & GetSurfaces ()
- void SetAnatomicRegion (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyCategory (SegmentHelper::BasicCodedEntry const &BSE)
- void SetPropertyType (SegmentHelper::BasicCodedEntry const &BSE)
- 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)

Static Public Member Functions

- static ALGOType GetALGOType (const char *type)
- static const char * GetALGOTypeString (ALGOType type)

Protected Attributes

- SegmentHelper::BasicCodedEntry AnatomicRegion
- SegmentHelper::BasicCodedEntry PropertyCategory
- SegmentHelper::BasicCodedEntry PropertyType
- std::string SegmentAlgorithmName
- ALGOType SegmentAlgorithmType

- std::string SegmentDescription
- std::string SegmentLabel
- unsigned short SegmentNumber
- unsigned long SurfaceCount
- SurfaceVector Surfaces

Additional Inherited Members

27.238.1 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

27.238.2 Member Typedef Documentation

27.238.2.1 typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector

27.238.3 Member Enumeration Documentation

27.238.3.1 enum gdcm::Segment::ALGOType

Enumerator

MANUAL

AUTOMATIC

ALGOType_END

27.238.4 Constructor & Destructor Documentation

27.238.4.1 gdcm::Segment::Segment ( )

27.238.4.2 virtual gdcm::Segment::~Segment ( ) [virtual]

27.238.5 Member Function Documentation

27.238.5.1 void gdcm::Segment::AddSurface (SmartPointer< Surface > *surface*)

27.238.5.2 static ALGOType gdcm::Segment::GetALGOType (const char * *type*) [static]

27.238.5.3 static const char* gdcm::Segment::GetALGOTypeString (ALGOType *type*) [static]

27.238.5.4 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion ( ) const

27.238.5.5 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ( )

27.238.5.6 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory ( ) const

27.238.5.7 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ( )

27.238.5.8 SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType ( ) const

27.238.5.9 SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ( )

27.238.5.10 const char* gdcm::Segment::GetSegmentAlgorithmName ( ) const

27.238.5.11 ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const

27.238.5.12 const char* gdcm::Segment::GetSegmentDescription ( ) const

27.238.5.13 const char* gdcm::Segment::GetSegmentLabel ( ) const

27.238.5.14 unsigned short gdcm::Segment::GetSegmentNumber ( ) const

27.238.5.15 SmartPointer< Surface > gdcm::Segment::GetSurface (const unsigned int *idx* = 0) const

27.238.5.16 unsigned long gdcm::Segment::GetSurfaceCount ( )

27.238.5.17 SurfaceVector const& gdcm::Segment::GetSurfaces ( ) const

27.238.5.18 SurfaceVector& gdcm::Segment::GetSurfaces ( )

27.238.5.19 void gdcm::Segment::SetAnatomicRegion (SegmentHelper::BasicCodedEntry const & *BSE*)

27.238.5.20 void gdcm::Segment::SetPropertyCategory (SegmentHelper::BasicCodedEntry const & *BSE*)

27.238.5.21 void gdcm::Segment::SetPropertyType (SegmentHelper::BasicCodedEntry const & *BSE*)

27.238.5.22 void gdcm::Segment::SetSegmentAlgorithmName (const char * *name*)

27.238.5.23 void gdcm::Segment::SetSegmentAlgorithmType (ALGOType *type*)

27.238.5.24 void gdcm::Segment::SetSegmentAlgorithmType (const char * *typeStr*)

27.238.5.25 void gdcm::Segment::SetSegmentDescription (const char * *description*)

27.238.5.26 void gdcm::Segment::SetSegmentLabel (const char * *label*)

27.238.5.27 void gdcm::Segment::SetSegmentNumber (const unsigned short *num*)

27.238.5.28 void gdcm::Segment::SetSurfaceCount (const unsigned long *nb*)

27.238.6 Member Data Documentation

27.238.6.1 SegmentHelper::BasicCodedEntry gdcm::Segment::AnatomicRegion [protected]

27.238.6.2 SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyCategory [protected]

27.238.6.3 SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyType [protected]

27.238.6.4 std::string gdcm::Segment::SegmentAlgorithmName [protected]

27.238.6.5 ALGOType gdcm::Segment::SegmentAlgorithmType [protected]

27.238.6.6 std::string gdcm::Segment::SegmentDescription [protected]

27.238.6.7 std::string gdcm::Segment::SegmentLabel [protected]

27.238.6.8 unsigned short gdcm::Segment::SegmentNumber [protected]

27.238.6.9 unsigned long gdcm::Segment::SurfaceCount [protected]

27.238.6.10 SurfaceVector gdcm::Segment::Surfaces [protected]

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

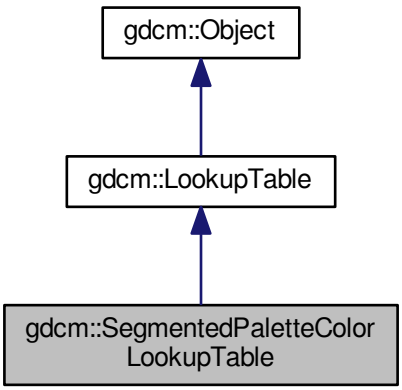
- gdcmSegment.h

27.239 gdcm::SegmentedPaletteColorLookupTable Class Reference

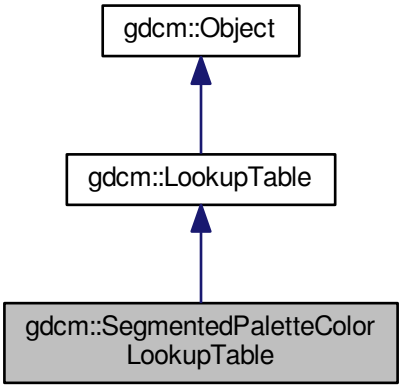
SegmentedPaletteColorLookupTable class.

`#include <gdcmSegmentedPaletteColorLookupTable.h>`

Inheritance diagram for gdcm::SegmentedPaletteColorLookupTable:



Collaboration diagram for gdcm::SegmentedPaletteColorLookupTable:



Public Member Functions

- SegmentedPaletteColorLookupTable ()
- ∼SegmentedPaletteColorLookupTable ()
- void Print (std::ostream &) const
- void SetLUT (LookupTableType type, const unsigned char ∗array, unsigned int length)

  Initialize a SegmentedPaletteColorLookupTable.

Additional Inherited Members

27.239.1 Detailed Description

SegmentedPaletteColorLookupTable class.

27.239.2 Constructor & Destructor Documentation

27.239.2.1 gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )

27.239.2.2 gdcm::SegmentedPaletteColorLookupTable::∼SegmentedPaletteColorLookupTable ( )

27.239.3 Member Function Documentation

27.239.3.1 void gdcm::SegmentedPaletteColorLookupTable::Print (std::ostream &) const [inline], [virtual]

Reimplemented from gdcm::LookupTable.

27.239.3.2 void gdcm::SegmentedPaletteColorLookupTable::SetLUT (LookupTableType *type*, const unsigned char * *array*, unsigned int *length*) [virtual]

Initialize a SegmentedPaletteColorLookupTable.

Reimplemented from gdcm::LookupTable.

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

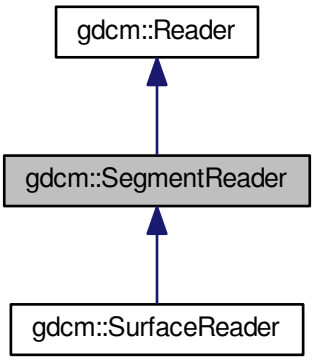
- gdcmSegmentedPaletteColorLookupTable.h

27.240 gdcm::SegmentReader Class Reference

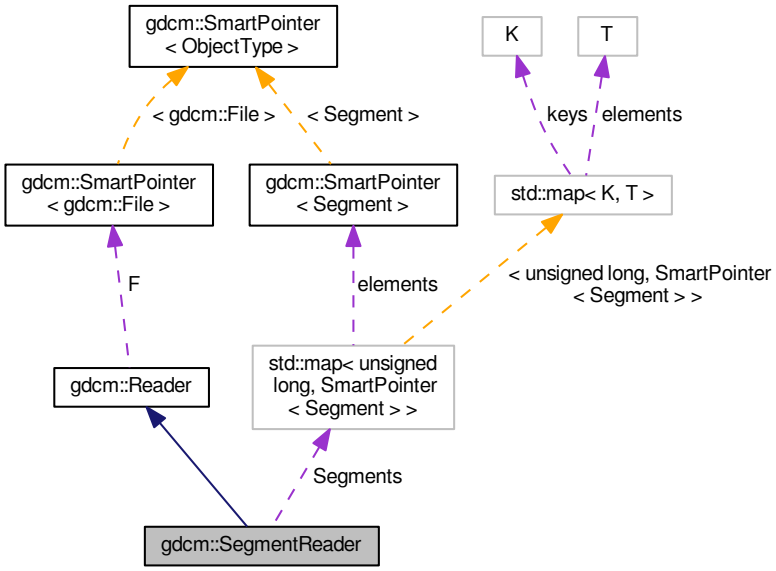
This class defines a segment reader. It reads attributes of group 0x0062.

`#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 ()
- virtual ~SegmentReader ()
- const SegmentVector GetSegments () const
- SegmentVector GetSegments ()
- virtual bool Read ()

  Read.

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 Attributes

- SegmentMap Segments

27.240.1 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

27.240.2 Member Typedef Documentation

27.240.2.1 typedef std::map< unsigned long, SmartPointer< Segment > > gdcm::SegmentReader::SegmentMap [protected]

27.240.2.2 typedef std::vector< SmartPointer< Segment > > gdcm::SegmentReader::SegmentVector

27.240.3 Constructor & Destructor Documentation

27.240.3.1 gdcm::SegmentReader::SegmentReader ( )

27.240.3.2 virtual gdcm::SegmentReader::~SegmentReader ( ) [virtual]

27.240.4 Member Function Documentation

27.240.4.1 const SegmentVector gdcm::SegmentReader::GetSegments ( ) const

27.240.4.2 SegmentVector gdcm::SegmentReader::GetSegments ( )

27.240.4.3 virtual bool gdcm::SegmentReader::Read ( ) [virtual]

Read.

Reimplemented from gdcm::Reader.

Reimplemented in gdcm::SurfaceReader.

27.240.4.4 bool gdcm::SegmentReader::ReadSegment (const Item & *segmentItem,* const unsigned int *idx*) [protected]

27.240.4.5 bool gdcm::SegmentReader::ReadSegments ( ) [protected]

27.240.5 Member Data Documentation

27.240.5.1 SegmentMap gdcm::SegmentReader::Segments [protected]

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

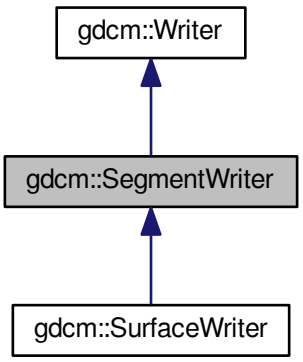
- gdcmSegmentReader.h

27.241 gdcm::SegmentWriter Class Reference

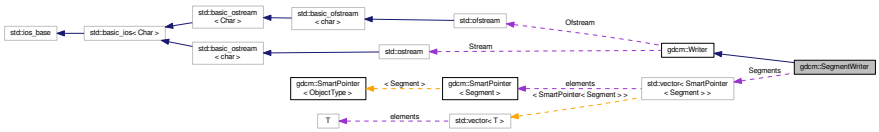
This class defines a segment writer. It writes attributes of group 0x0062.

`#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 ()
- virtual ~SegmentWriter ()
- void AddSegment (SmartPointer< Segment > segment)
- unsigned int GetNumberOfSegments () const
- SmartPointer< Segment > GetSegment (const unsigned int idx=0) const
- const SegmentVector & GetSegments () const
- SegmentVector & GetSegments ()
- void SetNumberOfSegments (const unsigned int size)

- void SetSegments (SegmentVector &segments)
- bool Write ()

  Write.

Protected Member Functions

- bool PrepareWrite ()

Protected Attributes

- SegmentVector Segments

27.241.1 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

27.241.2 Member Typedef Documentation

27.241.2.1 typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector

27.241.3 Constructor & Destructor Documentation

27.241.3.1 gdcm::SegmentWriter::SegmentWriter ( )

27.241.3.2 virtual gdcm::SegmentWriter::~SegmentWriter ( ) [virtual]

27.241.4 Member Function Documentation

27.241.4.1 void gdcm::SegmentWriter::AddSegment (SmartPointer< Segment > *segment*)

27.241.4.2 unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( ) const

27.241.4.3 SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (const unsigned int *idx* = 0) const

27.241.4.4 const SegmentVector& gdcm::SegmentWriter::GetSegments ( ) const

27.241.4.5 SegmentVector& gdcm::SegmentWriter::GetSegments ( )

27.241.4.6 bool gdcm::SegmentWriter::PrepareWrite ( ) [protected]

27.241.4.7 void gdcm::SegmentWriter::SetNumberOfSegments (const unsigned int *size*)

27.241.4.8 void gdcm::SegmentWriter::SetSegments (SegmentVector & *segments*)

27.241.4.9 bool gdcm::SegmentWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::Writer.

Reimplemented in gdcm::SurfaceWriter.

27.241.5 Member Data Documentation

27.241.5.1 SegmentVector gdcm::SegmentWriter::Segments [protected]

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

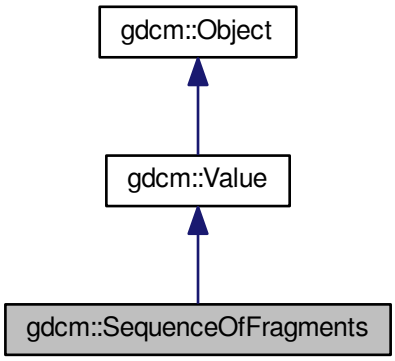
- gdcmSegmentWriter.h

27.242 gdcm::SequenceOfFragments Class Reference

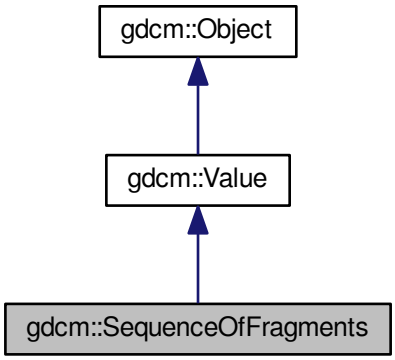
Class to represent a Sequence Of Fragments.

`#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 ()

  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

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfFragments () const
- const BasicOffsetTable & GetTable () const

- BasicOffsetTable & GetTable ()
- bool operator== (const Value &val) const
- void Print (std::ostream &os) const
- 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)

  Sets the actual SQ length.
- template<typename TSwap >
  std::ostream const & Write (std::ostream &os) const
- bool WriteBuffer (std::ostream &os) const

Static Public Member Functions

- static SmartPointer
  < SequenceOfFragments > New ()

Additional Inherited Members

27.242.1 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:

FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, and GetJPEGSamplePrecision.cxx.

27.242.2 Member Typedef Documentation

27.242.2.1 typedef FragmentVector::const_iterator gdcm::SequenceOfFragments::ConstIterator

27.242.2.2 typedef std::vector<Fragment> gdcm::SequenceOfFragments::FragmentVector

27.242.2.3 typedef FragmentVector::iterator gdcm::SequenceOfFragments::Iterator

27.242.2.4 typedef FragmentVector::size_type gdcm::SequenceOfFragments::SizeType

27.242.3 Constructor & Destructor Documentation

27.242.3.1 gdcm::SequenceOfFragments::SequenceOfFragments ( ) [inline]

constructor (UndefinedLength by default)

27.242.4 Member Function Documentation

27.242.4.1 void gdcm::SequenceOfFragments::AddFragment (Fragment const & *item*)

Appends a Fragment to the already added ones.

Examples:

FixBrokenJ2K.cxx.

27.242.4.2 Iterator gdcm::SequenceOfFragments::Begin ( ) [inline]

27.242.4.3 ConstIterator gdcm::SequenceOfFragments::Begin ( ) const [inline]

27.242.4.4 void gdcm::SequenceOfFragments::Clear ( ) [virtual]

Clear.

Implements gdcm::Value.

27.242.4.5 unsigned long gdcm::SequenceOfFragments::ComputeByteLength ( ) const

27.242.4.6 VL gdcm::SequenceOfFragments::ComputeLength ( ) const

27.242.4.7 Iterator gdcm::SequenceOfFragments::End ( ) [inline]

27.242.4.8 ConstIterator gdcm::SequenceOfFragments::End ( ) const [inline]

27.242.4.9 bool gdcm::SequenceOfFragments::GetBuffer (char * *buffer,* unsigned long *length*) const

27.242.4.10 bool gdcm::SequenceOfFragments::GetFragBuffer (unsigned int *fragNb,* char * *buffer,* unsigned long & *length*) const

27.242.4.11 const Fragment& gdcm::SequenceOfFragments::GetFragment (SizeType *num*) const

Examples:

FixBrokenJ2K.cxx, and FixJAIBugJPEGLS.cxx.

27.242.4.12 VL gdcm::SequenceOfFragments::GetLength ( ) const [inline], [virtual]

Returns the SQ length, as read from disk.

Implements gdcm::Value.

27.242.4.13 SizeType gdcm::SequenceOfFragments::GetNumberOfFragments ( ) const

Examples:

FixJAIBugJPEGLS.cxx.

27.242.4.14 const BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) const [inline]

27.242.4.15 BasicOffsetTable& gdcm::SequenceOfFragments::GetTable ( ) [inline]

27.242.4.16 static SmartPointer<SequenceOfFragments> gdcm::SequenceOfFragments::New ( ) [inline], [static]

27.242.4.17 bool gdcm::SequenceOfFragments::operator== (const Value & *val*) const [inline], [virtual]

Implements gdcm::Value.

27.242.4.18 void gdcm::SequenceOfFragments::Print (std::ostream & *os*) const [inline], [virtual]

Reimplemented from gdcm::Object.

27.242.4.19 template<typename TSwap > std::istream& gdcm::SequenceOfFragments::Read (std::istream & *is*, bool *readvalues* = true) [inline]

27.242.4.20 template<typename TSwap > std::istream& gdcm::SequenceOfFragments::ReadPreValue (std::istream & *is*) [inline]

References gdcmDebugMacro.

27.242.4.21 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().

27.242.4.22 void gdcm::SequenceOfFragments::SetLength (VL *length*) [inline], [virtual]

Sets the actual SQ length.

Implements gdcm::Value.

27.242.4.23 template<typename TSwap > std::ostream const& gdcm::SequenceOfFragments::Write (std::ostream & *os*) const [inline]

References gdcm::VL::Write(), and gdcm::Tag::Write().

27.242.4.24 bool gdcm::SequenceOfFragments::WriteBuffer (std::ostream & *os*) const

Examples:

GetJPEGSamplePrecision.cxx.

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

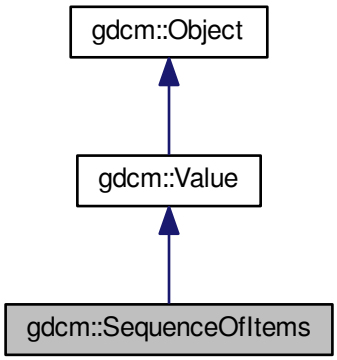
- gdcmSequenceOfFragments.h

27.243 gdcm::SequenceOfItems Class Reference

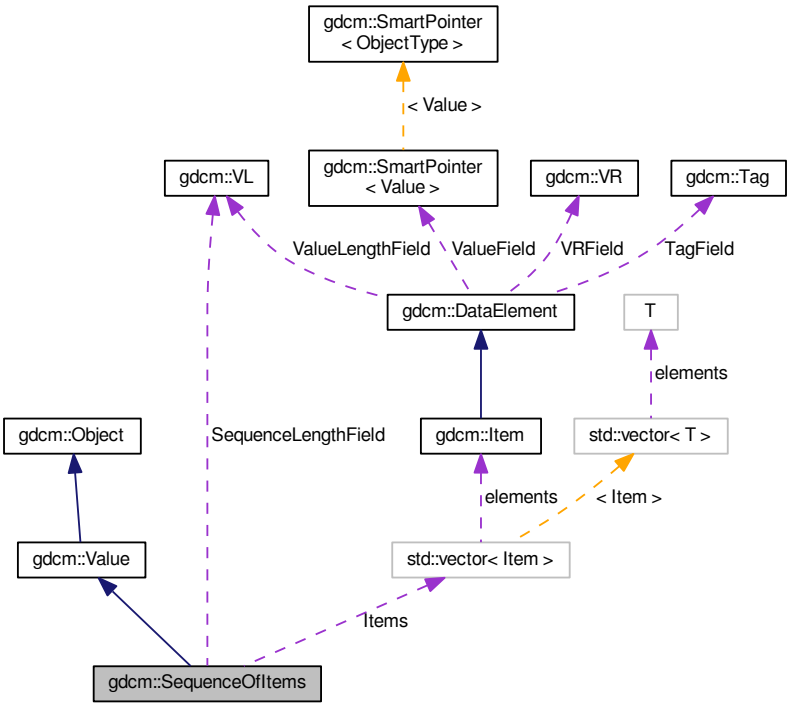
Class to represent a Sequence Of Items (value representation : SQ)

`#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.
- Iterator Begin ()
- ConstIterator Begin () const
- void Clear ()

  remove all items within the sequence
- template<typename TDE >
  VL ComputeLength () const
- Iterator End ()
- ConstIterator End () const

- bool FindDataElement (const Tag &t) const
- const Item & GetItem (SizeType position) const
- Item & GetItem (SizeType position)
- VL GetLength () const

  Returns the SQ length, as read from disk.
- SizeType GetNumberOfItems () const
- bool IsUndefinedLength () const

  return if Value Length if of undefined length
- SequenceOfItems & operator= (const SequenceOfItems &val)
- bool operator== (const Value &val) const
- void Print (std::ostream &os) const
- template<typename TDE , typename TSwap >
  std::istream & Read (std::istream &is, bool readvalues=true)
- bool RemoveItemByIndex (const SizeType index)
- void SetLength (VL length)

  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

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

27.243.1 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.

27.243.2 Member Typedef Documentation

27.243.2.1 typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator

27.243.2.2 typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector

27.243.2.3 typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator

27.243.2.4 typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType

27.243.3 Constructor & Destructor Documentation

27.243.3.1 gdcm::SequenceOfItems::SequenceOfItems ( ) [inline]

constructor (UndefinedLength by default)

27.243.4 Member Function Documentation

27.243.4.1 void gdcm::SequenceOfItems::AddItem (Item const & *item*)

Appends an Item to the already added ones.

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

27.243.4.2 Iterator gdcm::SequenceOfItems::Begin ( ) [inline]

27.243.4.3 ConstIterator gdcm::SequenceOfItems::Begin ( ) const [inline]

27.243.4.4 void gdcm::SequenceOfItems::Clear ( ) [virtual]

remove all items within the sequence

Implements gdcm::Value.

27.243.4.5 template<typename TDE > VL gdcm::SequenceOfItems::ComputeLength ( ) const

27.243.4.6 Iterator gdcm::SequenceOfItems::End ( ) [inline]

27.243.4.7 ConstIterator gdcm::SequenceOfItems::End ( ) const [inline]

27.243.4.8 bool gdcm::SequenceOfItems::FindDataElement (const Tag & *t*) const

27.243.4.9 const Item& gdcm::SequenceOfItems::GetItem (SizeType *position*) const

Examples:

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, and ReadAndDumpDICOMDIR.cxx.

27.243.4.10 Item& gdcm::SequenceOfItems::GetItem (SizeType *position*)

27.243.4.11 VL gdcm::SequenceOfItems::GetLength ( ) const [inline], [virtual]

Returns the SQ length, as read from disk.

Implements gdcm::Value.

27.243.4.12 SizeType gdcm::SequenceOfItems::GetNumberOfItems ( ) const [inline]

Examples:

ChangeSequenceUltrasound.cxx, DumpExamCard.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, ExtractEncryptedContent.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, and LargeVRDSExplicit.cxx.

27.243.4.13 bool gdcm::SequenceOfItems::IsUndefinedLength ( ) const [inline]

return if Value Length if of undefined length

27.243.4.14 static SmartPointer<SequenceOfItems> gdcm::SequenceOfItems::New ( ) [inline], [static]

Examples:

NewSequence.cs.

27.243.4.15 SequenceOfItems& gdcm::SequenceOfItems::operator= (const SequenceOfItems & *val*) [inline]

References Items, and SequenceLengthField.

27.243.4.16 bool gdcm::SequenceOfItems::operator== (const Value & *val*) const [inline], [virtual]

Implements gdcm::Value.

References Items, and SequenceLengthField.

27.243.4.17 void gdcm::SequenceOfItems::Print (std::ostream & *os*) const [inline], [virtual]

Reimplemented from gdcm::Object.

27.243.4.18 template<typename TDE , typename TSwap > std::istream& gdcm::SequenceOfItems::Read (std::istream & *is*, bool *readvalues* = true) [inline]

Examples:

ReadExplicitLengthSQIVR.cxx.

References gdcm::Item::Clear(), gdcmDebugMacro, gdcmWarningMacro, gdcm::Exception::GetDescription(), gdcm::Item::GetNestedDataSet(), gdcm::DataElement::GetTag(), gdcm::DataElement::GetVL(), gdcm::Item::Read(), and gdcm::DataSet::Size().

27.243.4.19 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

27.243.4.20 void gdcm::SequenceOfItems::SetLength (VL *length*) [inline], [virtual]

Sets the actual SQ length.

Implements gdcm::Value.

Examples:

ReadExplicitLengthSQIVR.cxx.

27.243.4.21 void gdcm::SequenceOfItems::SetLengthToUndefined ( )

Properly set the Sequence of Item to be undefined length.

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, GenAllVR.cxx, GenLongSeqs.cxx, and GenSeqs.cxx.

27.243.4.22 void gdcm::SequenceOfItems::SetNumberOfItems (SizeType *n*) [inline]

27.243.4.23 template<typename TDE , typename TSwap > std::ostream const& gdcm::SequenceOfItems::Write (std::ostream & *os*) const [inline]

References gdcm::VL::Write(), and gdcm::Tag::Write().

27.243.5 Member Data Documentation

27.243.5.1 ItemVector gdcm::SequenceOfItems::Items

Vector of Sequence Items.

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

27.243.5.2 VL gdcm::SequenceOfItems::SequenceLengthField

Total length of the Sequence (or 0xffffffff) if undefined.

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

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

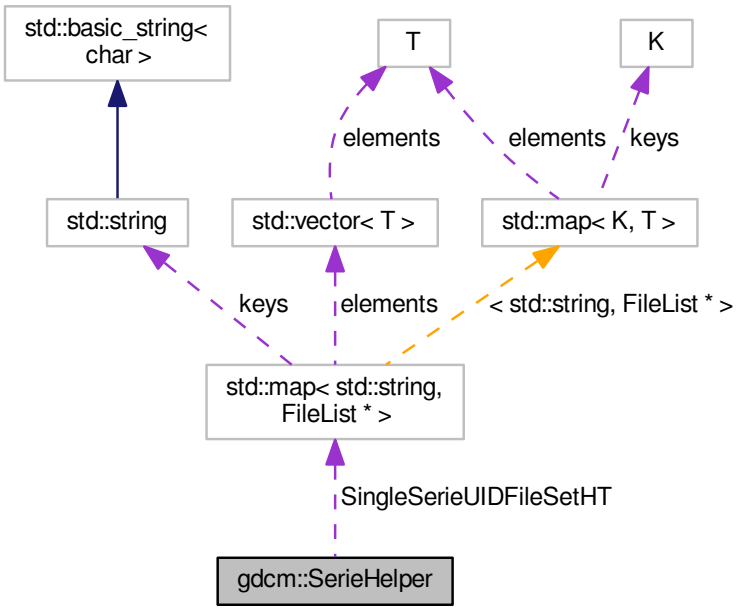
- gdcmSequenceOfItems.h

27.244 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 disapear soon, you've been warned.

`#include <gdcmSerieHelper.h>`

Collaboration diagram for gdcm::SerieHelper:



Classes

- struct Rule

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

- 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 ImagePositionPatientOrdering (FileList *fileSet)
- bool UserOrdering (FileList *fileSet)

Protected Attributes

- SingleSerieUIDFileSetmap::iterator ItFileSetHt
- SingleSerieUIDFileSetmap SingleSerieUIDFileSetHT

27.244.1 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 disapear soon, you've been warned.

Instead see ImageHelper or IPPSorter

27.244.2 Member Typedef Documentation

27.244.2.1 typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions [protected]

27.244.2.2 typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap [protected]

27.244.3 Constructor & Destructor Documentation

27.244.3.1 gdcm::SerieHelper::SerieHelper ( )

27.244.3.2 gdcm::SerieHelper::~SerieHelper ( )

27.244.4 Member Function Documentation

27.244.4.1 bool gdcm::SerieHelper::AddFile (FileWithName & *header*) [protected]

27.244.4.2 void gdcm::SerieHelper::AddFileName (std::string const & *filename*) [protected]

27.244.4.3 void gdcm::SerieHelper::AddRestriction (const std::string & *tag*)

27.244.4.4 void gdcm::SerieHelper::AddRestriction (uint16_t *group,* uint16_t *elem,* std::string const & *value,* int *op*)

27.244.4.5 void gdcm::SerieHelper::AddRestriction (const Tag & *tag*) [protected]

27.244.4.6 void gdcm::SerieHelper::Clear ( )

27.244.4.7 void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )

27.244.4.8 std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier (File * *inFile*)

27.244.4.9 bool gdcm::SerieHelper::FileNameOrdering (FileList * *fileList*) [protected]

27.244.4.10 FileList* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )

27.244.4.11 FileList* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )

27.244.4.12 bool gdcm::SerieHelper::ImagePositionPatientOrdering (FileList * *fileSet*) [protected]

27.244.4.13 void gdcm::SerieHelper::OrderFileList (FileList * *fileSet*)

27.244.4.14 void gdcm::SerieHelper::SetDirectory (std::string const & *dir,* bool *recursive* = false)

27.244.4.15 void gdcm::SerieHelper::SetLoadMode (int) [inline]

27.244.4.16 void gdcm::SerieHelper::SetUseSeriesDetails (bool *useSeriesDetails*)

27.244.4.17 bool gdcm::SerieHelper::UserOrdering (FileList * *fileSet*) [protected]

27.244.5 Member Data Documentation

27.244.5.1 SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt [protected]

27.244.5.2 SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT [protected]

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

- gdcmSerieHelper.h

27.245 gdcm::Series Class Reference

Series.

`#include <gdcmSeries.h>`

Public Member Functions

- Series ()

27.245.1 Detailed Description

Series.

27.245.2 Constructor & Destructor Documentation

27.245.2.1 gdcm::Series::Series ( ) [inline]

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

- gdcmSeries.h

27.246 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

27.246.1 Detailed Description

PS 3.4 Table B.3-1 SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

27.246.2 Constructor & Destructor Documentation

27.246.2.1 gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ( )

27.246.3 Member Function Documentation

27.246.3.1 void gdcm::network::ServiceClassApplicationInformation::Print (std::ostream & *os*) const

27.246.3.2 std::istream& gdcm::network::ServiceClassApplicationInformation::Read (std::istream & *is*)

27.246.3.3 void gdcm::network::ServiceClassApplicationInformation::SetTuple (uint8_t *levelofsupport,* uint8_t *levelofdigitalsig,* uint8_t *elementcoercion*)

27.246.3.4 size_t gdcm::network::ServiceClassApplicationInformation::Size ( ) const

27.246.3.5 const std::ostream& gdcm::network::ServiceClassApplicationInformation::Write (std::ostream & *os*) const

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

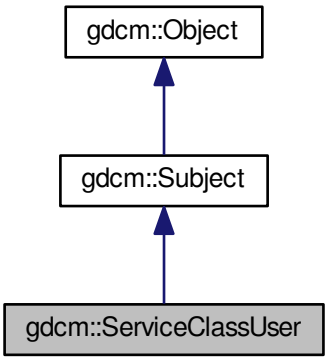
- gdcmServiceClassApplicationInformation.h

27.247 gdcm::ServiceClassUser Class Reference

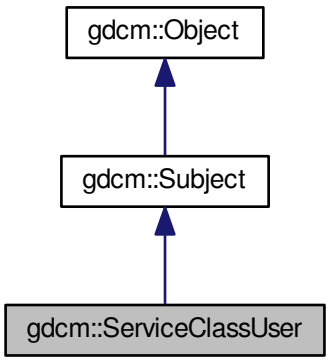
ServiceClassUser.

`#include <gdcmServiceClassUser.h>`

Inheritance diagram for gdcm::ServiceClassUser:



Collaboration diagram for gdcm::ServiceClassUser:



Public Member Functions

- ServiceClassUser ()
- ~ServiceClassUser ()
- 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.
- 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 (File const &file)
- bool SendStore (DataSet const &ds)

  Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.
- 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.

Static Public Member Functions

- static SmartPointer < ServiceClassUser > New ()

  for wrapped language: instanciate a reference counted object

Additional Inherited Members

27.247.1 Detailed Description

ServiceClassUser.

Examples:

CStoreQtProgress.cxx.

27.247.2 Constructor & Destructor Documentation

27.247.2.1 gdcm::ServiceClassUser::ServiceClassUser ( )

Construct a SCU with default:

- hostname = localhost
- port = 104

27.247.2.2 gdcm::ServiceClassUser::~ServiceClassUser ( )

27.247.3 Member Function Documentation

27.247.3.1 const char* gdcm::ServiceClassUser::GetAETitle ( ) const

27.247.3.2 const char* gdcm::ServiceClassUser::GetCalledAETitle ( ) const

27.247.3.3 double gdcm::ServiceClassUser::GetTimeout ( ) const

27.247.3.4 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.

27.247.3.5 bool gdcm::ServiceClassUser::IsPresentationContextAccepted (const PresentationContext & *pc*) const

Return if the passed in presentation was accepted during association negotiation.

27.247.3.6 static SmartPointer<ServiceClassUser> gdcm::ServiceClassUser::New ( ) [inline], [static]

for wrapped language: instanciate a reference counted object

27.247.3.7 bool gdcm::ServiceClassUser::SendEcho ( )

C-ECHO.

27.247.3.8 bool gdcm::ServiceClassUser::SendFind (const BaseRootQuery * *query*, std::vector< DataSet > & *retDatasets*)

C-FIND a query, return result are in retDatasets.

27.247.3.9 bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * *query*, const char * *outputdir*)

Execute a C-MOVE, based on query, return files are written in outputdir.

27.247.3.10 bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * *query*, std::vector< DataSet > & *retDatasets*)

Execute a C-MOVE, based on query, returned dataset are Implicit.

27.247.3.11 bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * *query*, std::vector< File > & *retFile*)

Execute a C-MOVE, based on query, returned Files are stored in vector.

27.247.3.12 bool gdcm::ServiceClassUser::SendStore (const char * *filename*)

Execute a C-STORE on file on disk, named filename.

Examples:

CStoreQtProgress.cxx.

27.247.3.13 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.

27.247.3.14 bool gdcm::ServiceClassUser::SendStore (DataSet const & *ds*)

Execute a C-STORE on a DataSet, the transfer syntax used will be Implicit.

27.247.3.15 void gdcm::ServiceClassUser::SetAETitle (const char * *aetitle*)

set calling ae title

27.247.3.16 void gdcm::ServiceClassUser::SetCalledAETitle (const char * *aetitle*)

set called ae title

Examples:

CStoreQtProgress.cxx.

27.247.3.17 void gdcm::ServiceClassUser::SetHostname (const char * *hostname*)

Set the name of the called hostname (hostname or IP address)

Examples:

CStoreQtProgress.cxx.

27.247.3.18 void gdcm::ServiceClassUser::SetPort (uint16_t *port*)

Set port of remote host (called application)

Examples:

CStoreQtProgress.cxx.

27.247.3.19 void gdcm::ServiceClassUser::SetPortSCP (uint16_t *portscp*)

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

27.247.3.20 void gdcm::ServiceClassUser::SetPresentationContexts (std::vector< PresentationContext > const & *pcs*)

Set the Presentation Context used for the Association.

Examples:

CStoreQtProgress.cxx.

27.247.3.21 void gdcm::ServiceClassUser::SetTimeout (double *t*)

set/get Timeout

Examples:

CStoreQtProgress.cxx.

27.247.3.22 bool gdcm::ServiceClassUser::StartAssociation ( )

Start the association. Need to call SetPresentationContexts before.

Examples:

CStoreQtProgress.cxx.

27.247.3.23 bool gdcm::ServiceClassUser::StopAssociation ( )

Stop the running association.

Examples:

CStoreQtProgress.cxx.

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

- gdcmServiceClassUser.h

27.248 gdcm::SHA1 Class Reference

Class for SHA1.

`#include <gdcmSHA1.h>`

Public Member Functions

- SHA1 ()
- ~SHA1 ()

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])

27.248.1 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

27.248.2 Constructor & Destructor Documentation

27.248.2.1 gdcm::SHA1::SHA1 ( )

27.248.2.2 gdcm::SHA1::~SHA1 ( )

27.248.3 Member Function Documentation

27.248.3.1 static bool gdcm::SHA1::Compute (const char * *buffer,* unsigned long *buf_len,* char *digest_str[20 *2+1]*) [static]

27.248.3.2 static bool gdcm::SHA1::ComputeFile (const char * *filename,* char *digest_str[20 *2+1]*) [static]

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

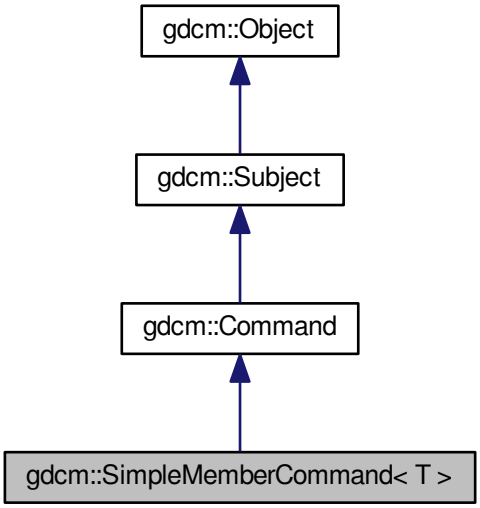
- gdcmSHA1.h

27.249 gdcm::SimpleMemberCommand< T > Class Template Reference

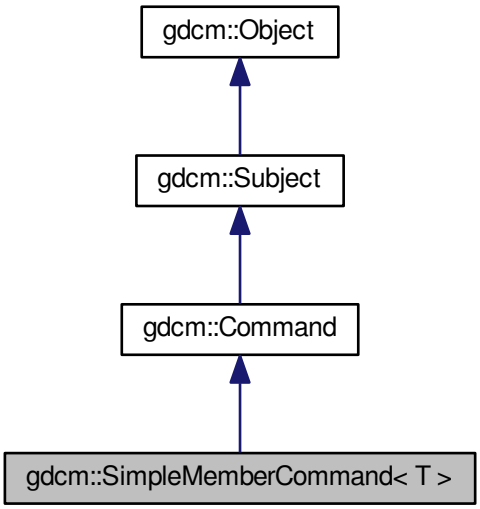
Command subclass that calls a pointer to a member function.

`#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

- virtual void Execute (Subject *, const Event &)
- virtual void Execute (const Subject *, const Event &)
- void SetCallbackFunction (T *object, TMemberFunctionPointer memberFunction)

Static Public Member Functions

- static SmartPointer
  < SimpleMemberCommand > New ()

Protected Member Functions

- SimpleMemberCommand ()
- virtual ~SimpleMemberCommand ()

Protected Attributes

- TMemberFunctionPointer m_MemberFunction
- T * m_This

27.249.1 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.

27.249.2 Member Typedef Documentation

27.249.2.1 template<typename T > typedef SimpleMemberCommand gdcm::SimpleMemberCommand< T >::Self

Standard class typedefs.

27.249.2.2 template<typename T > typedef void(T::* gdcm::SimpleMemberCommand< T >::TMemberFunctionPointer)()

A method callback.

27.249.3 Constructor & Destructor Documentation

27.249.3.1 template<typename T > gdcm::SimpleMemberCommand< T >::SimpleMemberCommand ( ) [inline], [protected]

Referenced by gdcm::SimpleMemberCommand< T >::New().

27.249.3.2 template<typename T > virtual gdcm::SimpleMemberCommand< T >::~SimpleMemberCommand ( ) [inline], [protected], [virtual]

27.249.4 Member Function Documentation

27.249.4.1 template<typename T > virtual void gdcm::SimpleMemberCommand< T >::Execute (Subject * , const Event &) [inline], [virtual]

Invoke the callback function.

Implements gdcm::Command.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction.

27.249.4.2 template<typename T > virtual void gdcm::SimpleMemberCommand< T >::Execute (const Subject * *caller*, const Event & *event*) [inline], [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.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction.

27.249.4.3 template<typename T > static SmartPointer<SimpleMemberCommand> gdcm::SimpleMemberCommand< T >::New ( ) [inline], [static]

Run-time type information (and related methods). Method for creation through the object factory.

References gdcm::SimpleMemberCommand< T >::SimpleMemberCommand().

27.249.4.4 template<typename T > void gdcm::SimpleMemberCommand< T >::SetCallbackFunction (T * *object*, TMemberFunctionPointer *memberFunction*) [inline]

Specify the callback function.

References gdcm::SimpleMemberCommand< T >::m_MemberFunction, and gdcm::SimpleMemberCommand< T >::m_This.

27.249.5 Member Data Documentation

27.249.5.1 template<typename T > TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction [protected]

Referenced by gdcm::SimpleMemberCommand< T >::Execute(), and gdcm::SimpleMemberCommand< T >::SetCallbackFunction().

27.249.5.2 template<typename T > T* gdcm::SimpleMemberCommand< T >::m_This [protected]

Referenced by gdcm::SimpleMemberCommand< T >::SetCallbackFunction().

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

- gdcmCommand.h

27.250 gdcm::SimpleSubjectWatcher Class Reference

SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events.

`#include <gdcmSimpleSubjectWatcher.h>`

Public Member Functions

- SimpleSubjectWatcher (Subject *s, const char *comment="")
- virtual ~SimpleSubjectWatcher ()

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 ()

27.250.1 Detailed Description

SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events.

Examples:

SimpleScanner.cxx.

27.250.2 Constructor & Destructor Documentation

27.250.2.1 gdcm::SimpleSubjectWatcher::SimpleSubjectWatcher (Subject * *s*, const char * *comment* = " ")

27.250.2.2 virtual gdcm::SimpleSubjectWatcher::~SimpleSubjectWatcher ( ) [virtual]

27.250.3 Member Function Documentation

27.250.3.1 virtual void gdcm::SimpleSubjectWatcher::EndFilter ( ) [protected], [virtual]

27.250.3.2 virtual void gdcm::SimpleSubjectWatcher::ShowAbort ( ) [protected], [virtual]

27.250.3.3 virtual void gdcm::SimpleSubjectWatcher::ShowAnonymization (Subject * *caller,* const Event & *evt*) [protected], [virtual]

27.250.3.4 virtual void gdcm::SimpleSubjectWatcher::ShowData (Subject * *caller,* const Event & *evt*) [protected], [virtual]

27.250.3.5 virtual void gdcm::SimpleSubjectWatcher::ShowDataSet (Subject * *caller,* const Event & *evt*) [protected], [virtual]

27.250.3.6 virtual void gdcm::SimpleSubjectWatcher::ShowFileName (Subject * *caller,* const Event & *evt*) [protected], [virtual]

Examples:

SimpleScanner.cxx.

27.250.3.7 virtual void gdcm::SimpleSubjectWatcher::ShowIteration ( ) [protected], [virtual]

27.250.3.8 virtual void gdcm::SimpleSubjectWatcher::ShowProgress (Subject * *caller,* const Event & *evt*) [protected], [virtual]

27.250.3.9 virtual void gdcm::SimpleSubjectWatcher::StartFilter ( ) [protected], [virtual]

27.250.3.10 void gdcm::SimpleSubjectWatcher::TestAbortOff ( ) [protected]

27.250.3.11 void gdcm::SimpleSubjectWatcher::TestAbortOn ( ) [protected]

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

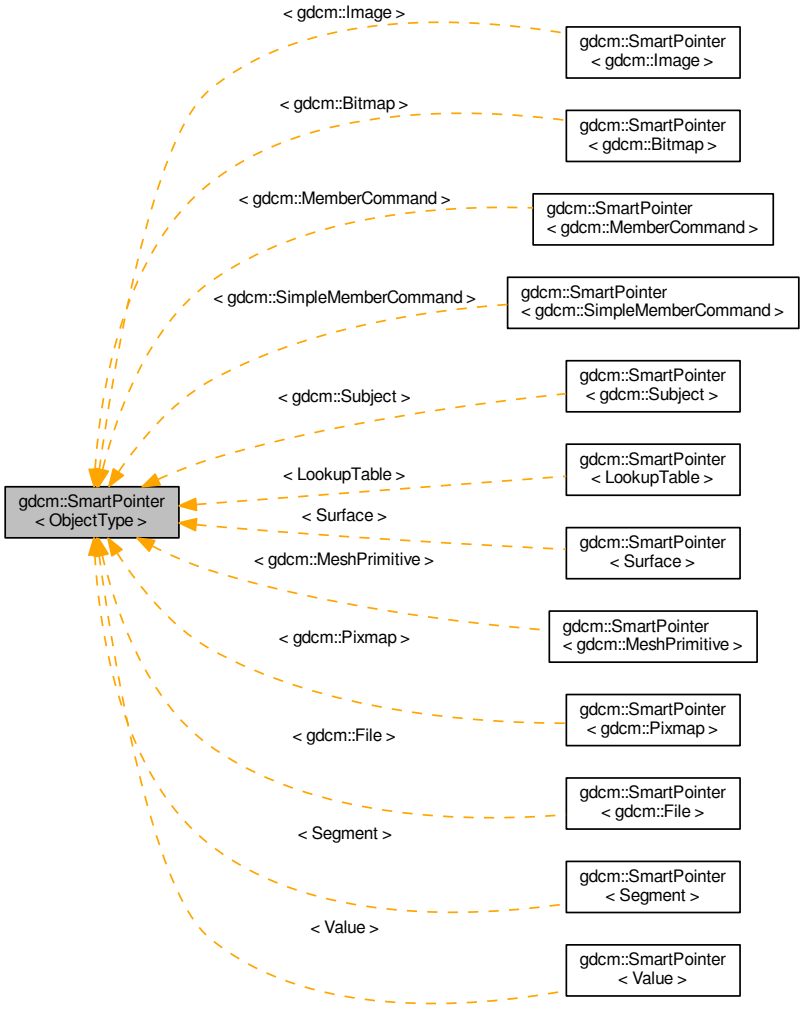
- gdcmSimpleSubjectWatcher.h

27.251 gdcm::SmartPointer< ObjectType > Class Template Reference

Class for Smart Pointer.

`#include <gdcmObject.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= (SmartPointer const &r)

  Overload operator assignment.

- SmartPointer & operator= (ObjectType *r)

  Overload operator assignment.

- SmartPointer & operator= (ObjectType const &r)

27.251.1 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:

ChangeSequenceUltrasound.cxx, CStoreQtProgress.cxx, DumpGEMSMovieGroup.cxx, DumpPhilipsECHO.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, LargeVRDSExplicit.cxx, ReadAndDumpDICOMDIR.cxx, ReadExplicitLengthSQIVR.cxx, and SimpleScanner.cxx.

27.251.2 Constructor & Destructor Documentation

27.251.2.1 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer ( ) [inline]

27.251.2.2 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (const SmartPointer< ObjectType > & *p*) [inline]

27.251.2.3 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (ObjectType * *p*) [inline]

27.251.2.4 template<class ObjectType> gdcm::SmartPointer< ObjectType >::SmartPointer (ObjectType const & *p*) [inline]

27.251.2.5 template<class ObjectType> gdcm::SmartPointer< ObjectType >::~SmartPointer ( ) [inline]

27.251.3 Member Function Documentation

27.251.3.1 template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::GetPointer ( ) const [inline]

Explicit function to retrieve the pointer.

27.251.3.2 template<class ObjectType> gdcm::SmartPointer< ObjectType >::operator ObjectType * ( ) const [inline]

Return pointer to object.

27.251.3.3 template<class ObjectType> ObjectType& gdcm::SmartPointer< ObjectType >::operator* ( ) const [inline]

27.251.3.4 template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::operator-> ( ) const [inline]

Overload operator ->

27.251.3.5 template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (SmartPointer< ObjectType > const & *r*) [inline]

Overload operator assignment.

Referenced by gdcm::SmartPointer< Value >::operator=().

27.251.3.6 template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (ObjectType * *r*) [inline]

Overload operator assignment.

27.251.3.7 template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (ObjectType const & *r*) [inline]

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

- gdcmObject.h
- gdcmSmartPointer.h

27.252 gdcm::network::SOPClassExtendedNegociationSub Class Reference

SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

`#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

27.252.1 Detailed Description

SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

27.252.2 Constructor & Destructor Documentation

27.252.2.1 gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ( )

27.252.3 Member Function Documentation

27.252.3.1 void gdcm::network::SOPClassExtendedNegociationSub::Print (std::ostream & *os*) const

27.252.3.2 std::istream& gdcm::network::SOPClassExtendedNegociationSub::Read (std::istream & *is*)

27.252.3.3 void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (const char * *uid*, uint8_t *levelofsupport* = 3, uint8_t *levelofdigitalsig* = 0, uint8_t *elementcoercion* = 2)

27.252.3.4 size_t gdcm::network::SOPClassExtendedNegociationSub::Size ( ) const

27.252.3.5 const std::ostream& gdcm::network::SOPClassExtendedNegociationSub::Write (std::ostream & *os*) const

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

- gdcmSOPClassExtendedNegociationSub.h

27.253 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 ()

27.253.1 Detailed Description

Class convert a class SOP Class UID into IOD.

Reference PS 3.4 Table B.5-1 STANDARD SOP CLASSES

27.253.2 Member Typedef Documentation

27.253.2.1 typedef const char* gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]

27.253.3 Member Function Documentation

27.253.3.1 static 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.

27.253.3.2 static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (const char * *sopclassuid*) [static]

27.253.3.3 static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( ) [static]

Return the number of SOP Class UID listed internally.

27.253.3.4 static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (const char * *iod*) [static]

27.253.3.5 static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (unsigned int *i*) [static]

27.253.3.6 static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( ) [static]

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

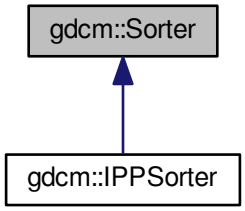
- gdcmSOPClassUIDToIOD.h

27.254 gdcm::Sorter Class Reference

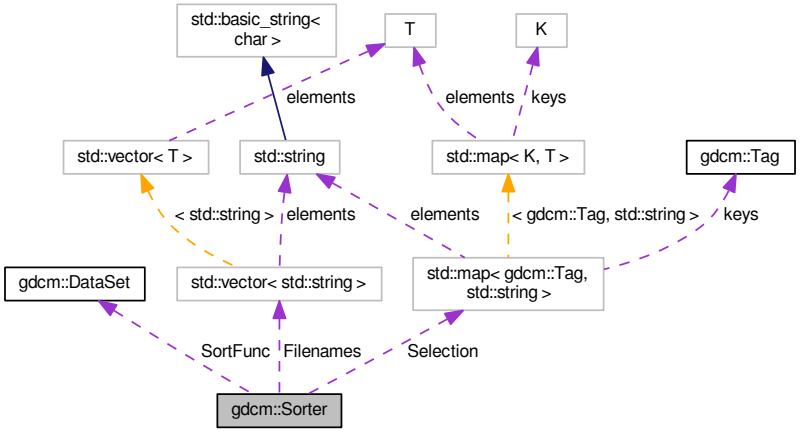
Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction.

`#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)
- 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

Friends

- std::ostream & operator<< (std::ostream &_os, const Sorter &s)

27.254.1 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 everytime you call Sort, all files specified as input paramater are *read*

See also

Scanner

Examples:

SortImage.cxx, and VolumeSorter.cxx.

27.254.2 Member Typedef Documentation

27.254.2.1 typedef std::map<Tag,std::string> gdcm::Sorter::SelectionMap [protected]

27.254.2.2 typedef bool(* gdcm::Sorter::SortFunction)(DataSet const &, DataSet const &)

Set the sort function which compares one dataset to the other.

27.254.3 Constructor & Destructor Documentation

27.254.3.1 gdcm::Sorter::Sorter ( )

27.254.3.2 virtual gdcm::Sorter::~Sorter ( ) [virtual]

27.254.4 Member Function Documentation

27.254.4.1 bool gdcm::Sorter::AddSelect (Tag const & *tag,* const char * *value*)

UNSUPPORTED FOR NOW.

27.254.4.2 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, gdcmorthoplanes.cxx, reslicesphere.cxx, SortImage.cxx, and VolumeSorter.cxx.

27.254.4.3 void gdcm::Sorter::Print (std::ostream & *os*) const

Print.

Examples:

gdcmorthoplanes.cxx, SortImage.cxx, and VolumeSorter.cxx.

Referenced by gdcm::operator<<().

27.254.4.4 void gdcm::Sorter::SetSortFunction (SortFunction *f*)

Examples:

SortImage.cxx, and VolumeSorter.cxx.

27.254.4.5 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.

27.254.4.6 virtual bool gdcm::Sorter::StableSort (std::vector< std::string > const & *filenames*) [virtual]

Examples:

SortImage.cxx, and VolumeSorter.cxx.

27.254.5 Friends And Related Function Documentation

27.254.5.1 std::ostream& operator<< (std::ostream & *_os,* const Sorter & *s*) [friend]

27.254.6 Member Data Documentation

27.254.6.1 std::vector<std::string> gdcm::Sorter::Filenames [protected]

27.254.6.2 std::map<Tag,std::string> gdcm::Sorter::Selection [protected]

27.254.6.3 SortFunction gdcm::Sorter::SortFunc [protected]

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

- gdcmSorter.h

27.255 gdcm::Spacing Class Reference

Class for Spacing.

`#include <gdcmSpacing.h>`

Public Types

- enum SpacingType {
  DETECTOR = 0,
  MAGNIFIED,
  CALIBRATED,
  UNKNOWN }

Public Member Functions

- Spacing ()
- ~Spacing ()

Static Public Member Functions

- static Attribute< 0x28, 0x34 > ComputePixelAspectRatioFromPixelSpacing (const Attribute< 0x28, 0x30 > &pixelspacing)

27.255.1 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

27.255.2 Member Enumeration Documentation

27.255.2.1 enum gdcm::Spacing::SpacingType

Enumerator

DETECTOR

MAGNIFIED

CALIBRATED

UNKNOWN

27.255.3 Constructor & Destructor Documentation

27.255.3.1 gdcm::Spacing::Spacing ( )

27.255.3.2 gdcm::Spacing::~Spacing ( )

27.255.4 Member Function Documentation

27.255.4.1 static 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

27.256 gdcm::Spectroscopy Class Reference

Spectroscopy class.

`#include <gdcmSpectroscopy.h>`

Public Member Functions

- Spectroscopy ()

27.256.1 Detailed Description

Spectroscopy class.

27.256.2 Constructor & Destructor Documentation

27.256.2.1 gdcm::Spectroscopy::Spectroscopy ( ) [inline]

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

- gdcmSpectroscopy.h

27.257 gdcm::SplitMosaicFilter Class Reference

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.

`#include <gdcmSplitMosaicFilter.h>`

Public Member Functions

- SplitMosaicFilter ()
- ~SplitMosaicFilter ()
- bool ComputeMOSAICDimensions (unsigned int dims[3])
- File & GetFile ()
- const File & GetFile () const
- const Image & GetImage () const
- Image & GetImage ()
- void SetFile (const File &f)
- void SetImage (const Image &image)
- bool Split ()

  Split the SIEMENS MOSAIC image.

27.257.1 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.

27.257.2 Constructor & Destructor Documentation

27.257.2.1 gdcm::SplitMosaicFilter::SplitMosaicFilter ( )

27.257.2.2 gdcm::SplitMosaicFilter::~SplitMosaicFilter ( )

27.257.3 Member Function Documentation

27.257.3.1 bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (unsigned int *dims[3]*)

Compute the new dimensions according to private information stored in the MOSAIC header.

27.257.3.2 File& gdcm::SplitMosaicFilter::GetFile ( ) [inline]

27.257.3.3 const File& gdcm::SplitMosaicFilter::GetFile ( ) const [inline]

27.257.3.4 const Image& gdcm::SplitMosaicFilter::GetImage ( ) const [inline]

27.257.3.5 Image& gdcm::SplitMosaicFilter::GetImage ( ) [inline]

27.257.3.6 void gdcm::SplitMosaicFilter::SetFile (const File & *f*) [inline]

27.257.3.7 void gdcm::SplitMosaicFilter::SetImage (const Image & *image*)

27.257.3.8 bool gdcm::SplitMosaicFilter::Split ( )

Split the SIEMENS MOSAIC image.

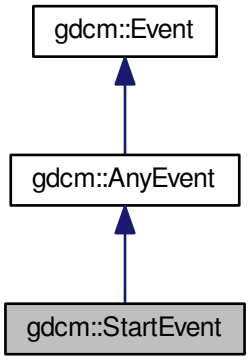
The documentation for this class was generated from the following file:

- gdcmSplitMosaicFilter.h

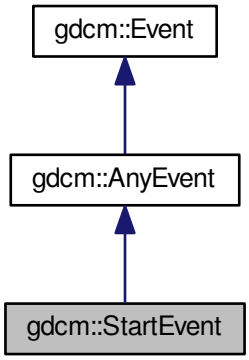
27.258 gdcm::StartEvent Class Reference

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

Inheritance diagram for gdcm::StartEvent:



Collaboration diagram for gdcm::StartEvent:



Additional Inherited Members

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

- gdcmEvent.h

27.259 gdcm::static_assert_test< x > Struct Template Reference

`#include <gdcmStaticAssert.h>`

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

- gdcmStaticAssert.h

27.260 gdcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference

`#include <gdcmStaticAssert.h>`

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

- gdcmStaticAssert.h

27.261 gdcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference

`#include <gdcmStaticAssert.h>`

Public Types

- enum { value = 1 }

27.261.1 Member Enumeration Documentation

27.261.1.1 anonymous enum

Enumerator

value

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

- gdcmStaticAssert.h

27.262 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)

27.262.1 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:

StreamImageReaderTest.cxx.

27.262.2 Constructor & Destructor Documentation

27.262.2.1 gdcm::StreamImageReader::StreamImageReader ( )

27.262.2.2 virtual gdcm::StreamImageReader::~StreamImageReader ( ) [virtual]

27.262.3 Member Function Documentation

27.262.3.1 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.

27.262.3.2 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:

StreamImageReaderTest.cxx.

27.262.3.3 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:

StreamImageReaderTest.cxx.

27.262.3.4 std::vector<unsigned int> gdcm::StreamImageReader::GetDimensionsValueForResolution (unsigned *int*)

27.262.3.5 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:

StreamImageReaderTest.cxx.

27.262.3.6 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:

StreamImageReaderTest.cxx.

27.262.3.7 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:

StreamImageReaderTest.cxx.

27.262.3.8 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.

27.262.3.9 void gdcm::StreamImageReader::SetStream (std::istream & *inStream*)

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

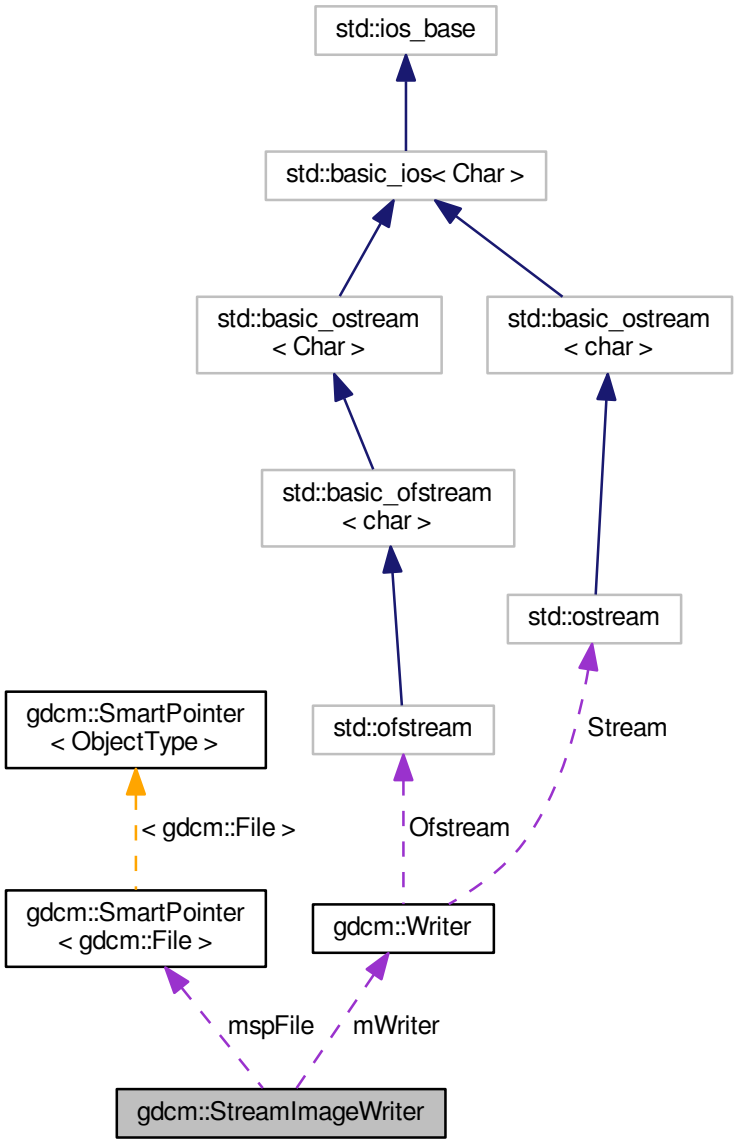
- gdcmStreamImageReader.h

27.263 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

27.263.1 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.

27.263.2 Constructor & Destructor Documentation

27.263.2.1 gdcm::StreamImageWriter::StreamImageWriter ( )

27.263.2.2 virtual gdcm::StreamImageWriter::~StreamImageWriter ( ) [virtual]

27.263.3 Member Function Documentation

27.263.3.1 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.

27.263.3.2 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.

27.263.3.3 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.

27.263.3.4 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.

27.263.3.5 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.

27.263.3.6 void gdcm::StreamImageWriter::SetStream (std::ostream & *inStream*)

Examples:

Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

27.263.3.7 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 accomodate 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.

27.263.3.8 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.

27.263.3.9 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

27.263.3.10 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.

27.263.4 Member Data Documentation

27.263.4.1 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.

27.263.4.2 int gdcm::StreamImageWriter::mElementOffsets1 [protected]

27.263.4.3 SmartPointer<File> gdcm::StreamImageWriter::mspFile [protected]

27.263.4.4 Writer gdcm::StreamImageWriter::mWriter [protected]

27.263.4.5 uint16_t gdcm::StreamImageWriter::mXMax [protected]

27.263.4.6 uint16_t gdcm::StreamImageWriter::mXMin [protected]

27.263.4.7 uint16_t gdcm::StreamImageWriter::mYMax [protected]

27.263.4.8 uint16_t gdcm::StreamImageWriter::mYMin [protected]

27.263.4.9 uint16_t gdcm::StreamImageWriter::mZMax [protected]

27.263.4.10 uint16_t gdcm::StreamImageWriter::mZMin [protected]

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

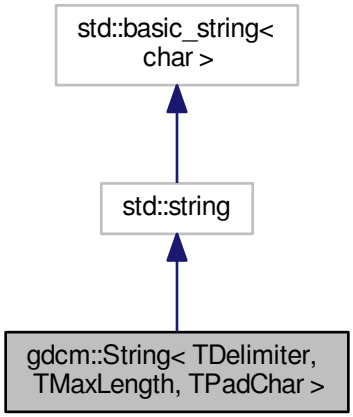
- gdcmStreamImageWriter.h

27.264 gdcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference

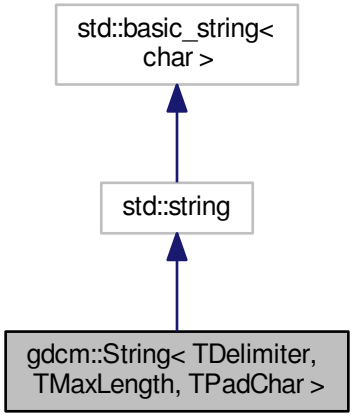
String.

`#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 value_type *s)
- String (const value_type *s, size_type n)
- String (const std::string &s, size_type pos=0, size_type n=npos)
- 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)

27.264.1 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. Noone actually respect the max length TPadChar is the string padding (0 or space)

27.264.2 Member Typedef Documentation

27.264.2.1 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_iterator **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::const_iterator**

27.264.2.2 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reference **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::const_reference**

27.264.2.3 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reverse_iterator **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::const_reverse_iterator**

27.264.2.4 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::difference_type **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::difference_type**

27.264.2.5 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::iterator **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::iterator**

27.264.2.6 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::pointer **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::pointer**

27.264.2.7 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reference **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::reference**

27.264.2.8 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reverse_iterator **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::reverse_iterator**

27.264.2.9 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::size_type **gdcm::String**< TDelimiter, TMaxLength, TPadChar >**::size_type**

27.264.2.10 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type

27.264.3 Constructor & Destructor Documentation

27.264.3.1 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( ) [inline]

String constructors.

27.264.3.2 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * *s*) [inline]

27.264.3.3 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * *s*, size_type *n*) [inline]

27.264.3.4 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]

27.264.4 Member Function Documentation

27.264.4.1 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< TDelimiter, TMaxLength, TPadChar >::Truncate().

27.264.4.2 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:

27.264.4.3 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...

27.264.4.4 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> static std::string gdcm::String< TDelimiter, TMaxLength, TPadChar >::Trim (const char * *input*) [inline], [static]

27.264.4.5 template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String<TDelimiter, TMaxLength, TPadChar> gdcm::String< TDelimiter, TMaxLength, TPadChar >::Truncate ( ) const [inline]

References gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid().

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

- gdcmString.h

27.265 gdcm::StringFilter Class Reference

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.

`#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, VL const &vl)
- 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 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

27.265.1 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:

ReadAndPrintAttributes.cxx.

27.265.2 Constructor & Destructor Documentation

27.265.2.1 gdcm::StringFilter::StringFilter ( )

27.265.2.2 gdcm::StringFilter::~StringFilter ( )

27.265.3 Member Function Documentation

27.265.3.1 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 syntatically correct

27.265.3.2 bool gdcm::StringFilter::ExecuteQuery (std::string const & *query*, DataSet const & *ds*, std::string & *value*) const [protected]

27.265.3.3 std::string gdcm::StringFilter::FromString (const Tag & *t*, const char * *value*, VL const & *vl*)

27.265.3.4 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)

27.265.3.5 File& gdcm::StringFilter::GetFile ( ) [inline]

27.265.3.6 const File& gdcm::StringFilter::GetFile ( ) const [inline]

27.265.3.7 void gdcm::StringFilter::SetDicts (const Dicts & *dicts*)

Allow user to pass in there own dicts.

27.265.3.8 void gdcm::StringFilter::SetFile (const File & *f*) [inline]

Set/Get File.

Examples:

ReadAndPrintAttributes.cxx, and SimplePrintPatientName.cs.

27.265.3.9 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:

ReadAndPrintAttributes.cxx.

27.265.3.10 std::string gdcm::StringFilter::ToString (const Tag & *t*) const

Directly from a Tag:

27.265.3.11 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.

27.265.3.12 std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & *t*) const

Directly from a Tag:

27.265.3.13 std::pair<std::string, std::string> gdcm::StringFilter::ToStringPair (const Tag & *t*, DataSet const & *ds*) const [protected]

27.265.3.14 void gdcm::StringFilter::UseDictAlways (bool) [inline]

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

- gdcmStringFilter.h

27.266 gdcm::Study Class Reference

Study.

`#include <gdcmStudy.h>`

Public Member Functions

- Study ()

27.266.1 Detailed Description

Study.

27.266.2 Constructor & Destructor Documentation

27.266.2.1 gdcm::Study::Study ( ) [inline]

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

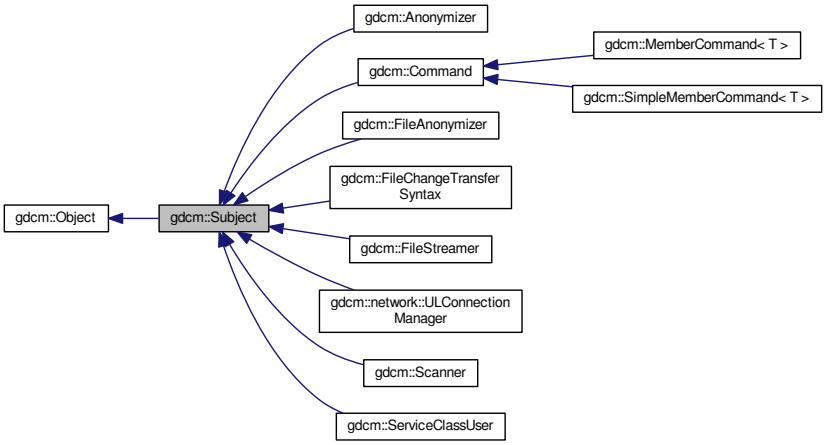
- gdcmStudy.h

27.267 gdcm::Subject Class Reference

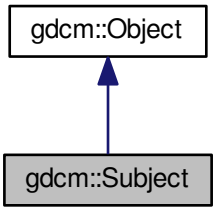
Subject.

`#include <gdcmSubject.h>`

Inheritance diagram for gdcm::Subject:



Collaboration diagram for gdcm::Subject:



Public Member Functions

- Subject ()
- ~Subject ()
- 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)

Additional Inherited Members

27.267.1 Detailed Description

Subject.

See also

Command Event

Examples:

SimpleScanner.cxx.

27.267.2 Constructor & Destructor Documentation

27.267.2.1 gdcm::Subject::Subject ( )

27.267.2.2 gdcm::Subject::~Subject ( )

27.267.3 Member Function Documentation

27.267.3.1 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

27.267.3.2 unsigned long gdcm::Subject::AddObserver (const Event & *event,* Command *) const

27.267.3.3 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 asign 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.

27.267.3.4 bool gdcm::Subject::HasObserver (const Event & *event*) const

Return true if an observer is registered for this event.

27.267.3.5 void gdcm::Subject::InvokeEvent (const Event &)

Call Execute on all the Commands observing this event id.

27.267.3.6 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.

27.267.3.7 void gdcm::Subject::RemoveAllObservers ( )

Remove all observers .

27.267.3.8 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:

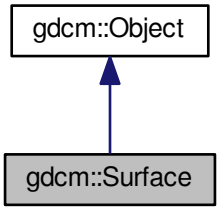
- gdcmSubject.h

27.268 gdcm::Surface Class Reference

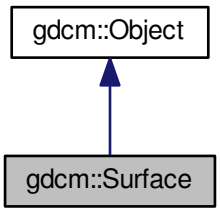
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

`#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 ()
- virtual ~Surface ()
- SegmentHelper::BasicCodedEntry
  const & GetAlgorithmFamily () const
- SegmentHelper::BasicCodedEntry & GetAlgorithmFamily ()
- 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 const & GetMeshPrimitive () const
- MeshPrimitive & GetMeshPrimitive ()
- unsigned long GetNumberOfSurfacePoints () const
- unsigned long GetNumberOfVectors () const
- const DataElement & GetPointCoordinatesData () const

- DataElement & GetPointCoordinatesData ()
- const float * GetPointPositionAccuracy () const
- const float * GetPointsBoundingBoxCoordinates () const
- SegmentHelper::BasicCodedEntry const & GetProcessingAlgorithm () const
- SegmentHelper::BasicCodedEntry & GetProcessingAlgorithm ()
- 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
- const DataElement & GetVectorCoordinateData () const
- DataElement & GetVectorCoordinateData ()
- 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 unsigned short vl[3])
- void SetRecommendedDisplayCIELabValue (const unsigned short vl, const unsigned int idx=0)
- void SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > &vl)
- 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)

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

27.268.1 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

27.268.2 Member Enumeration Documentation

27.268.2.1 enum gdcm::Surface::STATES

Enumerator

NO

YES

UNKNOWN

STATES_END

27.268.2.2 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

27.268.3 Constructor & Destructor Documentation

27.268.3.1 gdcm::Surface::Surface ( )

27.268.3.2 virtual gdcm::Surface::~Surface ( ) [virtual]

27.268.4 Member Function Documentation

27.268.4.1 SegmentHelper::BasicCodedEntry const& gdcm::Surface::GetAlgorithmFamily ( ) const

27.268.4.2 SegmentHelper::BasicCodedEntry& gdcm::Surface::GetAlgorithmFamily ( )

27.268.4.3 const char* gdcm::Surface::GetAlgorithmName ( ) const

27.268.4.4 const char* gdcm::Surface::GetAlgorithmVersion ( ) const

27.268.4.5 const float* gdcm::Surface::GetAxisOfRotation ( ) const

Note

Pointer is null if undefined

27.268.4.6 const float* gdcm::Surface::GetCenterOfRotation ( ) const

Note

Pointer is null if undefined

27.268.4.7 STATES gdcm::Surface::GetFiniteVolume ( ) const

27.268.4.8 STATES gdcm::Surface::GetManifold ( ) const

27.268.4.9 float gdcm::Surface::GetMaximumPointDistance ( ) const

27.268.4.10 float gdcm::Surface::GetMeanPointDistance ( ) const

27.268.4.11 MeshPrimitive const& gdcm::Surface::GetMeshPrimitive ( ) const

27.268.4.12 MeshPrimitive& gdcm::Surface::GetMeshPrimitive ( )

27.268.4.13 unsigned long gdcm::Surface::GetNumberOfSurfacePoints ( ) const

27.268.4.14 unsigned long gdcm::Surface::GetNumberOfVectors ( ) const

27.268.4.15 const DataElement& gdcm::Surface::GetPointCoordinatesData ( ) const

27.268.4.16 DataElement& gdcm::Surface::GetPointCoordinatesData ( )

27.268.4.17 const float* gdcm::Surface::GetPointPositionAccuracy ( ) const

Note

Pointer is null if undefined

27.268.4.18 const float* gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const

Note

Pointer is null if undefined

27.268.4.19 **SegmentHelper::BasicCodedEntry** const& gdcm::Surface::GetProcessingAlgorithm ( ) const

27.268.4.20 **SegmentHelper::BasicCodedEntry**& gdcm::Surface::GetProcessingAlgorithm ( )

27.268.4.21 const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue ( ) const

27.268.4.22 unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (const unsigned int *idx*) const

27.268.4.23 unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const

27.268.4.24 float gdcm::Surface::GetRecommendedPresentationOpacity ( ) const

27.268.4.25 **VIEWType** gdcm::Surface::GetRecommendedPresentationType ( ) const

27.268.4.26 static **STATES** gdcm::Surface::GetSTATES (const char * *state*) [static]

27.268.4.27 static const char* gdcm::Surface::GetSTATESString (**STATES** *state*) [static]

27.268.4.28 const char* gdcm::Surface::GetSurfaceComments ( ) const

27.268.4.29 unsigned long gdcm::Surface::GetSurfaceNumber ( ) const

27.268.4.30 bool gdcm::Surface::GetSurfaceProcessing ( ) const

27.268.4.31 const char* gdcm::Surface::GetSurfaceProcessingDescription ( ) const

27.268.4.32 float gdcm::Surface::GetSurfaceProcessingRatio ( ) const

27.268.4.33 const float* gdcm::Surface::GetVectorAccuracy ( ) const

27.268.4.34 const **DataElement**& gdcm::Surface::GetVectorCoordinateData ( ) const

27.268.4.35 **DataElement**& gdcm::Surface::GetVectorCoordinateData ( )

27.268.4.36 unsigned short gdcm::Surface::GetVectorDimensionality ( ) const

27.268.4.37 static **VIEWType** gdcm::Surface::GetVIEWType (const char * *type*) [static]

27.268.4.38 static const char* gdcm::Surface::GetVIEWTypeString (**VIEWType** *type*) [static]

27.268.4.39 void gdcm::Surface::SetAlgorithmFamily (**SegmentHelper::BasicCodedEntry** const & *BSE*)

27.268.4.40 void gdcm::Surface::SetAlgorithmName (const char * *str*)

27.268.4.41 void gdcm::Surface::SetAlgorithmVersion (const char * *str*)

27.268.4.42 void gdcm::Surface::SetAxisOfRotation (const float * *axis*)

27.268.4.43 void gdcm::Surface::SetCenterOfRotation (const float * *center*)

27.268.4.44 void gdcm::Surface::SetFiniteVolume (**STATES** *state*)

27.268.4.45 void gdcm::Surface::SetManifold (**STATES** *state*)

27.268.4.46 void gdcm::Surface::SetMaximumPointDistance (float *maximum*)

27.268.4.47 void gdcm::Surface::SetMeanPointDistance (float *average*)

27.268.4.48 void gdcm::Surface::SetMeshPrimitive (**MeshPrimitive** & *mp*)

27.268.4.49 void gdcm::Surface::SetNumberOfSurfacePoints (const unsigned long *nb*)

27.268.4.50 void gdcm::Surface::SetNumberOfVectors (const unsigned long *nb*)

27.268.4.51 void gdcm::Surface::SetPointCoordinatesData (**DataElement** const & *de*)

27.268.4.52 void gdcm::Surface::SetPointPositionAccuracy (const float * *accuracies*)

27.268.4.53 void gdcm::Surface::SetPointsBoundingBoxCoordinates (const float * *coordinates*)

27.268.4.54 void gdcm::Surface::SetProcessingAlgorithm (**SegmentHelper::BasicCodedEntry** const & *BSE*)

27.268.4.55 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl[3]*)

27.268.4.56 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl,* const unsigned int *idx* = 0)

27.268.4.57 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > & *vl*)

27.268.4.58 void gdcm::Surface::SetRecommendedDisplayGrayscaleValue (const unsigned short *vl*)

27.268.4.59 void gdcm::Surface::SetRecommendedPresentationOpacity (const float *opacity*)

27.268.4.60 void gdcm::Surface::SetRecommendedPresentationType (**VIEWType** *type*)

27.268.4.61 void gdcm::Surface::SetSurfaceComments (const char * *comment*)

27.268.4.62 void gdcm::Surface::SetSurfaceNumber (const unsigned long *nb*)

27.268.4.63 void gdcm::Surface::SetSurfaceProcessing (bool *b*)

27.268.4.64 void gdcm::Surface::SetSurfaceProcessingDescription (const char * *description*)

27.268.4.65 void gdcm::Surface::SetSurfaceProcessingRatio (const float *ratio*)

27.268.4.66 void gdcm::Surface::SetVectorAccuracy (const float * *accuracy*)

27.268.4.67 void gdcm::Surface::SetVectorCoordinateData (**DataElement** const & *de*)

27.268.4.68 void gdcm::Surface::SetVectorDimensionality (const unsigned short *dim*)

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

- gdcmSurface.h

27.269 gdcm::SurfaceHelper Class Reference

SurfaceHelper Helper class for Surface object.

`#include <gdcmSurfaceHelper.h>`

Public Types

- typedef std::vector< unsigned short > ColorArray

Static Public Member Functions

- template<typename T , typename U >
  static std::vector< T > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U rangeMax=255)
  Convert a DICOM CIE-Lab (after reading) color into RGB.
- template<typename U >
  static std::vector< float > RecommendedDisplayCIELabToRGB (const ColorArray &CIELab, const U rangeMax=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 rangeMax=255)
  Convert a RGB color into DICOM grayscale (ready to write).

27.269.1 Detailed Description

SurfaceHelper Helper class for Surface object.

27.269.2 Member Typedef Documentation

27.269.2.1 typedef std::vector< unsigned short > gdcm::SurfaceHelper::ColorArray

27.269.3 Member Function Documentation

27.269.3.1 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.

27.269.3.2 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.

27.269.3.3 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.

27.269.3.4 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:

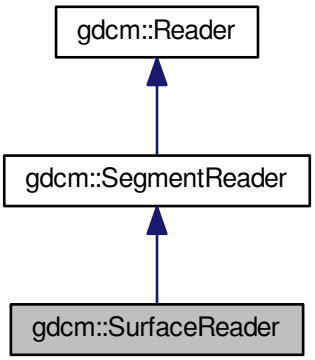
- gdcmSurfaceHelper.h

27.270 gdcm::SurfaceReader Class Reference

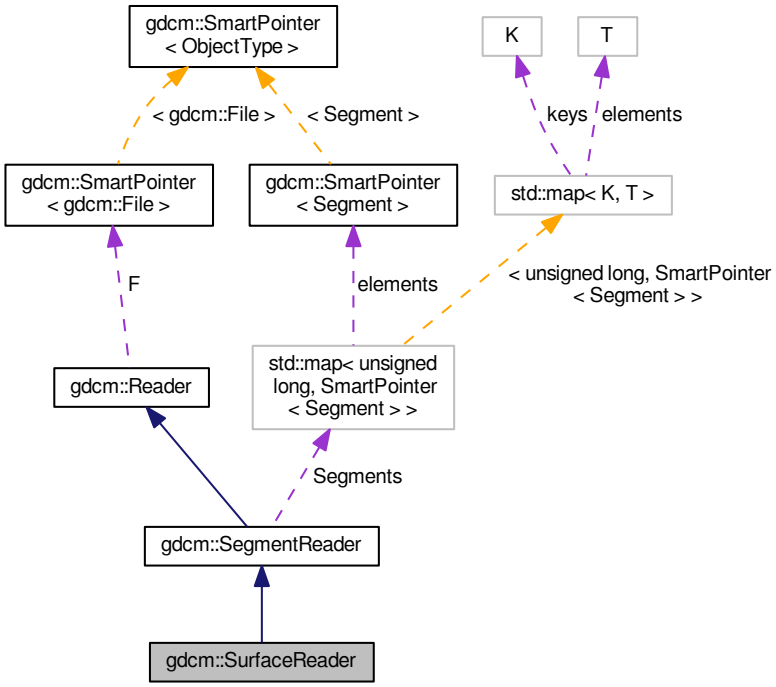
This class defines a SURFACE IE reader. It reads surface mesh module attributes.

`#include <gdcmSurfaceReader.h>`

Inheritance diagram for gdcm::SurfaceReader:



Collaboration diagram for gdcm::SurfaceReader:



Public Member Functions

- SurfaceReader ()
- virtual ~SurfaceReader ()
- unsigned long GetNumberOfSurfaces () const
- virtual bool Read ()

  Read.

Protected Member Functions

- bool ReadPointMacro (SmartPointer< Surface > surface, const DataSet &surfaceDS)
- bool ReadSurface (const Item &surfaceItem, const unsigned long idx)
- bool ReadSurfaces ()

Additional Inherited Members

27.270.1 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

27.270.2 Constructor & Destructor Documentation

27.270.2.1 gdcm::SurfaceReader::SurfaceReader ( )

27.270.2.2 virtual gdcm::SurfaceReader::~SurfaceReader ( ) [virtual]

27.270.3 Member Function Documentation

27.270.3.1 unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ( ) const

27.270.3.2 virtual bool gdcm::SurfaceReader::Read ( ) [virtual]

Read.

Reimplemented from gdcm::SegmentReader.

27.270.3.3 bool gdcm::SurfaceReader::ReadPointMacro (SmartPointer< Surface > *surface,* const DataSet & *surfaceDS*) [protected]

27.270.3.4 bool gdcm::SurfaceReader::ReadSurface (const Item & *surfaceItem,* const unsigned long *idx*) [protected]

27.270.3.5 bool gdcm::SurfaceReader::ReadSurfaces ( ) [protected]

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

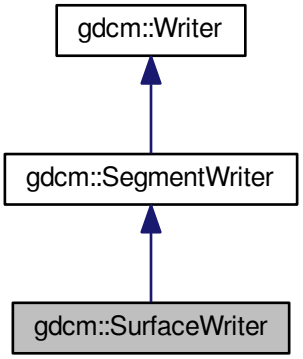
- gdcmSurfaceReader.h

27.271 gdcm::SurfaceWriter Class Reference

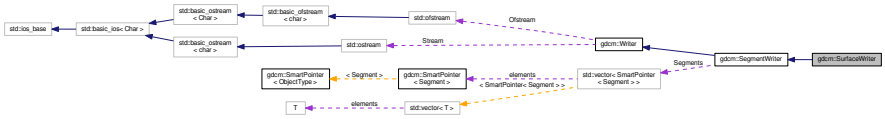
This class defines a SURFACE IE writer. It writes surface mesh module attributes.

`#include <gdcmSurfaceWriter.h>`

Inheritance diagram for gdcm::SurfaceWriter:



Collaboration diagram for gdcm::SurfaceWriter:



Public Member Functions

- SurfaceWriter ()
- virtual ~SurfaceWriter ()
- unsigned long GetNumberOfSurfaces ()
- void SetNumberOfSurfaces (const unsigned long nb)
- bool Write ()

  Write.

Protected Member Functions

- void ComputeNumberOfSurfaces ()
- bool PrepareWrite ()
- bool PrepareWritePointMacro (SmartPointer< Surface > surface, DataSet &surfaceDS, const TransferSyntax &ts)

Protected Attributes

- unsigned long NumberOfSurfaces

Additional Inherited Members

27.271.1 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

27.271.2 Constructor & Destructor Documentation

27.271.2.1 gdcm::SurfaceWriter::SurfaceWriter ( )

27.271.2.2 virtual gdcm::SurfaceWriter::~SurfaceWriter ( ) [virtual]

27.271.3 Member Function Documentation

27.271.3.1 void gdcm::SurfaceWriter::ComputeNumberOfSurfaces ( ) [protected]

27.271.3.2 unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )

27.271.3.3 bool gdcm::SurfaceWriter::PrepareWrite ( ) [protected]

27.271.3.4 bool gdcm::SurfaceWriter::PrepareWritePointMacro (SmartPointer< Surface > *surface*, DataSet & *surfaceDS*, const TransferSyntax & *ts*) [protected]

27.271.3.5 void gdcm::SurfaceWriter::SetNumberOfSurfaces (const unsigned long *nb*)

27.271.3.6 bool gdcm::SurfaceWriter::Write ( ) [virtual]

Write.

Reimplemented from gdcm::SegmentWriter.

27.271.4 Member Data Documentation

27.271.4.1 unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]

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

- gdcmSurfaceWriter.h

27.272 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)

27.272.1 Detailed Description

SwapCode representation.

Examples:

TestByteSwap.cxx.

27.272.2 Member Enumeration Documentation

27.272.2.1 enum gdcm::SwapCode::SwapCodeType

Enumerator

Unknown

LittleEndian

BigEndian

BadLittleEndian

BadBigEndian

27.272.3 Constructor & Destructor Documentation

27.272.3.1 gdcm::SwapCode::SwapCode (SwapCodeType *sc* = Unknown) [inline]

27.272.4 Member Function Documentation

27.272.4.1 static int gdcm::SwapCode::GetIndex (SwapCode const & *sc*) [static], [protected]

27.272.4.2 static const char* gdcm::SwapCode::GetSwapCodeString (SwapCode const & *sc*) [static]

Referenced by gdcm::operator<<().

27.272.4.3 gdcm::SwapCode::operator SwapCode::SwapCodeType ( ) const [inline]

27.272.5 Friends And Related Function Documentation

27.272.5.1 std::ostream& operator<< (std::ostream & *os*, const SwapCode & *sc*) [friend]

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

- gdcmSwapCode.h

27.273 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)

27.273.1 Member Function Documentation

27.273.1.1 template<typename T > static T gdcm::SwapperDoOp::Swap (T *val*) [static]

Referenced by gdcm::Item::Read().

27.273.1.2 template<typename T > static void gdcm::SwapperDoOp::SwapArray (T * *array*, size_t *n*) [inline], [static]

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

- gdcmSwapper.h

27.274 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)

27.274.1 Detailed Description

Examples:

ReadExplicitLengthSQIVR.cxx.

27.274.2 Member Function Documentation

27.274.2.1 template<typename T > static T gdcm::SwapperNoOp::Swap (T *val*) [inline], [static]

Referenced by gdcm::EncodingImplementation< VR::VRBINARY >::Write().

27.274.2.2 template<typename T > static 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

27.275 gdcm::System Class Reference

Class to do system operation.

`#include <gdcmSystem.h>`

Static Public Member Functions

- 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 sytem.
- 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)

27.275.1 Detailed Description

Class to do system operation.

OS independent functionalities

27.275.2 Member Function Documentation

27.275.2.1 static bool gdcm::System::DeleteDirectory (const char * *source*) [static]

remove a directory named source

27.275.2.2 static 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

27.275.2.3 static bool gdcm::System::FileExists (const char * *filename*) [static]

Check whether the specified file exist on the sytem.

Examples:

EncapsulateFileInRawData.cxx, gdcmorthoplanes.cxx, and MagnifyFile.cxx.

27.275.2.4 static bool gdcm::System::FileIsDirectory (const char * *name*) [static]

Check whether the file specified is a directory:

Examples:

gdcmorthoplanes.cxx, and threadgdcm.cxx.

27.275.2.5 static bool gdcm::System::FileIsSymlink (const char * *name*) [static]

Check whether name is a symlink.

27.275.2.6 static 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.

27.275.2.7 static time_t gdcm::System::FileTime (const char * *filename*) [static]

Return the time of last modification of file 0 if the file does not exist

27.275.2.8 static 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]

27.275.2.9 static 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

27.275.2.10 static const char* gdcm::System::GetCurrentModuleFileName ( ) [static]

Return the directory the current module is located: NOT THREAD SAFE

27.275.2.11 static const char* gdcm::System::GetCurrentProcessFileName ( ) [static]

Return the directory the current process (executable) is located: NOT THREAD SAFE

27.275.2.12 static const char* gdcm::System::GetCurrentResourcesDirectory ( ) [static]

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

27.275.2.13 static 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

27.275.2.14 static bool gdcm::System::GetHostName (char *hostname[255]*) [static]

Retrieve the hostname, only the first 255 byte are copyied. This may come handy to specify the Station Name

27.275.2.15 static const char* gdcm::System::GetLastSystemError ( ) [static]

Return the last error.

27.275.2.16 static const char* gdcm::System::GetLocaleCharset ( ) [static]

return `locale charmap`

27.275.2.17 static bool gdcm::System::GetPermissions (const char * *file,* unsigned short & *mode*) [static], [protected]

NOT THREAD SAFE.

27.275.2.18 static const char* gdcm::System::GetTimezoneOffsetFromUTC ( ) [static]

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

27.275.2.19 static bool gdcm::System::MakeDirectory (const char * *path*) [static]

Create a directory name path.

27.275.2.20 static 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)

27.275.2.21 static 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

27.275.2.22 static bool gdcm::System::RemoveFile (const char * *source*) [static]

remove a file named source

27.275.2.23 static bool gdcm::System::SetPermissions (const char * *file,* unsigned short *mode*) [static], [protected]

27.275.2.24 static int gdcm::System::StrCaseCmp (const char * *s1,* const char * *s2*) [static]

consistent func for C99 spec of strcasecmp/strncasecmp

27.275.2.25 static int gdcm::System::StrNCaseCmp (const char * *s1,* const char * *s2,* size_t *n*) [static]

Precondition

n != 0

27.275.2.26 static 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

27.275.2.27 static 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

27.276 gdcm::Table Class Reference

Table.

`#include <gdcmTable.h>`

Public Types

- typedef std::map< Tag, TableEntry > MapTableEntry

Public Member Functions

- Table ()
- ~Table ()
- const TableEntry & GetTableEntry (const Tag &tag) const
- void InsertEntry (Tag const &tag, TableEntry const &te)

Friends

- std::ostream & operator<< (std::ostream &_os, const Table &_val)

27.276.1 Detailed Description

Table.

27.276.2 Member Typedef Documentation

27.276.2.1 typedef std::map<Tag, TableEntry> gdcm::Table::MapTableEntry

27.276.3 Constructor & Destructor Documentation

27.276.3.1 gdcm::Table::Table ( ) [inline]

27.276.3.2 gdcm::Table::~Table ( ) [inline]

27.276.4 Member Function Documentation

27.276.4.1 const TableEntry& gdcm::Table::GetTableEntry (const Tag & *tag*) const [inline]

27.276.4.2 void gdcm::Table::InsertEntry (Tag const & *tag*, TableEntry const & *te*) [inline]

27.276.5 Friends And Related Function Documentation

27.276.5.1 std::ostream& operator<< (std::ostream & *_os*, const Table & *_val*) [friend]

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

- gdcmTable.h

27.277 gdcm::TableEntry Class Reference

TableEntry.

`#include <gdcmTableEntry.h>`

Public Member Functions

- TableEntry (const char *attribute=0, Type const &type=Type(), const char *des=0)
- ~TableEntry ()

27.277.1 Detailed Description

TableEntry.

27.277.2 Constructor & Destructor Documentation

27.277.2.1 gdcm::TableEntry::TableEntry (const char * *attribute* = 0, Type const & *type* = Type (), const char * *des* = 0) [inline]

27.277.2.2 gdcm::TableEntry::~TableEntry ( ) [inline]

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

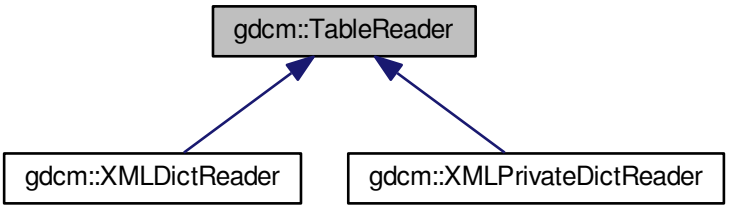
- gdcmTableEntry.h

27.278 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 ()
- 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)

27.278.1 Detailed Description

Class for representing a TableReader.

Note

This class is an empty shell meant to be derived

27.278.2 Constructor & Destructor Documentation

27.278.2.1 gdcm::TableReader::TableReader (Defs & *defs*) [inline]

27.278.2.2 virtual gdcm::TableReader::~TableReader ( ) [inline], [virtual]

27.278.3 Member Function Documentation

27.278.3.1 virtual void gdcm::TableReader::CharacterDataHandler (const char * *data,* int *length*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

27.278.3.2 virtual void gdcm::TableReader::EndElement (const char * *name*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

27.278.3.3 const Defs& gdcm::TableReader::GetDefs ( ) const [inline]

27.278.3.4 const char* gdcm::TableReader::GetFilename ( ) [inline]

27.278.3.5 void gdcm::TableReader::HandleIOD (const char ** *atts*)

27.278.3.6 void gdcm::TableReader::HandleIODEntry (const char ** *atts*)

27.278.3.7 void gdcm::TableReader::HandleMacro (const char ** *atts*)

27.278.3.8 void gdcm::TableReader::HandleMacroEntry (const char ** *atts*)

27.278.3.9 void gdcm::TableReader::HandleMacroEntryDescription (const char ** *atts*)

27.278.3.10 void gdcm::TableReader::HandleModule (const char ** *atts*)

27.278.3.11 void gdcm::TableReader::HandleModuleEntry (const char ** *atts*)

27.278.3.12 void gdcm::TableReader::HandleModuleEntryDescription (const char ** *atts*)

27.278.3.13 void gdcm::TableReader::HandleModuleInclude (const char ** *atts*)

27.278.3.14 int gdcm::TableReader::Read ( )

27.278.3.15 void gdcm::TableReader::SetFilename (const char * *filename*) [inline]

27.278.3.16 virtual void gdcm::TableReader::StartElement (const char * *name,* const char ** *atts*) [virtual]

Reimplemented in gdcm::XMLDictReader, and gdcm::XMLPrivateDictReader.

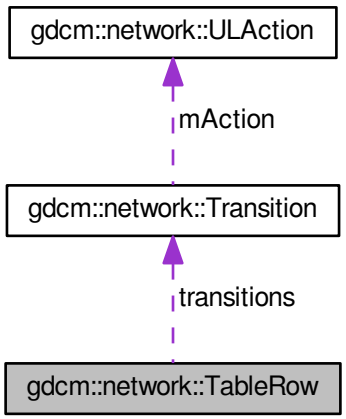
The documentation for this class was generated from the following file:

- gdcmTableReader.h

27.279 gdcm::network::TableRow Class Reference

`#include <gdcmULTransitionTable.h>`

Collaboration diagram for gdcm::network::TableRow:



Public Member Functions

- TableRow ()

- ~TableRow ()

Public Attributes

- Transition * transitions [cMaxStateID]

27.279.1 Constructor & Destructor Documentation

27.279.1.1 gdcm::network::TableRow::TableRow ( ) [inline]

References gdcm::network::cMaxStateID, and transitions.

27.279.1.2 gdcm::network::TableRow::~TableRow ( ) [inline]

References gdcm::network::cMaxStateID, and transitions.

27.279.2 Member Data Documentation

27.279.2.1 Transition* gdcm::network::TableRow::transitions[cMaxStateID]

Referenced by TableRow(), and ~TableRow().

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

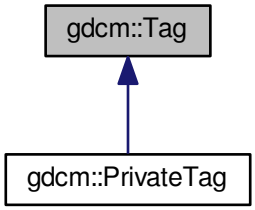
- gdcmULTransitionTable.h

27.280 gdcm::Tag Class Reference

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)

`#include <gdcmTag.h>`

Inheritance diagram for gdcm::Tag:



Public Member Functions

- 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.
- Tag (const Tag &_val)
- 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
- const uint16_t & operator[] (const unsigned int &_id) const

  Returns the Group or Element of the given Tag, depending on id (0/1)
- uint16_t & operator[] (const unsigned int &_id)

  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)

27.280.1 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:

ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DumpToSQLITE3.cxx, DuplicatePCDE.cxx, Encapsulate↩ FileInRawData.cxx, ExtractEncryptedContent.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_↩ Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, gdcmrtionplan.cxx, gdcmrtplan.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetJPEGSamplePrecision.cxx, GetSequenceUltrasound.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, Merge↩ TwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, PublicDict.cxx, ReadAndDumpDICOMDIR.cxx, ReadAndPrint↩ Attributes.cxx, ReadExplicitLengthSQIVR.cxx, rle2img.cxx, SimpleScanner.cxx, SortImage.cxx, StreamImage↩ ReaderTest.cxx, TraverseModules.cxx, and VolumeSorter.cxx.

27.280.2 Constructor & Destructor Documentation

27.280.2.1 gdcm::Tag::Tag (uint16_t *group*, uint16_t *element*) [inline]

Constructor with 2∗uint16_t.

27.280.2.2 gdcm::Tag::Tag (uint32_t *tag* = 0) [inline]

Constructor with 1∗uint32_t Prefer the cstor that takes two uint16_t.

27.280.2.3 gdcm::Tag::Tag (const Tag & *_val*) [inline]

References tag.

27.280.3 Member Function Documentation

27.280.3.1 uint16_t gdcm::Tag::GetElement ( ) const [inline]

Returns the 'Element number' of the given Tag.

Examples:

DuplicatePCDE.cxx, and PublicDict.cxx.

Referenced by gdcm::DataSet::ComputeGroupLength(), IsGroupXX(), gdcm::PrivateDict::PrintXML(), gdcm::PrivateTag::PrivateTag(), gdcm::SequenceOfFragments::ReadValue(), and SetPrivateCreator().

27.280.3.2 uint32_t gdcm::Tag::GetElementTag ( ) const [inline]

Returns the full tag value of the given Tag.

27.280.3.3 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::FileMetaInformation::Insert(), gdcm::DataSet::Insert(), IsGroupXX(), gdcm::PrivateDict::PrintXML(), gdcm::SequenceOfFragments::ReadValue(), gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(), and SetPrivateCreator().

27.280.3.4 uint32_t gdcm::Tag::GetLength ( ) const [inline]

return the length of tag (read: size on disk)

27.280.3.5 Tag gdcm::Tag::GetPrivateCreator ( ) const [inline]

Return the Private Creator Data Element tag of a private data element.

References SetElement().

27.280.3.6 bool gdcm::Tag::IsGroupLength ( ) const [inline]

return whether the tag correspond to a group length tag:

27.280.3.7 bool gdcm::Tag::IsGroupXX (const Tag & *t*) const [inline]

e.g 6002,3000 belong to groupXX: 6000,3000

References GetElement(), GetGroup(), and IsPrivate().

27.280.3.8 bool gdcm::Tag::IsIllegal ( ) const [inline]

return if the tag is considered to be an illegal tag

27.280.3.9 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.

Referenced by IsGroupXX(), and SetPrivateCreator().

27.280.3.10 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.

27.280.3.11 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...

27.280.3.12 bool gdcm::Tag::operator!= (const Tag & *_val*) const [inline]

References tag.

27.280.3.13 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, and tags.

27.280.3.14 bool gdcm::Tag::operator<= (const Tag & *t2*) const [inline]

27.280.3.15 Tag& gdcm::Tag::operator= (const Tag & *_val*) [inline]

References tag.

27.280.3.16 bool gdcm::Tag::operator== (const Tag & *_val*) const [inline]

References tag.

27.280.3.17 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)

27.280.3.18 uint16_t& gdcm::Tag::operator[] (const unsigned int & *_id*) [inline]

Returns the Group or Element of the given Tag, depending on id (0/1)

27.280.3.19 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)

27.280.3.20 std::string gdcm::Tag::PrintAsContinuousUpperCaseString ( ) const

Same as PrintAsContinuousString, but hexadecimal [a-f] are printed using upper case.

27.280.3.21 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

27.280.3.22 template<typename TSwap > std::istream& gdcm::Tag::Read (std::istream & *is*) [inline]

Read a tag from binary representation.

27.280.3.23 bool gdcm::Tag::ReadFromCommaSeparatedString (const char * *str*)

Read from a comma separated string. This is a highly user oriented function, the string should be formated 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

27.280.3.24 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)

27.280.3.25 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

27.280.3.26 void gdcm::Tag::SetElement (uint16_t *element*) [inline]

Sets the 'Element number' of the given Tag.

Examples:

DuplicatePCDE.cxx, and PublicDict.cxx.

Referenced by GetPrivateCreator(), and gdcm::operator>>().

27.280.3.27 void gdcm::Tag::SetElementTag (uint16_t *group,* uint16_t *element*) [inline]

Sets the 'Group number' & 'Element number' of the given Tag.

27.280.3.28 void gdcm::Tag::SetElementTag (uint32_t *tag*) [inline]

Sets the full tag value of the given Tag.

27.280.3.29 void gdcm::Tag::SetGroup (uint16_t *group*) [inline]

Sets the 'Group number' of the given Tag.

Referenced by gdcm::operator>>().

27.280.3.30 void gdcm::Tag::SetPrivateCreator (Tag const & *t*) [inline]

Set private creator:

Examples:

DuplicatePCDE.cxx.

References GetElement(), GetGroup(), and IsPrivate().

27.280.3.31 template<typename TSwap > const std::ostream& gdcm::Tag::Write (std::ostream & *os*) const [inline]

Write a tag in binary rep.

Referenced by gdcm::SequenceOfItems::Write(), gdcm::Item::Write(), and gdcm::SequenceOfFragments::Write().

27.280.4 Friends And Related Function Documentation

27.280.4.1 std::ostream& operator<< (std::ostream & *_os,* const Tag & *_val*) [friend]

27.280.4.2 std::istream& operator>> (std::istream & *_is,* Tag & *_val*) [friend]

27.280.5 Member Data Documentation

27.280.5.1 char gdcm::Tag::bytes[4]

27.280.5.2 uint32_t gdcm::Tag::tag

Referenced by operator!=(), operator<(), operator=(), operator==(), and Tag().

27.280.5.3 uint16_t gdcm::Tag::tags[2]

Referenced by operator<().

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

- gdcmTag.h

27.281 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.

27.281.1 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`

27.281.2 Constructor & Destructor Documentation

27.281.2.1 gdcm::TagPath::TagPath ( )

27.281.2.2 gdcm::TagPath::~TagPath ( )

27.281.3 Member Function Documentation

27.281.3.1 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

27.281.3.2 bool gdcm::TagPath::ConstructFromTagList (Tag const * *l*, unsigned int *n*)

Construct from a list of tags.

27.281.3.3 static bool gdcm::TagPath::IsValid (const char * *path*) [static]

Return if path is valid or not.

27.281.3.4 void gdcm::TagPath::Print (std::ostream &) const

27.281.3.5 bool gdcm::TagPath::Push (Tag const & *t*)

27.281.3.6 bool gdcm::TagPath::Push (unsigned int *itemnum*)

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

- gdcmTagPath.h

27.282 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 ()
- ~Testing ()
- 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, unsigned long 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=0)
- static const wchar_t * GetTempDirectoryW (const wchar_t *subdir=0)

  NOT THREAD SAFE.
- static const char * GetTempFilename (const char *filename, const char *subdir=0)

  NOT THREAD SAFE.
- static const wchar_t * GetTempFilenameW (const wchar_t *filename, const wchar_t *subdir=0)

  NOT THREAD SAFE.

27.282.1 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

27.282.2 Member Typedef Documentation

27.282.2.1 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

27.282.2.2 typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType)[2]

return the table that map the media storage (as string) of a filename (gdcmData)

27.282.3 Constructor & Destructor Documentation

27.282.3.1 gdcm::Testing::Testing ( ) [inline]

27.282.3.2 gdcm::Testing::~Testing ( ) [inline]

27.282.4 Member Function Documentation

27.282.4.1 static bool gdcm::Testing::ComputeFileMD5 (const char * *filename,* char *digest_str[33]*) [static]

Examples:

MetaImageMD5Activiz.cs.

27.282.4.2 static bool gdcm::Testing::ComputeMD5 (const char * *buffer,* unsigned long *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.

27.282.4.3 static const char* gdcm::Testing::GetDataExtraRoot ( ) [static]

Return the GDCM DATA EXTRA ROOT.

Examples:

DiscriminateVolume.cxx, reslicesphere.cxx, and VolumeSorter.cxx.

27.282.4.4 static const char* gdcm::Testing::GetDataRoot ( ) [static]

Return the GDCM DATA ROOT.

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, and MagnifyFile.cxx.

27.282.4.5 static const char* gdcm::Testing::GetFileName (unsigned int *file*) [static]

Examples:

MetaImageMD5Activiz.cs.

27.282.4.6 static const char* const* gdcm::Testing::GetFileNames ( ) [static]

return the table of fullpath to gdcmData DICOM files:

Examples:

TestReader.cxx.

27.282.4.7 static int gdcm::Testing::GetLossyFlagFromFile (const char * *filepath*) [static]

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

27.282.4.8 static const char* const* gdcm::Testing::GetMD5DataImage (unsigned int *file*) [static]

27.282.4.9 static **MD5DataImagesType** gdcm::Testing::GetMD5DataImages ( ) [static]

27.282.4.10 static 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.

27.282.4.11 static const char* gdcm::Testing::GetMD5FromFile (const char * *filepath*) [static]

27.282.4.12 static const char* const* gdcm::Testing::GetMediaStorageDataFile (unsigned int *file*) [static]

27.282.4.13 static **MediaStorageDataFilesType** gdcm::Testing::GetMediaStorageDataFiles ( ) [static]

27.282.4.14 static const char* gdcm::Testing::GetMediaStorageFromFile (const char * *filepath*) [static]

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

27.282.4.15 static unsigned int gdcm::Testing::GetNumberOfFileNames ( ) [static]

Examples:

MetaImageMD5Activiz.cs.

27.282.4.16 static unsigned int gdcm::Testing::GetNumberOfMD5DataImages ( ) [static]

27.282.4.17 static unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ( ) [static]

27.282.4.18 static const char* gdcm::Testing::GetPixelSpacingDataRoot ( ) [static]

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

27.282.4.19 static 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

27.282.4.20 static 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

27.282.4.21 static const char* gdcm::Testing::GetSourceDirectory ( ) [static]

Examples:

BasicAnonymizer.cs, and ClinicalTrialIdentificationWorkflow.cs.

27.282.4.22 static 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

27.282.4.23 static const char* gdcm::Testing::GetTempDirectory (const char * *subdir* = 0) [static]

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

Examples:

MetaImageMD5Activiz.cs.

27.282.4.24 static const wchar_t* gdcm::Testing::GetTempDirectoryW (const wchar_t * *subdir* = 0) [static]

NOT THREAD SAFE.

27.282.4.25 static const char* gdcm::Testing::GetTempFilename (const char * *filename,* const char * *subdir* = 0) [static]

NOT THREAD SAFE.

Examples:

MetaImageMD5Activiz.cs.

27.282.4.26 static const wchar_t* gdcm::Testing::GetTempFilenameW (const wchar_t * *filename,* const wchar_t * *subdir* = 0) [static]

NOT THREAD SAFE.

27.282.4.27 void gdcm::Testing::Print (std::ostream & *os* = `std::cout`)

Print.

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

- gdcmTesting.h

27.283 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)

  Explicitely set the stream which receive Debug messages:
- static void SetError (bool debug)

  Turn error messages on (default: true)
- static void SetErrorStream (std::ostream &os)

  Explicitely 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)

  Explicitely set the stream which receive Warning messages:
- static void WarningOff ()
- static void WarningOn ()

27.283.1 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.

27.283.2 Constructor & Destructor Documentation

27.283.2.1 gdcm::Trace::Trace ( )

27.283.2.2 gdcm::Trace::~Trace ( )

27.283.3 Member Function Documentation

27.283.3.1 static void gdcm::Trace::DebugOff ( ) [static]

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

27.283.3.2 static void gdcm::Trace::DebugOn ( ) [static]

Examples:

Fake_Image_Using_Stream_Image_Writer.cxx, and StreamImageReaderTest.cxx.

27.283.3.3 static void gdcm::Trace::ErrorOff ( ) [static]

Examples:

MetaImageMD5Activiz.cs.

27.283.3.4 static void gdcm::Trace::ErrorOn ( ) [static]

27.283.3.5 static bool gdcm::Trace::GetDebugFlag ( ) [static]

27.283.3.6 static std::ostream& gdcm::Trace::GetDebugStream ( ) [static]

27.283.3.7 static bool gdcm::Trace::GetErrorFlag ( ) [static]

27.283.3.8 static std::ostream& gdcm::Trace::GetErrorStream ( ) [static]

27.283.3.9 static std::ostream& gdcm::Trace::GetStream ( ) [static]

27.283.3.10 static bool gdcm::Trace::GetWarningFlag ( ) [static]

27.283.3.11 static std::ostream& gdcm::Trace::GetWarningStream ( ) [static]

27.283.3.12 static void gdcm::Trace::SetDebug (bool *debug*) [static]

Turn debug messages on (default: false)

Examples:

DumpToSQLITE3.cxx.

27.283.3.13 static void gdcm::Trace::SetDebugStream (std::ostream & *os*) [static]

Explicitely set the stream which receive Debug messages:

27.283.3.14 static void gdcm::Trace::SetError (bool *debug*) [static]

Turn error messages on (default: true)

27.283.3.15 static void gdcm::Trace::SetErrorStream (std::ostream & *os*) [static]

Explicitely set the stream which receive Error messages:

Examples:

CStoreQtProgress.cxx.

27.283.3.16 static void gdcm::Trace::SetStream (std::ostream & *os*) [static]

Explicitely set the ostream for gdcm::Trace to report to This will set the DebugStream, WarningStream and ErrorStream at once:

27.283.3.17 static void gdcm::Trace::SetStreamToFile (const char * *filename*) [static]

Explicitely set the filename for gdcm::Trace to report to The file will be created (it will not append to existing file)

27.283.3.18 static void gdcm::Trace::SetWarning (bool *debug*) [static]

Turn warning messages on (default: true)

Examples:

DumpToSQLITE3.cxx.

27.283.3.19 static void gdcm::Trace::SetWarningStream (std::ostream & *os*) [static]

Explicitely set the stream which receive Warning messages:

27.283.3.20 static void gdcm::Trace::WarningOff ( ) [static]

Examples:

MetaImageMD5Activiz.cs, and TestReader.cxx.

27.283.3.21 static 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

27.284 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,
CT_private_ELE,
JPIPReferenced,
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)

27.284.1 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:

GetJPEGSamplePrecision.cxx, and LargeVRDSExplicit.cxx.

27.284.2 Member Enumeration Documentation

27.284.2.1 enum gdcm::TransferSyntax::NegociatedType

Enumerator

Unknown

Explicit

Implicit

27.284.2.2 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
CT_private_ELE
JPIPReferenced
TS_END

27.284.3 Constructor & Destructor Documentation

27.284.3.1 gdcm::TransferSyntax::TransferSyntax (TSType *type* = ImplicitVRLittleEndian) [inline]

27.284.4 Member Function Documentation

27.284.4.1 bool gdcm::TransferSyntax::CanStoreLossy ( ) const

return true if TransFer Syntax Allow storing of Lossy Pixel Data

27.284.4.2 NegociatedType gdcm::TransferSyntax::GetNegociatedType ( ) const

27.284.4.3 const char* gdcm::TransferSyntax::GetString ( ) const [inline]

References GetTSString().

27.284.4.4 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.

27.284.4.5 static const char* gdcm::TransferSyntax::GetTSString (TSType *ts*) [static]

Examples:

LargeVRDSExplicit.cxx.

Referenced by GetString(), and gdcm::operator<<().

27.284.4.6 static TSType gdcm::TransferSyntax::GetTSType (const char * *str*) [static]

27.284.4.7 bool gdcm::TransferSyntax::IsEncapsulated ( ) const

Examples:

ExtractIconFromFile.cxx.

27.284.4.8 bool gdcm::TransferSyntax::IsEncoded ( ) const

27.284.4.9 bool gdcm::TransferSyntax::IsExplicit ( ) const

27.284.4.10 bool gdcm::TransferSyntax::IsImplicit ( ) const

27.284.4.11 bool gdcm::TransferSyntax::IsLossless ( ) const

Return true if the transfer syntax algorithm is a lossless algorithm

27.284.4.12 bool gdcm::TransferSyntax::IsLossy ( ) const

Return true if the transfer syntax algorithm is a lossy algorithm

27.284.4.13 bool gdcm::TransferSyntax::IsValid ( ) const [inline]

27.284.4.14 gdcm::TransferSyntax::operator TSType ( ) const [inline]

27.284.5 Friends And Related Function Documentation

27.284.5.1 std::ostream& operator<< (std::ostream & *os*, const TransferSyntax & *ts*) [friend]

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

- gdcmTransferSyntax.h

27.285 gdcm::network::TransferSyntaxSub Class Reference

TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

`#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

27.285.1 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

27.285.2 Constructor & Destructor Documentation

27.285.2.1 gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )

27.285.3 Member Function Documentation

27.285.3.1 const char* gdcm::network::TransferSyntaxSub::GetName ( ) const [inline]

27.285.3.2 bool gdcm::network::TransferSyntaxSub::operator== (const TransferSyntaxSub & *ts*) const [inline]

27.285.3.3 void gdcm::network::TransferSyntaxSub::Print (std::ostream & *os*) const

27.285.3.4 std::istream& gdcm::network::TransferSyntaxSub::Read (std::istream & *is*)

27.285.3.5 void gdcm::network::TransferSyntaxSub::SetName (const char * *name*)

27.285.3.6 void gdcm::network::TransferSyntaxSub::SetNameFromUID (UIDs::TSName *tsname*)

27.285.3.7 size_t gdcm::network::TransferSyntaxSub::Size ( ) const

27.285.3.8 const std::ostream& gdcm::network::TransferSyntaxSub::Write (std::ostream & *os*) const

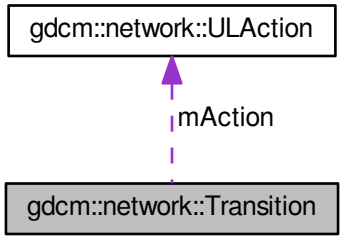
The documentation for this class was generated from the following file:

- gdcmTransferSyntaxSub.h

27.286 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

27.286.1 Constructor & Destructor Documentation

27.286.1.1 gdcm::network::Transition::Transition ( ) [inline]

References gdcm::network::eStaDoesNotExist.

Referenced by MakeNew().

27.286.1.2 gdcm::network::Transition::~Transition ( ) [inline]

References mAction.

27.286.1.3 gdcm::network::Transition::Transition (int *inEndState,* ULAction * *inAction*) [inline]

27.286.2 Member Function Documentation

27.286.2.1 static Transition* gdcm::network::Transition::MakeNew (int *inEndState,* ULAction * *inAction*) [inline], [static]

References Transition().

27.286.3 Member Data Documentation

27.286.3.1 ULAction* gdcm::network::Transition::mAction

Referenced by ~Transition().

27.286.3.2 int gdcm::network::Transition::mEnd

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

- gdcmULTransitionTable.h

27.287 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)

27.287.1 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.

27.287.2 Member Enumeration Documentation

27.287.2.1 enum gdcm::Type::TypeType

Enumerator

T1
T1C
T2
T2C
T3
UNKNOWN

27.287.3 Constructor & Destructor Documentation

27.287.3.1 gdcm::Type::Type (TypeType *type* = UNKNOWN) [inline]

27.287.4 Member Function Documentation

27.287.4.1 static const char* gdcm::Type::GetTypeString (TypeType *type*) [static]

Referenced by gdcm::operator<<().

27.287.4.2 static TypeType gdcm::Type::GetTypeType (const char * *type*) [static]

Referenced by gdcm::ModuleEntry::ModuleEntry().

27.287.4.3 gdcm::Type::operator TypeType ( ) const [inline]

27.287.5 Friends And Related Function Documentation

27.287.5.1 std::ostream& operator<< (std::ostream & *os*, const Type & *vr*) [friend]

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

- gdcmType.h

27.288 gdcm::UI Struct Reference

`#include <gdcmVR.h>`

Public Attributes

- char Internal [64+1]

Friends

- std::ostream & operator<< (std::ostream &_os, const UI &_val)

27.288.1 Friends And Related Function Documentation

27.288.1.1 std::ostream& operator<< (std::ostream & *_os,* const UI & *_val*) [friend]

27.288.2 Member Data Documentation

27.288.2.1 char gdcm::UI::Internal[64+1]

Referenced by gdcm::operator<<().

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

- gdcmVR.h

27.289 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)

27.289.1 Detailed Description

Class for generating unique UID.

Note

bla Usage: When constructing a Series or Study UID, user *has* to keep around the UID, otherwise the UI↩ D 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.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.↩ cxx, StreamImageReaderTest.cxx, and uid_unique.cxx.

27.289.2 Constructor & Destructor Documentation

27.289.2.1 gdcm::UIDGenerator::UIDGenerator ( ) [inline]

By default the root of a UID is a GDCM Root...

27.289.3 Member Function Documentation

27.289.3.1 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.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GetSubSequenceData.↩ cxx, StreamImageReaderTest.cxx, and uid_unique.cxx.

27.289.3.2 static bool gdcm::UIDGenerator::GenerateUUID (unsigned char * *uuid_data*) [static], [protected]

27.289.3.3 static const char* gdcm::UIDGenerator::GetGDCMUID ( ) [static]

Return the default (GDCM) root UID:

27.289.3.4 static const char* gdcm::UIDGenerator::GetRoot ( ) [static]

Examples:

ClinicalTrialIdentificationWorkflow.cs, ReformatFile.cs, and StandardizeFiles.cs.

27.289.3.5 static 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)

27.289.3.6 static 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 garanteeing uniqueness of UIDs when the root is longer than 26 bytes.

Examples:

ClinicalTrialIdentificationWorkflow.cs, ReformatFile.cs, StandardizeFiles.cs, and uid_unique.cxx.

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

- gdcmUIDGenerator.h

27.290 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,
  JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLossless↩

JPEGImageCompression = 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,

BreastTomosynthesisImageStorage }

- 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_5_1_4_1_1_13_1_3 }

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)

27.290.1 Detailed Description

all known uids

Examples:

GenerateStandardSOPClasses.cxx.

27.290.2 Member Typedef Documentation

27.290.2.1 typedef const char* const(* gdcm::UIDs::TransferSyntaxStringsType)[2]

27.290.3 Member Enumeration Documentation

27.290.3.1 enum gdcm::UIDs::TSName

Enumerator

VerificationSOPClass

ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM

ExplicitVRLittleEndian

DeflatedExplicitVRLittleEndian

ExplicitVRBigEndian

JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression

JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only

JPEGExtendedProcess35Retired

JPEGSpectralSelectionNonHierarchicalProcess68Retired

JPEGSpectralSelectionNonHierarchicalProcess79Retired

JPEGFullProgressionNonHierarchicalProcess1012Retired

JPEGFullProgressionNonHierarchicalProcess1113Retired

JPEGLosslessNonHierarchicalProcess14
JPEGLosslessNonHierarchicalProcess15Retired
JPEGExtendedHierarchicalProcess1618Retired
JPEGExtendedHierarchicalProcess1719Retired
JPEGSpectralSelectionHierarchicalProcess2022Retired
JPEGSpectralSelectionHierarchicalProcess2123Retired
JPEGFullProgressionHierarchicalProcess2426Retired
JPEGFullProgressionHierarchicalProcess2527Retired
JPEGLosslessHierarchicalProcess28Retired
JPEGLosslessHierarchicalProcess29Retired
JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImage

JPEGLSLosslessImageCompression
JPEGLSLossyNearLosslessImageCompression
JPEG2000ImageCompressionLosslessOnly
JPEG2000ImageCompression
JPEG2000Part2MulticomponentImageCompressionLosslessOnly
JPEG2000Part2MulticomponentImageCompression
JPIPReferenced
JPIPReferencedDeflate
MPEG2MainProfileMainLevel
RLELossless
RFC2557MIMEencapsulation
XMLEncoding
MediaStorageDirectoryStorage
TalairachBrainAtlasFrameofReference
SPM2T1FrameofReference
SPM2T2FrameofReference
SPM2PDFrameofReference
SPM2EPIFrameofReference
SPM2FILT1FrameofReference
SPM2PETFrameofReference
SPM2TRANSMFrameofReference
SPM2SPECTFrameofReference
SPM2GRAYFrameofReference
SPM2WHITEFrameofReference
SPM2CSFFrameofReference
SPM2BRAINMASKFrameofReference
SPM2AVG305T1FrameofReference
SPM2AVG152T1FrameofReference
SPM2AVG152T2FrameofReference
SPM2AVG152PDFrameofReference

SPM2SINGLESUBJT1FrameofReference
ICBM452T1FrameofReference
ICBMSingleSubjectMRIFrameofReference
BasicStudyContentNotificationSOPClassRetired
StorageCommitmentPushModelSOPClass
StorageCommitmentPushModelSOPInstance
StorageCommitmentPullModelSOPClassRetired
StorageCommitmentPullModelSOPInstanceRetired
ProceduralEventLoggingSOPClass
ProceduralEventLoggingSOPInstance
SubstanceAdministrationLoggingSOPClass
SubstanceAdministrationLoggingSOPInstance
DICOMUIDRegistry
DICOMControlledTerminology
DICOMApplicationContextName
DetachedPatientManagementSOPClassRetired
DetachedPatientManagementMetaSOPClassRetired
DetachedVisitManagementSOPClassRetired
DetachedStudyManagementSOPClassRetired
StudyComponentManagementSOPClassRetired
ModalityPerformedProcedureStepSOPClass
ModalityPerformedProcedureStepRetrieveSOPClass
ModalityPerformedProcedureStepNotificationSOPClass
DetachedResultsManagementSOPClassRetired
DetachedResultsManagementMetaSOPClassRetired
DetachedStudyManagementMetaSOPClassRetired
DetachedInterpretationManagementSOPClassRetired
StorageServiceClass
BasicFilmSessionSOPClass
BasicFilmBoxSOPClass
BasicGrayscaleImageBoxSOPClass
BasicColorImageBoxSOPClass
ReferencedImageBoxSOPClassRetired
BasicGrayscalePrintManagementMetaSOPClass
ReferencedGrayscalePrintManagementMetaSOPClassRetired
PrintJobSOPClass
BasicAnnotationBoxSOPClass
PrinterSOPClass
PrinterConfigurationRetrievalSOPClass
PrinterSOPInstance
PrinterConfigurationRetrievalSOPInstance
BasicColorPrintManagementMetaSOPClass

ReferencedColorPrintManagementMetaSOPClassRetired
VOILUTBoxSOPClass
PresentationLUTSOPClass
ImageOverlayBoxSOPClassRetired
BasicPrintImageOverlayBoxSOPClassRetired
PrintQueueSOPInstanceRetired
PrintQueueManagementSOPClassRetired
StoredPrintStorageSOPClassRetired
HardcopyGrayscaleImageStorageSOPClassRetired
HardcopyColorImageStorageSOPClassRetired
PullPrintRequestSOPClassRetired
PullStoredPrintManagementMetaSOPClassRetired
MediaCreationManagementSOPClassUID
ComputedRadiographyImageStorage
DigitalXRayImageStorageForPresentation
DigitalXRayImageStorageForProcessing
DigitalMammographyXRayImageStorageForPresentation
DigitalMammographyXRayImageStorageForProcessing
DigitalIntraoralXRayImageStorageForPresentation
DigitalIntraoralXRayImageStorageForProcessing
CTImageStorage
EnhancedCTImageStorage
UltrasoundMultiframeImageStorageRetired
UltrasoundMultiframeImageStorage
MRImageStorage
EnhancedMRImageStorage
MRSpectroscopyStorage
NuclearMedicineImageStorageRetired
UltrasoundImageStorageRetired
UltrasoundImageStorage
SecondaryCaptureImageStorage
MultiframeSingleBitSecondaryCaptureImageStorage
MultiframeGrayscaleByteSecondaryCaptureImageStorage
MultiframeGrayscaleWordSecondaryCaptureImageStorage
MultiframeTrueColorSecondaryCaptureImageStorage
StandaloneOverlayStorageRetired
StandaloneCurveStorageRetired
WaveformStorageTrialRetired
GeneralECGWaveformStorage
AmbulatoryECGWaveformStorage
HemodynamicWaveformStorage
CardiacElectrophysiologyWaveformStorage

BasicVoiceAudioWaveformStorage
StandaloneModalityLUTStorageRetired
StandaloneVOILUTStorageRetired
GrayscaleSoftcopyPresentationStateStorageSOPClass
ColorSoftcopyPresentationStateStorageSOPClass
PseudoColorSoftcopyPresentationStateStorageSOPClass
BlendingSoftcopyPresentationStateStorageSOPClass
XRayAngiographicImageStorage
EnhancedXAImageStorage
XRayRadiofluoroscopicImageStorage
EnhancedXRFImageStorage
XRay3DAngiographicImageStorage
XRay3DCraniofacialImageStorage
XRayAngiographicBiPlaneImageStorageRetired
NuclearMedicineImageStorage
RawDataStorage
SpatialRegistrationStorage
SpatialFiducialsStorage
DeformableSpatialRegistrationStorage
SegmentationStorage
RealWorldValueMappingStorage
VLImageStorageTrialRetired
VLMultiframeImageStorageTrialRetired
VLEndoscopicImageStorage
VideoEndoscopicImageStorage
VLMicroscopicImageStorage
VideoMicroscopicImageStorage
VLSlideCoordinatesMicroscopicImageStorage
VLPhotographicImageStorage
VideoPhotographicImageStorage
OphthalmicPhotography8BitImageStorage
OphthalmicPhotography16BitImageStorage
StereometricRelationshipStorage
OphthalmicTomographyImageStorage
TextSRStorageTrialRetired
AudioSRStorageTrialRetired
DetailSRStorageTrialRetired
ComprehensiveSRStorageTrialRetired
BasicTextSRStorage
EnhancedSRStorage
ComprehensiveSRStorage
ProcedureLogStorage

MammographyCADSRStorage
KeyObjectSelectionDocumentStorage
ChestCADSRStorage
XRayRadiationDoseSRStorage
EncapsulatedPDFStorage
EncapsulatedCDAStorage
PositronEmissionTomographyImageStorage
StandalonePETCurveStorageRetired
RTImageStorage
RTDoseStorage
RTStructureSetStorage
RTBeamsTreatmentRecordStorage
RTPlanStorage
RTBrachyTreatmentRecordStorage
RTTreatmentSummaryRecordStorage
RTIonPlanStorage
RTIonBeamsTreatmentRecordStorage
PatientRootQueryRetrieveInformationModelFIND
PatientRootQueryRetrieveInformationModelMOVE
PatientRootQueryRetrieveInformationModelGET
StudyRootQueryRetrieveInformationModelFIND
StudyRootQueryRetrieveInformationModelMOVE
StudyRootQueryRetrieveInformationModelGET
PatientStudyOnlyQueryRetrieveInformationModelFINDRetired
PatientStudyOnlyQueryRetrieveInformationModelMOVERetired
PatientStudyOnlyQueryRetrieveInformationModelGETRetired
ModalityWorklistInformationModelFIND
GeneralPurposeWorklistInformationModelFIND
GeneralPurposeScheduledProcedureStepSOPClass
GeneralPurposePerformedProcedureStepSOPClass
GeneralPurposeWorklistManagementMetaSOPClass
InstanceAvailabilityNotificationSOPClass
RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft
RTConventionalMachineVerificationSupplement74FrozenDraft
RTIonMachineVerificationSupplement74FrozenDraft
UnifiedWorklistandProcedureStepServiceClass
UnifiedProcedureStepPushSOPClass
UnifiedProcedureStepWatchSOPClass
UnifiedProcedureStepPullSOPClass
UnifiedProcedureStepEventSOPClass
UnifiedWorklistandProcedureStepSOPInstance
GeneralRelevantPatientInformationQuery

BreastImagingRelevantPatientInformationQuery
CardiacRelevantPatientInformationQuery
HangingProtocolStorage
HangingProtocolInformationModelFIND
HangingProtocolInformationModelMOVE
ProductCharacteristicsQuerySOPClass
SubstanceApprovalQuerySOPClass
dicomDeviceName
dicomDescription
dicomManufacturer
dicomManufacturerModelName
dicomSoftwareVersion
dicomVendorData
dicomAETitle
dicomNetworkConnectionReference
dicomApplicationCluster
dicomAssociationInitiator
dicomAssociationAcceptor
dicomHostname
dicomPort
dicomSOPClass
dicomTransferRole
dicomTransferSyntax
dicomPrimaryDeviceType
dicomRelatedDeviceReference
dicomPreferredCalledAETitle
dicomTLSCyphersuite
dicomAuthorizedNodeCertificateReference
dicomThisNodeCertificateReference
dicomInstalled
dicomStationName
dicomDeviceSerialNumber
dicomInstitutionName
dicomInstitutionAddress
dicomInstitutionDepartmentName
dicomIssuerOfPatientID
dicomPreferredCallingAETitle
dicomSupportedCharacterSet
dicomConfigurationRoot
dicomDevicesRoot
dicomUniqueAETitlesRegistryRoot
dicomDevice

dicomNetworkAE

dicomNetworkConnection

dicomUniqueAETitle

dicomTransferCapability

VLWholeSlideMicroscopyImageStorage

EnhancedUSVolumeStorage

SurfaceSegmentationStorage

BreastTomosynthesisImageStorage

27.290.3.2 enum gdcm::UIDs::TSType

Enumerator

uid_1_2_840_10008_1_1

uid_1_2_840_10008_1_2

uid_1_2_840_10008_1_2_1

uid_1_2_840_10008_1_2_1_99

uid_1_2_840_10008_1_2_2

uid_1_2_840_10008_1_2_4_50

uid_1_2_840_10008_1_2_4_51

uid_1_2_840_10008_1_2_4_52

uid_1_2_840_10008_1_2_4_53

uid_1_2_840_10008_1_2_4_54

uid_1_2_840_10008_1_2_4_55

uid_1_2_840_10008_1_2_4_56

uid_1_2_840_10008_1_2_4_57

uid_1_2_840_10008_1_2_4_58

uid_1_2_840_10008_1_2_4_59

uid_1_2_840_10008_1_2_4_60

uid_1_2_840_10008_1_2_4_61

uid_1_2_840_10008_1_2_4_62

uid_1_2_840_10008_1_2_4_63

uid_1_2_840_10008_1_2_4_64

uid_1_2_840_10008_1_2_4_65

uid_1_2_840_10008_1_2_4_66

uid_1_2_840_10008_1_2_4_70

uid_1_2_840_10008_1_2_4_80

uid_1_2_840_10008_1_2_4_81

uid_1_2_840_10008_1_2_4_90

uid_1_2_840_10008_1_2_4_91

uid_1_2_840_10008_1_2_4_92

uid_1_2_840_10008_1_2_4_93

uid_1_2_840_10008_1_2_4_94
uid_1_2_840_10008_1_2_4_95
uid_1_2_840_10008_1_2_4_100
uid_1_2_840_10008_1_2_5
uid_1_2_840_10008_1_2_6_1
uid_1_2_840_10008_1_2_6_2
uid_1_2_840_10008_1_3_10
uid_1_2_840_10008_1_4_1_1
uid_1_2_840_10008_1_4_1_2
uid_1_2_840_10008_1_4_1_3
uid_1_2_840_10008_1_4_1_4
uid_1_2_840_10008_1_4_1_5
uid_1_2_840_10008_1_4_1_6
uid_1_2_840_10008_1_4_1_7
uid_1_2_840_10008_1_4_1_8
uid_1_2_840_10008_1_4_1_9
uid_1_2_840_10008_1_4_1_10
uid_1_2_840_10008_1_4_1_11
uid_1_2_840_10008_1_4_1_12
uid_1_2_840_10008_1_4_1_13
uid_1_2_840_10008_1_4_1_14
uid_1_2_840_10008_1_4_1_15
uid_1_2_840_10008_1_4_1_16
uid_1_2_840_10008_1_4_1_17
uid_1_2_840_10008_1_4_1_18
uid_1_2_840_10008_1_4_2_1
uid_1_2_840_10008_1_4_2_2
uid_1_2_840_10008_1_9
uid_1_2_840_10008_1_20_1
uid_1_2_840_10008_1_20_1_1
uid_1_2_840_10008_1_20_2
uid_1_2_840_10008_1_20_2_1
uid_1_2_840_10008_1_40
uid_1_2_840_10008_1_40_1
uid_1_2_840_10008_1_42
uid_1_2_840_10008_1_42_1
uid_1_2_840_10008_2_6_1
uid_1_2_840_10008_2_16_4
uid_1_2_840_10008_3_1_1_1
uid_1_2_840_10008_3_1_2_1_1
uid_1_2_840_10008_3_1_2_1_4
uid_1_2_840_10008_3_1_2_2_1

uid_1_2_840_10008_3_1_2_3_1
uid_1_2_840_10008_3_1_2_3_2
uid_1_2_840_10008_3_1_2_3_3
uid_1_2_840_10008_3_1_2_3_4
uid_1_2_840_10008_3_1_2_3_5
uid_1_2_840_10008_3_1_2_5_1
uid_1_2_840_10008_3_1_2_5_4
uid_1_2_840_10008_3_1_2_5_5
uid_1_2_840_10008_3_1_2_6_1
uid_1_2_840_10008_4_2
uid_1_2_840_10008_5_1_1_1
uid_1_2_840_10008_5_1_1_2
uid_1_2_840_10008_5_1_1_4
uid_1_2_840_10008_5_1_1_4_1
uid_1_2_840_10008_5_1_1_4_2
uid_1_2_840_10008_5_1_1_9
uid_1_2_840_10008_5_1_1_9_1
uid_1_2_840_10008_5_1_1_14
uid_1_2_840_10008_5_1_1_15
uid_1_2_840_10008_5_1_1_16
uid_1_2_840_10008_5_1_1_16_376
uid_1_2_840_10008_5_1_1_17
uid_1_2_840_10008_5_1_1_17_376
uid_1_2_840_10008_5_1_1_18
uid_1_2_840_10008_5_1_1_18_1
uid_1_2_840_10008_5_1_1_22
uid_1_2_840_10008_5_1_1_23
uid_1_2_840_10008_5_1_1_24
uid_1_2_840_10008_5_1_1_24_1
uid_1_2_840_10008_5_1_1_25
uid_1_2_840_10008_5_1_1_26
uid_1_2_840_10008_5_1_1_27
uid_1_2_840_10008_5_1_1_29
uid_1_2_840_10008_5_1_1_30
uid_1_2_840_10008_5_1_1_31
uid_1_2_840_10008_5_1_1_32
uid_1_2_840_10008_5_1_1_33
uid_1_2_840_10008_5_1_4_1_1_1
uid_1_2_840_10008_5_1_4_1_1_1_1
uid_1_2_840_10008_5_1_4_1_1_1_1_1
uid_1_2_840_10008_5_1_4_1_1_1_2
uid_1_2_840_10008_5_1_4_1_1_1_2_1

uid_1_2_840_10008_5_1_4_1_1_1_3
uid_1_2_840_10008_5_1_4_1_1_1_3_1
uid_1_2_840_10008_5_1_4_1_1_2
uid_1_2_840_10008_5_1_4_1_1_2_1
uid_1_2_840_10008_5_1_4_1_1_3
uid_1_2_840_10008_5_1_4_1_1_3_1
uid_1_2_840_10008_5_1_4_1_1_4
uid_1_2_840_10008_5_1_4_1_1_4_1
uid_1_2_840_10008_5_1_4_1_1_4_2
uid_1_2_840_10008_5_1_4_1_1_5
uid_1_2_840_10008_5_1_4_1_1_6
uid_1_2_840_10008_5_1_4_1_1_6_1
uid_1_2_840_10008_5_1_4_1_1_7
uid_1_2_840_10008_5_1_4_1_1_7_1
uid_1_2_840_10008_5_1_4_1_1_7_2
uid_1_2_840_10008_5_1_4_1_1_7_3
uid_1_2_840_10008_5_1_4_1_1_7_4
uid_1_2_840_10008_5_1_4_1_1_8
uid_1_2_840_10008_5_1_4_1_1_9
uid_1_2_840_10008_5_1_4_1_1_9_1
uid_1_2_840_10008_5_1_4_1_1_9_1_1
uid_1_2_840_10008_5_1_4_1_1_9_1_2
uid_1_2_840_10008_5_1_4_1_1_9_1_3
uid_1_2_840_10008_5_1_4_1_1_9_2_1
uid_1_2_840_10008_5_1_4_1_1_9_3_1
uid_1_2_840_10008_5_1_4_1_1_9_4_1
uid_1_2_840_10008_5_1_4_1_1_10
uid_1_2_840_10008_5_1_4_1_1_11
uid_1_2_840_10008_5_1_4_1_1_11_1
uid_1_2_840_10008_5_1_4_1_1_11_2
uid_1_2_840_10008_5_1_4_1_1_11_3
uid_1_2_840_10008_5_1_4_1_1_11_4
uid_1_2_840_10008_5_1_4_1_1_12_1
uid_1_2_840_10008_5_1_4_1_1_12_1_1
uid_1_2_840_10008_5_1_4_1_1_12_2
uid_1_2_840_10008_5_1_4_1_1_12_2_1
uid_1_2_840_10008_5_1_4_1_1_13_1_1
uid_1_2_840_10008_5_1_4_1_1_13_1_2
uid_1_2_840_10008_5_1_4_1_1_12_3
uid_1_2_840_10008_5_1_4_1_1_20
uid_1_2_840_10008_5_1_4_1_1_66
uid_1_2_840_10008_5_1_4_1_1_66_1

uid_1_2_840_10008_5_1_4_1_1_66_2
uid_1_2_840_10008_5_1_4_1_1_66_3
uid_1_2_840_10008_5_1_4_1_1_66_4
uid_1_2_840_10008_5_1_4_1_1_67
uid_1_2_840_10008_5_1_4_1_1_77_1
uid_1_2_840_10008_5_1_4_1_1_77_2
uid_1_2_840_10008_5_1_4_1_1_77_1_1
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1
uid_1_2_840_10008_5_1_4_1_1_77_1_2
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
uid_1_2_840_10008_5_1_4_1_1_77_1_3
uid_1_2_840_10008_5_1_4_1_1_77_1_4
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
uid_1_2_840_10008_5_1_4_1_1_88_1
uid_1_2_840_10008_5_1_4_1_1_88_2
uid_1_2_840_10008_5_1_4_1_1_88_3
uid_1_2_840_10008_5_1_4_1_1_88_4
uid_1_2_840_10008_5_1_4_1_1_88_11
uid_1_2_840_10008_5_1_4_1_1_88_22
uid_1_2_840_10008_5_1_4_1_1_88_33
uid_1_2_840_10008_5_1_4_1_1_88_40
uid_1_2_840_10008_5_1_4_1_1_88_50
uid_1_2_840_10008_5_1_4_1_1_88_59
uid_1_2_840_10008_5_1_4_1_1_88_65
uid_1_2_840_10008_5_1_4_1_1_88_67
uid_1_2_840_10008_5_1_4_1_1_104_1
uid_1_2_840_10008_5_1_4_1_1_104_2
uid_1_2_840_10008_5_1_4_1_1_128
uid_1_2_840_10008_5_1_4_1_1_129
uid_1_2_840_10008_5_1_4_1_1_481_1
uid_1_2_840_10008_5_1_4_1_1_481_2
uid_1_2_840_10008_5_1_4_1_1_481_3
uid_1_2_840_10008_5_1_4_1_1_481_4
uid_1_2_840_10008_5_1_4_1_1_481_5
uid_1_2_840_10008_5_1_4_1_1_481_6
uid_1_2_840_10008_5_1_4_1_1_481_7
uid_1_2_840_10008_5_1_4_1_1_481_8
uid_1_2_840_10008_5_1_4_1_1_481_9

uid_1_2_840_10008_5_1_4_1_2_1_1
uid_1_2_840_10008_5_1_4_1_2_1_2
uid_1_2_840_10008_5_1_4_1_2_1_3
uid_1_2_840_10008_5_1_4_1_2_2_1
uid_1_2_840_10008_5_1_4_1_2_2_2
uid_1_2_840_10008_5_1_4_1_2_2_3
uid_1_2_840_10008_5_1_4_1_2_3_1
uid_1_2_840_10008_5_1_4_1_2_3_2
uid_1_2_840_10008_5_1_4_1_2_3_3
uid_1_2_840_10008_5_1_4_31
uid_1_2_840_10008_5_1_4_32_1
uid_1_2_840_10008_5_1_4_32_2
uid_1_2_840_10008_5_1_4_32_3
uid_1_2_840_10008_5_1_4_32
uid_1_2_840_10008_5_1_4_33
uid_1_2_840_10008_5_1_4_34_1
uid_1_2_840_10008_5_1_4_34_2
uid_1_2_840_10008_5_1_4_34_3
uid_1_2_840_10008_5_1_4_34_4
uid_1_2_840_10008_5_1_4_34_4_1
uid_1_2_840_10008_5_1_4_34_4_2
uid_1_2_840_10008_5_1_4_34_4_3
uid_1_2_840_10008_5_1_4_34_4_4
uid_1_2_840_10008_5_1_4_34_5
uid_1_2_840_10008_5_1_4_37_1
uid_1_2_840_10008_5_1_4_37_2
uid_1_2_840_10008_5_1_4_37_3
uid_1_2_840_10008_5_1_4_38_1
uid_1_2_840_10008_5_1_4_38_2
uid_1_2_840_10008_5_1_4_38_3
uid_1_2_840_10008_5_1_4_41
uid_1_2_840_10008_5_1_4_42
uid_1_2_840_10008_15_0_3_1
uid_1_2_840_10008_15_0_3_2
uid_1_2_840_10008_15_0_3_3
uid_1_2_840_10008_15_0_3_4
uid_1_2_840_10008_15_0_3_5
uid_1_2_840_10008_15_0_3_6
uid_1_2_840_10008_15_0_3_7
uid_1_2_840_10008_15_0_3_8
uid_1_2_840_10008_15_0_3_9
uid_1_2_840_10008_15_0_3_10

uid_1_2_840_10008_15_0_3_11
uid_1_2_840_10008_15_0_3_12
uid_1_2_840_10008_15_0_3_13
uid_1_2_840_10008_15_0_3_14
uid_1_2_840_10008_15_0_3_15
uid_1_2_840_10008_15_0_3_16
uid_1_2_840_10008_15_0_3_17
uid_1_2_840_10008_15_0_3_18
uid_1_2_840_10008_15_0_3_19
uid_1_2_840_10008_15_0_3_20
uid_1_2_840_10008_15_0_3_21
uid_1_2_840_10008_15_0_3_22
uid_1_2_840_10008_15_0_3_23
uid_1_2_840_10008_15_0_3_24
uid_1_2_840_10008_15_0_3_25
uid_1_2_840_10008_15_0_3_26
uid_1_2_840_10008_15_0_3_27
uid_1_2_840_10008_15_0_3_28
uid_1_2_840_10008_15_0_3_29
uid_1_2_840_10008_15_0_3_30
uid_1_2_840_10008_15_0_3_31
uid_1_2_840_10008_15_0_4_1
uid_1_2_840_10008_15_0_4_2
uid_1_2_840_10008_15_0_4_3
uid_1_2_840_10008_15_0_4_4
uid_1_2_840_10008_15_0_4_5
uid_1_2_840_10008_15_0_4_6
uid_1_2_840_10008_15_0_4_7
uid_1_2_840_10008_15_0_4_8
uid_1_2_840_10008_5_1_4_1_1_77_1_6
uid_1_2_840_10008_5_1_4_1_1_6_2
uid_1_2_840_10008_5_1_4_1_1_66_5
uid_1_2_840_10008_5_1_4_1_1_13_1_3

27.290.4 Member Function Documentation

27.290.4.1 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<<().

27.290.4.2 static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( ) [static]

27.290.4.3 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<<().

27.290.4.4 static const char* const* gdcm::UIDs::GetTransferSyntaxString (unsigned int *ts*) [static]

27.290.4.5 static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ( ) [static]

27.290.4.6 static const char* gdcm::UIDs::GetUIDName (unsigned int *ts*) [static]

27.290.4.7 static const char* gdcm::UIDs::GetUIDString (unsigned int *ts*) [static]

27.290.4.8 gdcm::UIDs::operator TSType ( ) const [inline]

27.290.4.9 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:

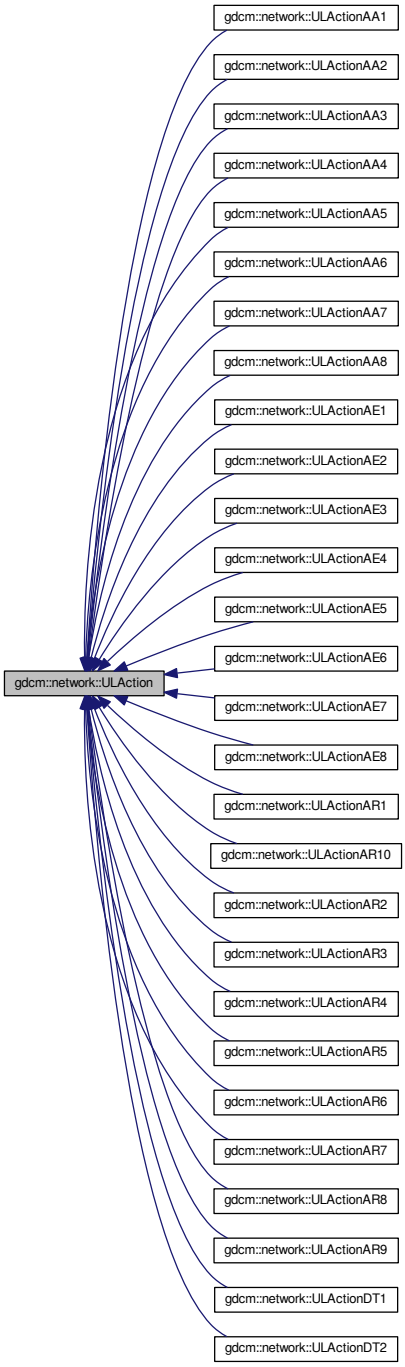
- gdcmUIDs.h

27.291 gdcm::network::ULAction Class Reference

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.

`#include <gdcmULAction.h>`

Inheritance diagram for gdcm::network::ULAction:



Public Member Functions

- ULAction ()
- virtual ~ULAction ()

- virtual EStateID PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaiting↩ ForEvent, EEventID &outRaisedEvent)=0

27.291.1 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 thos 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.

27.291.2 Constructor & Destructor Documentation

27.291.2.1 gdcm::network::ULAction::ULAction ( ) [inline]

27.291.2.2 virtual gdcm::network::ULAction::~ULAction ( ) [inline], [virtual]

27.291.3 Member Function Documentation

27.291.3.1 virtual EStateID gdcm::network::ULAction::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [pure virtual]

Implemented in gdcm::network::ULActionAR10, gdcm::network::ULActionAR9, gdcm::network::ULActionAE8, gdcm↩ ::network::ULActionAA8, gdcm::network::ULActionAR8, gdcm::network::ULActionAE7, gdcm::network::ULActionA↩ A7, gdcm::network::ULActionAR7, gdcm::network::ULActionAE6, gdcm::network::ULActionAA6, gdcm::network::↩ ULActionAR6, gdcm::network::ULActionAA5, gdcm::network::ULActionAE5, gdcm::network::ULActionAR5, gdcm↩ ::network::ULActionAA4, gdcm::network::ULActionAE4, gdcm::network::ULActionAR4, gdcm::network::ULActionA↩ A3, gdcm::network::ULActionAE3, gdcm::network::ULActionAR3, gdcm::network::ULActionAA2, gdcm::network::↩ ULActionAE2, gdcm::network::ULActionAR2, gdcm::network::ULActionDT2, gdcm::network::ULActionAA1, gdcm↩ ::network::ULActionAE1, gdcm::network::ULActionAR1, and gdcm::network::ULActionDT1.

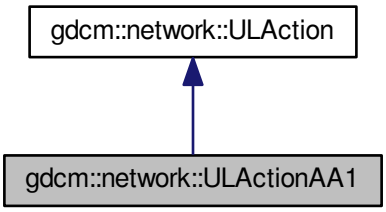
The documentation for this class was generated from the following file:

- gdcmULAction.h

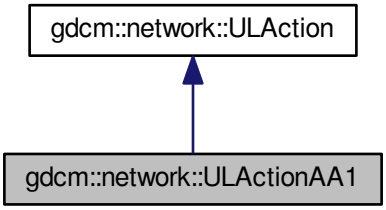
27.292 gdcm::network::ULActionAA1 Class Reference

`#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)

27.292.1 Member Function Documentation

27.292.1.1 EStateID gdcm::network::ULActionAA1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

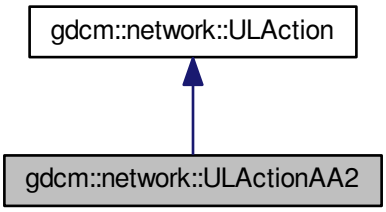
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

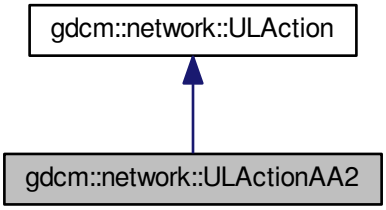
27.293 gdcm::network::ULActionAA2 Class Reference

`#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)

27.293.1 Member Function Documentation

27.293.1.1 EStateID gdcm::network::ULActionAA2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

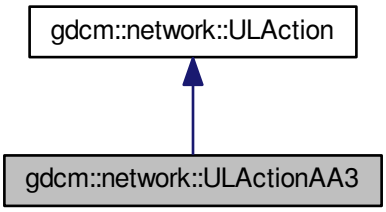
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

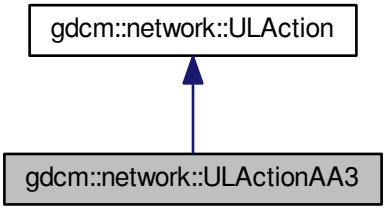
27.294 gdcm::network::ULActionAA3 Class Reference

`#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)

27.294.1 Member Function Documentation

27.294.1.1 EStateID gdcm::network::ULActionAA3::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

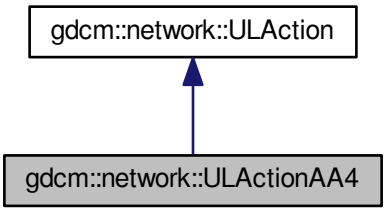
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

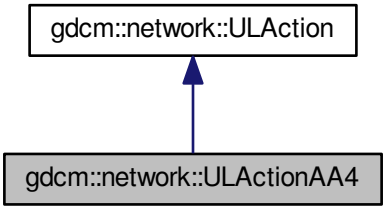
27.295 gdcm::network::ULActionAA4 Class Reference

`#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)

27.295.1 Member Function Documentation

27.295.1.1 EStateID gdcm::network::ULActionAA4::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

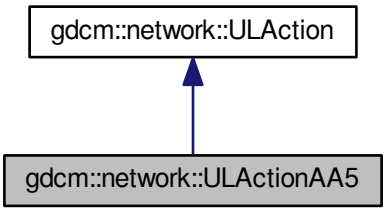
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

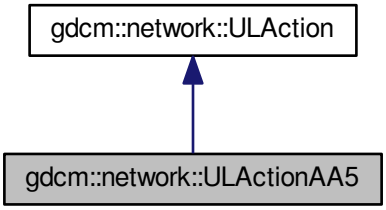
27.296 gdcm::network::ULActionAA5 Class Reference

`#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)

27.296.1 Member Function Documentation

27.296.1.1 EStateID gdcm::network::ULActionAA5::PerformAction (Subject ∗ *s,* ULEvent & *inEvent,* ULConnection & *inConnection,* bool & *outWaitingForEvent,* EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

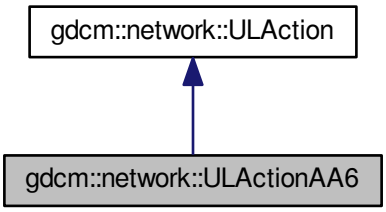
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

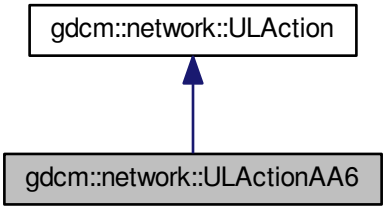
27.297 gdcm::network::ULActionAA6 Class Reference

`#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)

27.297.1 Member Function Documentation

27.297.1.1 EStateID gdcm::network::ULActionAA6::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

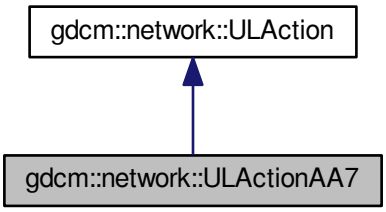
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

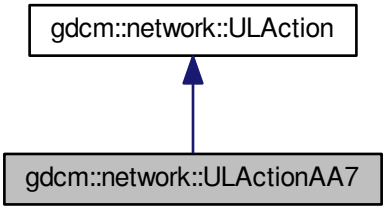
27.298 gdcm::network::ULActionAA7 Class Reference

`#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)

27.298.1 Member Function Documentation

27.298.1.1 EStateID gdcm::network::ULActionAA7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

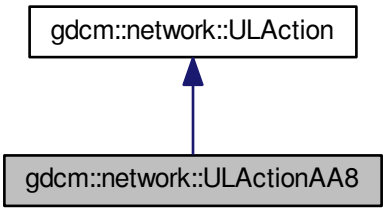
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

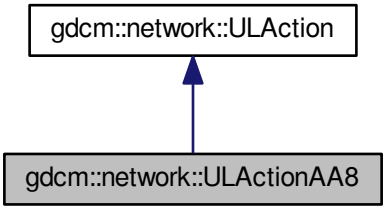
27.299 gdcm::network::ULActionAA8 Class Reference

`#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)

27.299.1 Member Function Documentation

27.299.1.1 EStateID gdcm::network::ULActionAA8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

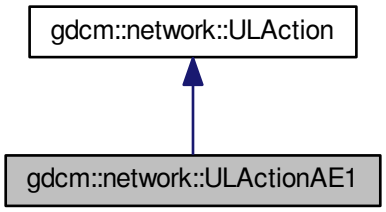
The documentation for this class was generated from the following file:

- gdcmULActionAA.h

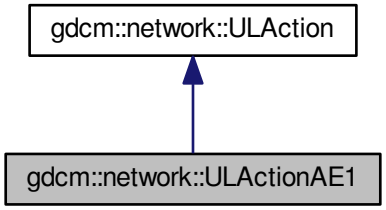
27.300 gdcm::network::ULActionAE1 Class Reference

`#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)

27.300.1 Member Function Documentation

27.300.1.1 EStateID gdcm::network::ULActionAE1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

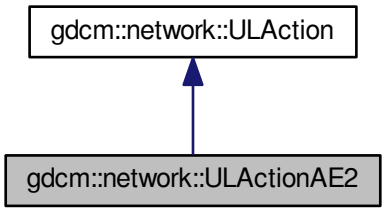
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

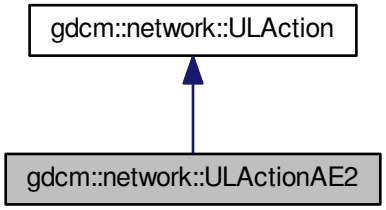
27.301 gdcm::network::ULActionAE2 Class Reference

`#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)

27.301.1 Member Function Documentation

27.301.1.1 EStateID gdcm::network::ULActionAE2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

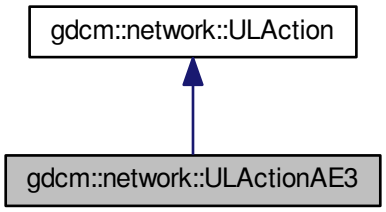
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

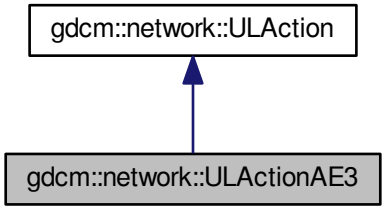
27.302 gdcm::network::ULActionAE3 Class Reference

`#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)

27.302.1 Member Function Documentation

27.302.1.1 EStateID gdcm::network::ULActionAE3::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

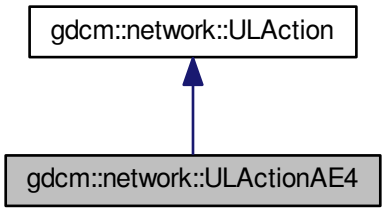
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

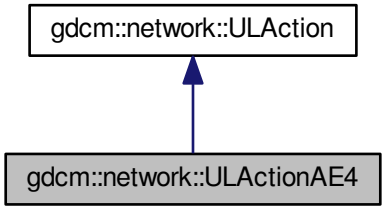
27.303 gdcm::network::ULActionAE4 Class Reference

`#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)

27.303.1 Member Function Documentation

27.303.1.1 EStateID gdcm::network::ULActionAE4::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

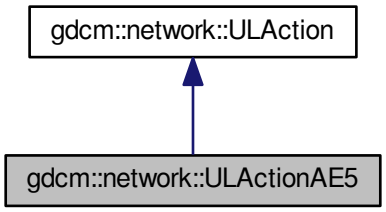
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

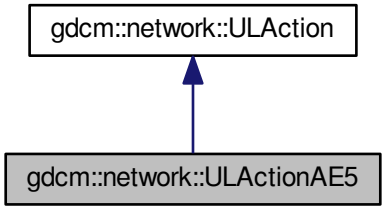
27.304 gdcm::network::ULActionAE5 Class Reference

`#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)

27.304.1 Member Function Documentation

27.304.1.1 EStateID gdcm::network::ULActionAE5::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

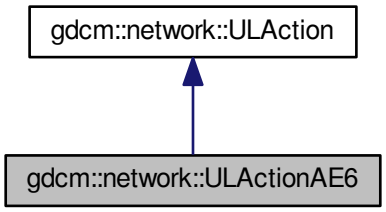
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

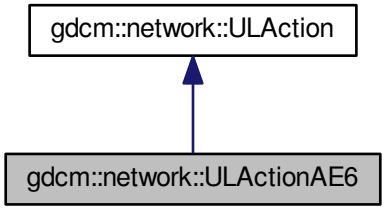
27.305 gdcm::network::ULActionAE6 Class Reference

`#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)

27.305.1 Member Function Documentation

27.305.1.1 EStateID gdcm::network::ULActionAE6::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

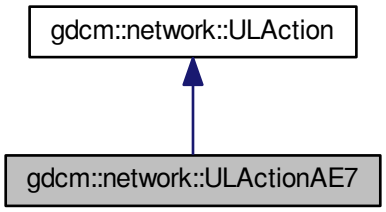
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

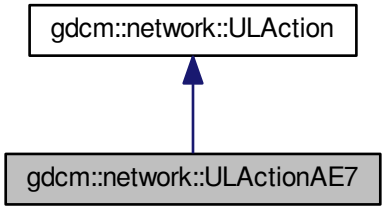
27.306 gdcm::network::ULActionAE7 Class Reference

`#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)

27.306.1 Member Function Documentation

27.306.1.1 EStateID gdcm::network::ULActionAE7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

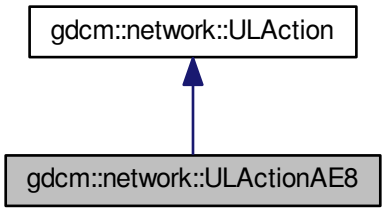
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

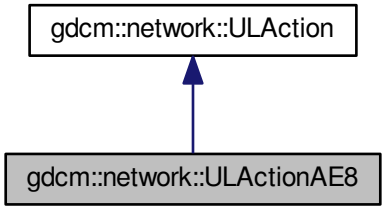
27.307 gdcm::network::ULActionAE8 Class Reference

`#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)

27.307.1 Member Function Documentation

27.307.1.1 EStateID gdcm::network::ULActionAE8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

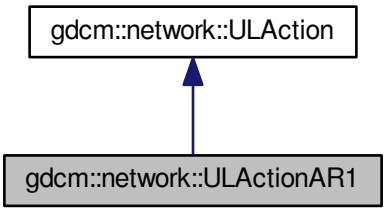
The documentation for this class was generated from the following file:

- gdcmULActionAE.h

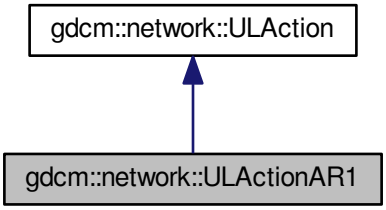
27.308 gdcm::network::ULActionAR1 Class Reference

`#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)

27.308.1 Member Function Documentation

27.308.1.1 EStateID gdcm::network::ULActionAR1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

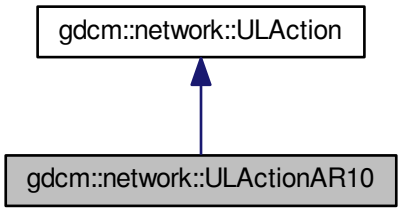
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

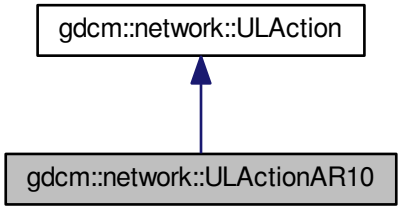
27.309 gdcm::network::ULActionAR10 Class Reference

`#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)

27.309.1 Member Function Documentation

27.309.1.1 EStateID gdcm::network::ULActionAR10::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

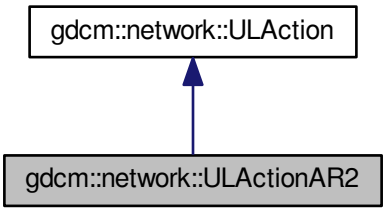
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

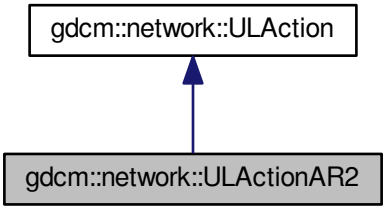
27.310 gdcm::network::ULActionAR2 Class Reference

`#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)

27.310.1 Member Function Documentation

27.310.1.1 EStateID gdcm::network::ULActionAR2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

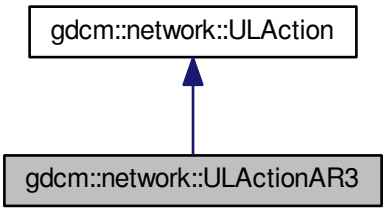
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

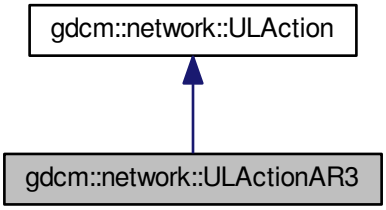
27.311 gdcm::network::ULActionAR3 Class Reference

`#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)

27.311.1 Member Function Documentation

27.311.1.1 EStateID gdcm::network::ULActionAR3::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

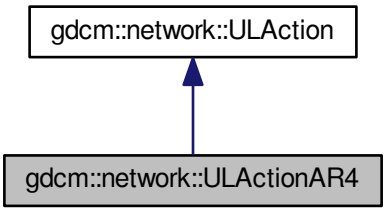
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

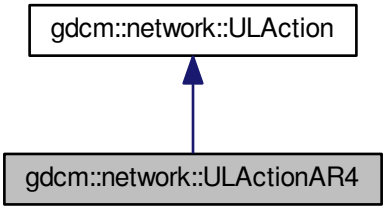
27.312 gdcm::network::ULActionAR4 Class Reference

`#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)

27.312.1 Member Function Documentation

27.312.1.1 EStateID gdcm::network::ULActionAR4::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

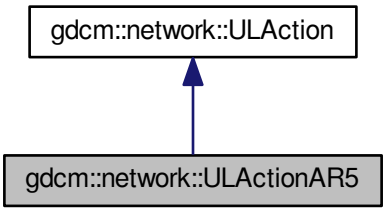
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

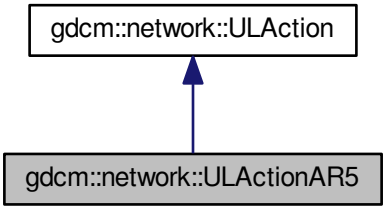
27.313 gdcm::network::ULActionAR5 Class Reference

`#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)

27.313.1 Member Function Documentation

27.313.1.1 EStateID gdcm::network::ULActionAR5::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

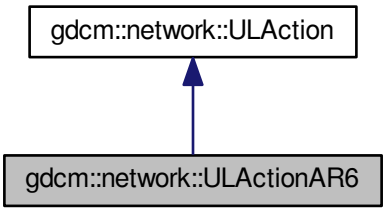
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

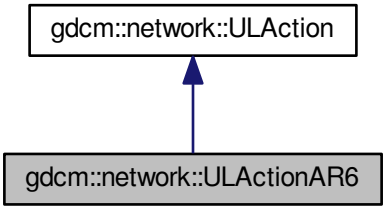
27.314 gdcm::network::ULActionAR6 Class Reference

`#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)

27.314.1 Member Function Documentation

27.314.1.1 EStateID gdcm::network::ULActionAR6::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

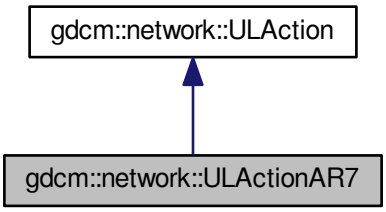
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

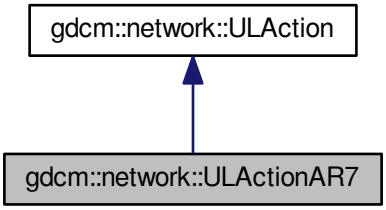
27.315 gdcm::network::ULActionAR7 Class Reference

`#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)

27.315.1 Member Function Documentation

27.315.1.1 EStateID gdcm::network::ULActionAR7::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

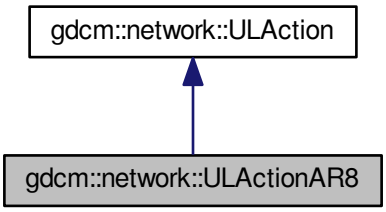
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

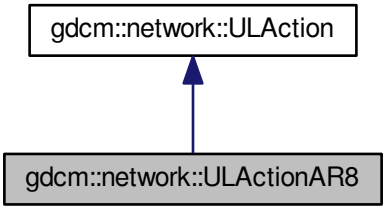
27.316 gdcm::network::ULActionAR8 Class Reference

`#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)

27.316.1 Member Function Documentation

27.316.1.1 EStateID gdcm::network::ULActionAR8::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

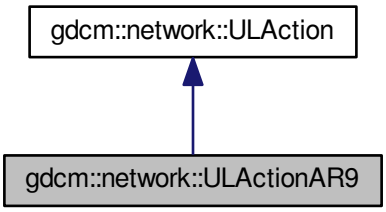
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

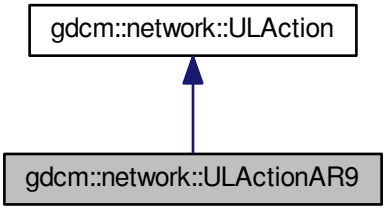
27.317 gdcm::network::ULActionAR9 Class Reference

`#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)

27.317.1 Member Function Documentation

27.317.1.1 EStateID gdcm::network::ULActionAR9::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

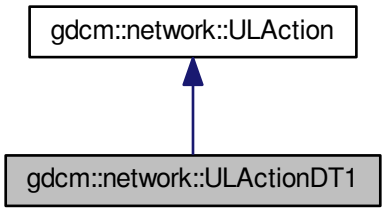
The documentation for this class was generated from the following file:

- gdcmULActionAR.h

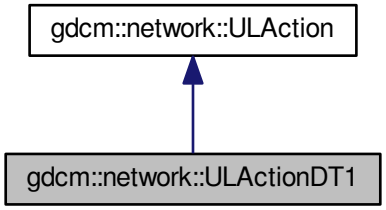
27.318 gdcm::network::ULActionDT1 Class Reference

`#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)

27.318.1 Member Function Documentation

27.318.1.1 EStateID gdcm::network::ULActionDT1::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

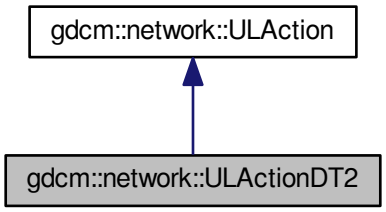
The documentation for this class was generated from the following file:

- gdcmULActionDT.h

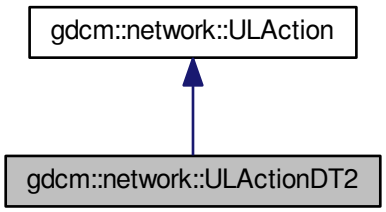
27.319 gdcm::network::ULActionDT2 Class Reference

`#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)

27.319.1 Member Function Documentation

27.319.1.1 EStateID gdcm::network::ULActionDT2::PerformAction (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) [virtual]

Implements gdcm::network::ULAction.

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

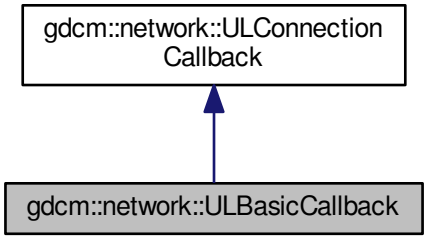
- gdcmULActionDT.h

27.320 gdcm::network::ULBasicCallback Class Reference

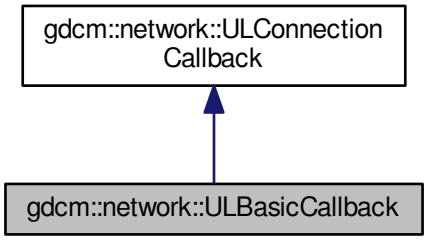
ULBasicCallback This is the most basic of callbacks for how the ULConnectionManager handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to disk, etc. This class is NOT THREAD SAFE. Access the dataset vector after the entire set of datasets has been returned by the ULConnectionManager.

`#include <gdcmULBasicCallback.h>`

Inheritance diagram for gdcm::network::ULBasicCallback:



Collaboration diagram for gdcm::network::ULBasicCallback:



Public Member Functions

- ULBasicCallback ()
- virtual ~ULBasicCallback ()
- std::vector< DataSet > const & GetDataSets () const
- std::vector< DataSet > const & GetResponses () const
- virtual void HandleDataSet (const DataSet &inDataSet)
- virtual void HandleResponse (const DataSet &inDataSet)

Additional Inherited Members

27.320.1 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.

27.320.2 Constructor & Destructor Documentation

27.320.2.1 gdcm::network::ULBasicCallback::ULBasicCallback ( ) [inline]

27.320.2.2 virtual gdcm::network::ULBasicCallback::~ULBasicCallback ( ) [inline], [virtual]

27.320.3 Member Function Documentation

27.320.3.1 std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetDataSets ( ) const

27.320.3.2 std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetResponses ( ) const

27.320.3.3 virtual void gdcm::network::ULBasicCallback::HandleDataSet (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

27.320.3.4 virtual void gdcm::network::ULBasicCallback::HandleResponse (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

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

- gdcmULBasicCallback.h

27.321 gdcm::network::ULConnection Class Reference

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.

`#include <gdcmULConnection.h>`

Public Member Functions

- ULConnection (const ULConnectionInfo &inUserInformation)
- virtual ~ULConnection ()
- void AddAcceptedPresentationContext (const PresentationContextAC &inPC)
- PresentationContextRQ FindContext (const DataElement &de) const
- std::vector
  < PresentationContextAC >
  const & GetAcceptedPresentationContexts () const

- std::vector
  < PresentationContextAC > & GetAcceptedPresentationContexts ()
- 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 SetMaxPDUSize (uint32_t inSize)
- void SetPresentationContexts (const std::vector< PresentationContextRQ > &inContexts)
- void SetPresentationContexts (const std::vector< PresentationContext > &inContexts)
- void SetState (const EStateID &inState)
- void StopProtocol ()

Friends

- class ULActionAE6
- class ULConnectionManager

27.321.1 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.

27.321.2 Constructor & Destructor Documentation

27.321.2.1 gdcm::network::ULConnection::ULConnection (const ULConnectionInfo & *inUserInformation*)

27.321.2.2 virtual gdcm::network::ULConnection::~ULConnection ( ) [virtual]

27.321.3 Member Function Documentation

27.321.3.1 void gdcm::network::ULConnection::AddAcceptedPresentationContext (const PresentationContextAC & *inPC*)

27.321.3.2 PresentationContextRQ gdcm::network::ULConnection::FindContext (const DataElement & *de*) const

27.321.3.3 std::vector< PresentationContextAC > const& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( ) const

27.321.3.4 std::vector< PresentationContextAC >& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( )

27.321.3.5 const ULConnectionInfo& gdcm::network::ULConnection::GetConnectionInfo ( ) const

27.321.3.6 uint32_t gdcm::network::ULConnection::GetMaxPDUSize ( ) const

27.321.3.7 const PresentationContextAC* gdcm::network::ULConnection::GetPresentationContextACByID (uint8_t *id*) const

27.321.3.8 uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (PresentationContextRQ const & *pc*) const

return 0 upon error

27.321.3.9 const PresentationContextRQ* gdcm::network::ULConnection::GetPresentationContextRQByID (uint8_t *id*) const

27.321.3.10 std::vector< PresentationContextRQ > const& gdcm::network::ULConnection::GetPresentationContexts ( ) const

27.321.3.11 std::iostream* gdcm::network::ULConnection::GetProtocol ( )

27.321.3.12 EStateID gdcm::network::ULConnection::GetState ( ) const

27.321.3.13 ARTIMTimer& gdcm::network::ULConnection::GetTimer ( )

27.321.3.14 bool gdcm::network::ULConnection::InitializeConnection ( )

used to establish scu connections

27.321.3.15 bool gdcm::network::ULConnection::InitializeIncomingConnection ( )

used to establish scp connections

27.321.3.16 void gdcm::network::ULConnection::SetMaxPDUSize (uint32_t *inSize*)

27.321.3.17 void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< PresentationContextRQ > & *inContexts*)

27.321.3.18 void gdcm::network::ULConnection::SetPresentationContexts (const std::vector< PresentationContext > & *inContexts*)

27.321.3.19 void gdcm::network::ULConnection::SetState (const EStateID & *inState*)

27.321.3.20 void gdcm::network::ULConnection::StopProtocol ( )

27.321.4 Friends And Related Function Documentation

27.321.4.1 friend class ULActionAE6 [friend]

27.321.4.2 friend class ULConnectionManager [friend]

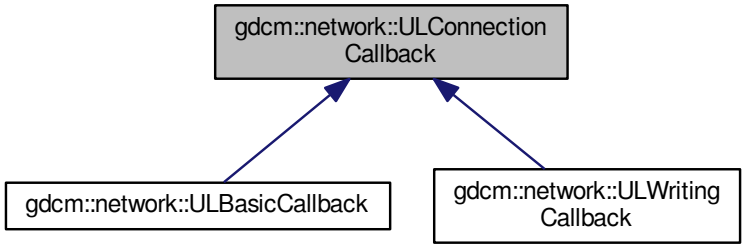
The documentation for this class was generated from the following file:

- gdcmULConnection.h

27.322 gdcm::network::ULConnectionCallback Class Reference

`#include <gdcmULConnectionCallback.h>`

Inheritance diagram for gdcm::network::ULConnectionCallback:



Public Member Functions

- ULConnectionCallback ()
- virtual ~ULConnectionCallback ()
- 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

27.322.1 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.

27.322.2 Constructor & Destructor Documentation

27.322.2.1 gdcm::network::ULConnectionCallback::ULConnectionCallback ( ) [inline]

27.322.2.2 virtual gdcm::network::ULConnectionCallback::~ULConnectionCallback ( ) [inline], [virtual]

27.322.3 Member Function Documentation

27.322.3.1 void gdcm::network::ULConnectionCallback::DataSetHandled ( ) [inline], [protected]

27.322.3.2 bool gdcm::network::ULConnectionCallback::DataSetHandles ( ) const [inline]

27.322.3.3 virtual void gdcm::network::ULConnectionCallback::HandleDataSet (const DataSet & *inDataSet*) [pure virtual]

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

27.322.3.4 virtual void gdcm::network::ULConnectionCallback::HandleResponse (const DataSet & *inDataSet*) [pure virtual]

Implemented in gdcm::network::ULBasicCallback, and gdcm::network::ULWritingCallback.

27.322.3.5 void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( ) [inline]

27.322.3.6 void gdcm::network::ULConnectionCallback::SetImplicitFlag (const bool *imp*) [inline]

27.322.4 Member Data Documentation

27.322.4.1 bool gdcm::network::ULConnectionCallback::mImplicit [protected]

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

- gdcmULConnectionCallback.h

27.323 gdcm::network::ULConnectionInfo Class Reference

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.

`#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 *inCallingAE↩ Title, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void SetMaxPDULength (unsigned long inMaxPDULength)

27.323.1 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.

27.323.2 Constructor & Destructor Documentation

27.323.2.1 gdcm::network::ULConnectionInfo::ULConnectionInfo ( )

27.323.3 Member Function Documentation

27.323.3.1 const char* gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const

27.323.3.2 std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const

27.323.3.3 unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const

27.323.3.4 int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const

27.323.3.5 const char* gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const

27.323.3.6 unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const

27.323.3.7 bool gdcm::network::ULConnectionInfo::Initialize (UserInformation const & *inUserInformation,* const char * *inCalledAETitle,* const char * *inCallingAETitle,* unsigned long *inCalledIPAddress,* int *inCalledIPPort,* std::string *inCalledComputerName*)

27.323.3.8 void gdcm::network::ULConnectionInfo::SetMaxPDULength (unsigned long *inMaxPDULength*)

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

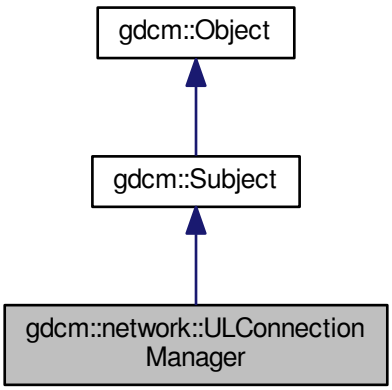
- gdcmULConnectionInfo.h

27.324 gdcm::network::ULConnectionManager Class Reference

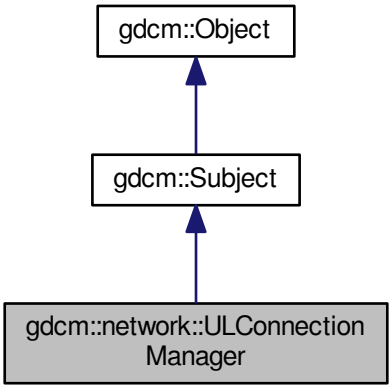
ULConnectionManager The ULConnectionManager performs actions on the ULConnection given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

`#include <gdcmULConnectionManager.h>`

Inheritance diagram for gdcm::network::ULConnectionManager:



Collaboration diagram for gdcm::network::ULConnectionManager:



Public Member Functions

- ULConnectionManager ()
- ~ULConnectionManager ()
- 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 > SendStore (const File &file)
- void SendStore (const File &file, ULConnectionCallback *inCallback)

  callback based API

Additional Inherited Members

27.324.1 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.

27.324.2 Constructor & Destructor Documentation

27.324.2.1 gdcm::network::ULConnectionManager::ULConnectionManager ( )

27.324.2.2 gdcm::network::ULConnectionManager::~ULConnectionManager ( )

27.324.3 Member Function Documentation

27.324.3.1 bool gdcm::network::ULConnectionManager::BreakConnection (const double & *inTimeout*)

27.324.3.2 void gdcm::network::ULConnectionManager::BreakConnectionNow ( )

27.324.3.3 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.

27.324.3.4 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

27.324.3.5 std::vector<**PresentationDataValue**> gdcm::network::ULConnectionManager::SendEcho ( )

27.324.3.6 std::vector<**DataSet**> gdcm::network::ULConnectionManager::SendFind (const **BaseRootQuery** * *inRootQuery*)

27.324.3.7 void gdcm::network::ULConnectionManager::SendFind (const **BaseRootQuery** * *inRootQuery,* **ULConnectionCallback** * *inCallback*)

27.324.3.8 std::vector<**DataSet**> gdcm::network::ULConnectionManager::SendMove (const **BaseRootQuery** * *inRootQuery*)

27.324.3.9 bool gdcm::network::ULConnectionManager::SendMove (const **BaseRootQuery** * *inRootQuery,* **ULConnectionCallback** * *inCallback*)

return false upon error

27.324.3.10 std::vector<**DataSet**> gdcm::network::ULConnectionManager::SendStore (const **File** & *file*)

27.324.3.11 void gdcm::network::ULConnectionManager::SendStore (const File & *file,* ULConnectionCallback * *inCallback*)

callback based API

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

- gdcmULConnectionManager.h

27.325 gdcm::network::ULEvent Class Reference

ULEvent base class for network events.

`#include <gdcmULEvent.h>`

Public Member Functions

- ULEvent (const EEventID &inEventID, std::vector< BasePDU * > const &inBasePDU)
- ULEvent (const EEventID &inEventID, BasePDU *inBasePDU)
- ~ULEvent ()
- EEventID GetEvent () const
- std::vector< BasePDU * > const & GetPDUs () const
- void SetEvent (const EEventID &inEvent)
- void SetPDU (std::vector< BasePDU * > const &inPDU)

27.325.1 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!

27.325.2 Constructor & Destructor Documentation

27.325.2.1 gdcm::network::ULEvent::ULEvent (const EEventID & *inEventID,* std::vector< BasePDU * > const & *inBasePDU*) [inline]

27.325.2.2 gdcm::network::ULEvent::ULEvent (const EEventID & *inEventID,* BasePDU * *inBasePDU*) [inline]

27.325.2.3 gdcm::network::ULEvent::~ULEvent ( ) [inline]

27.325.3 Member Function Documentation

27.325.3.1 EEventID gdcm::network::ULEvent::GetEvent ( ) const [inline]

27.325.3.2 std::vector<BasePDU*> const& gdcm::network::ULEvent::GetPDUs ( ) const [inline]

27.325.3.3 void gdcm::network::ULEvent::SetEvent (const EEventID & *inEvent*) [inline]

27.325.3.4 void gdcm::network::ULEvent::SetPDU (std::vector< BasePDU * > const & *inPDU*) [inline]

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

- gdcmULEvent.h

27.326 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

27.326.1 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.

27.326.2 Constructor & Destructor Documentation

27.326.2.1 gdcm::network::ULTransitionTable::ULTransitionTable ( )

27.326.3 Member Function Documentation

27.326.3.1 void gdcm::network::ULTransitionTable::HandleEvent (Subject * *s*, ULEvent & *inEvent*, ULConnection & *inConnection*, bool & *outWaitingForEvent*, EEventID & *outRaisedEvent*) const

27.326.3.2 void gdcm::network::ULTransitionTable::PrintTable ( ) const

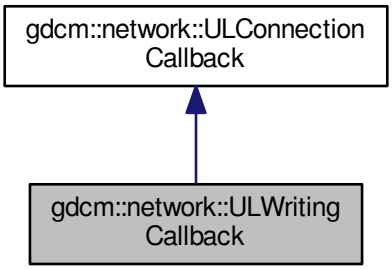
The documentation for this class was generated from the following file:

- gdcmULTransitionTable.h

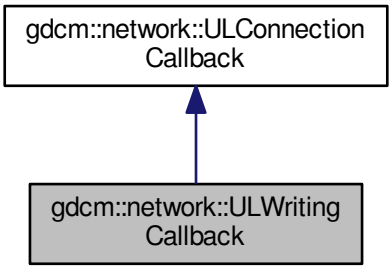
27.327 gdcm::network::ULWritingCallback Class Reference

`#include <gdcmULWritingCallback.h>`

Inheritance diagram for gdcm::network::ULWritingCallback:



Collaboration diagram for gdcm::network::ULWritingCallback:



Public Member Functions

- ULWritingCallback ()
- virtual ~ULWritingCallback ()
- virtual void HandleDataSet (const DataSet &inDataSet)
- virtual void HandleResponse (const DataSet &inDataSet)
- void SetDirectory (const std::string &inDirectoryName)

  provide the directory into which all files are written.

Additional Inherited Members

27.327.1 Constructor & Destructor Documentation

27.327.1.1 gdcm::network::ULWritingCallback::ULWritingCallback ( ) [inline]

27.327.1.2 virtual gdcm::network::ULWritingCallback::~ULWritingCallback ( ) [inline], [virtual]

27.327.2 Member Function Documentation

27.327.2.1 virtual void gdcm::network::ULWritingCallback::HandleDataSet (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

27.327.2.2 virtual void gdcm::network::ULWritingCallback::HandleResponse (const DataSet & *inDataSet*) [virtual]

Implements gdcm::network::ULConnectionCallback.

27.327.2.3 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:

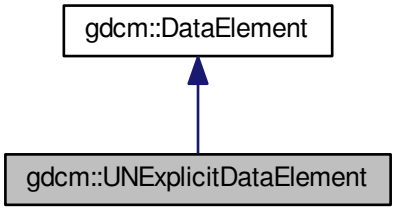
- gdcmULWritingCallback.h

27.328 gdcm::UNExplicitDataElement Class Reference

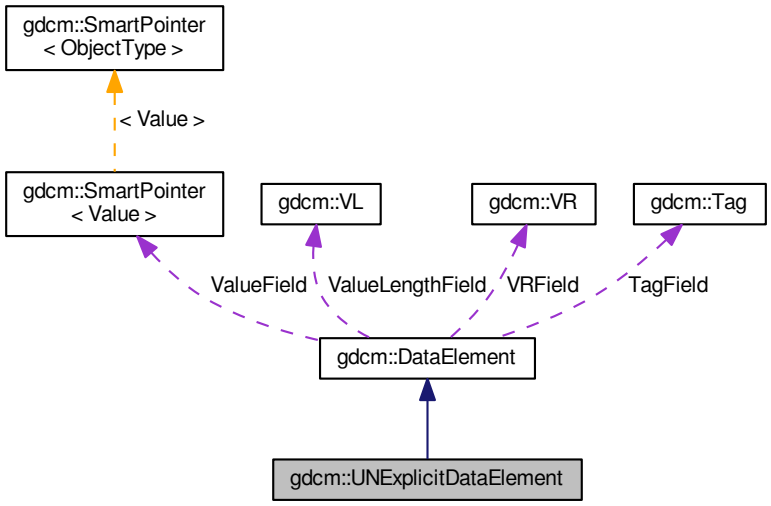
Class to read/write a DataElement as UNExplicit Data Element.

`#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)

Additional Inherited Members

27.328.1 Detailed Description

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

Note

bla

27.328.2 Member Function Documentation

27.328.2.1 VL gdcm::UNExplicitDataElement::GetLength ( ) const

27.328.2.2 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::Read (std::istream & *is*)

27.328.2.3 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadPreValue (std::istream & *is*)

27.328.2.4 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadValue (std::istream & *is*, bool *readvalues* = `true`)

27.328.2.5 template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadWithLength (std::istream & *is*, VL & *length*)

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

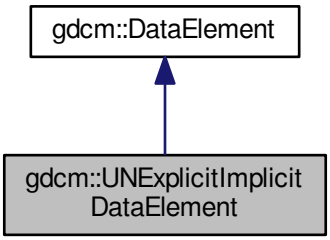
- gdcmUNExplicitDataElement.h

27.329 gdcm::UNExplicitImplicitDataElement Class Reference

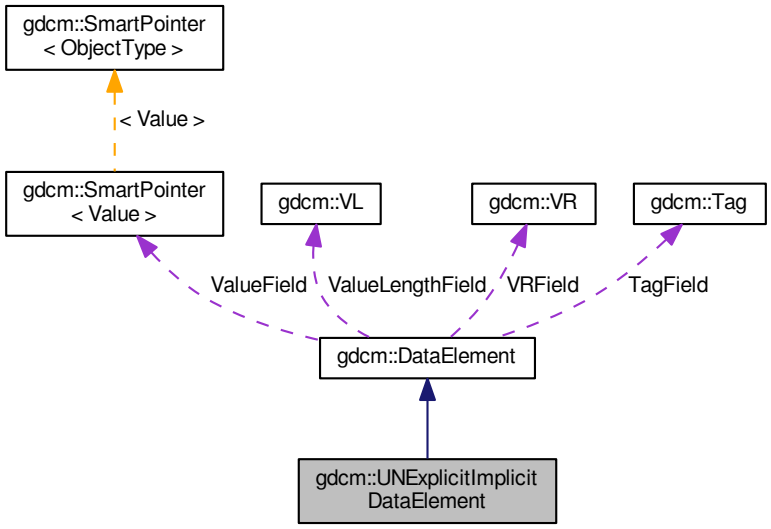
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:

```
#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)

Additional Inherited Members

27.329.1 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

27.329.2 Member Function Documentation

27.329.2.1 VL gdcm::UNExplicitImplicitDataElement::GetLength ( ) const

27.329.2.2 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::Read (std::istream & *is*)

27.329.2.3 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue (std::istream & *is*)

27.329.2.4 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadValue (std::istream & *is*)

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

- gdcmUNExplicitImplicitDataElement.h

27.330 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)

27.330.1 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

27.330.2 Member Function Documentation

27.330.2.1 static 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

27.330.2.2 static 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

27.331 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)

27.331.1 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.

27.331.2 Member Enumeration Documentation

27.331.2.1 enum gdcm::Usage::UsageType

Enumerator

Mandatory

Conditional

UserOption

Invalid

27.331.3 Constructor & Destructor Documentation

27.331.3.1 gdcm::Usage::Usage (UsageType *type* = Invalid) [inline]

27.331.4 Member Function Documentation

27.331.4.1 static const char* gdcm::Usage::GetUsageString (UsageType *type*) [static]

Referenced by gdcm::operator<<().

27.331.4.2 static UsageType gdcm::Usage::GetUsageType (const char * *type*) [static]

27.331.4.3 gdcm::Usage::operator UsageType ( ) const [inline]

27.331.5 Friends And Related Function Documentation

27.331.5.1 std::ostream& operator<< (std::ostream & *os*, const Usage & *vr*) [friend]

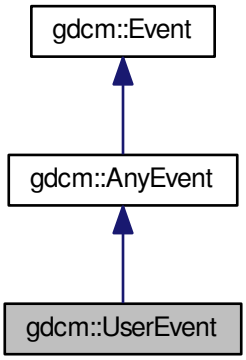
The documentation for this class was generated from the following file:

- gdcmUsage.h

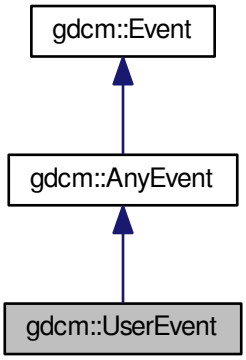
27.332 gdcm::UserEvent Class Reference

`#include <gdcmEvent.h>`

Inheritance diagram for gdcm::UserEvent:



Collaboration diagram for gdcm::UserEvent:



Additional Inherited Members

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

- gdcmEvent.h

27.333 gdcm::network::UserInformation Class Reference

UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

`#include <gdcmUserInformation.h>`

Public Member Functions

- UserInformation ()
- ~UserInformation ()
- void AddRoleSelectionSub (RoleSelectionSub const &r)
- void AddSOPClassExtendedNegociationSub (SOPClassExtendedNegociationSub const &s)
- const MaximumLengthSub & GetMaximumLengthSub () const
- MaximumLengthSub & GetMaximumLengthSub ()
- 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

27.333.1 Detailed Description

UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

TODO what is the goal of :

Table 9-20 USER INFORMATION ITEM FIELDS

27.333.2 Constructor & Destructor Documentation

27.333.2.1 gdcm::network::UserInformation::UserInformation ( )

27.333.2.2 gdcm::network::UserInformation::~UserInformation ( )

27.333.3 Member Function Documentation

27.333.3.1 void gdcm::network::UserInformation::AddRoleSelectionSub (RoleSelectionSub const & *r*)

27.333.3.2 void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (SOPClassExtendedNegociation↩ Sub const & *s*)

27.333.3.3 const MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ( ) const [inline]

27.333.3.4 MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ( ) [inline]

27.333.3.5 UserInformation& gdcm::network::UserInformation::operator= (const UserInformation &)

27.333.3.6 void gdcm::network::UserInformation::Print (std::ostream & *os*) const

27.333.3.7 std::istream& gdcm::network::UserInformation::Read (std::istream & *is*)

27.333.3.8 size_t gdcm::network::UserInformation::Size ( ) const

27.333.3.9 const std::ostream& gdcm::network::UserInformation::Write (std::ostream & *os*) const

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

- gdcmUserInformation.h

27.334 gdcm::UUIDGenerator Class Reference

Class for generating unique UUID generate DCE 1.1 uid.

`#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.

27.334.1 Detailed Description

Class for generating unique UUID generate DCE 1.1 uid.

27.334.2 Member Function Documentation

27.334.2.1 const char* gdcm::UUIDGenerator::Generate ( )

Return the generated uuid NOT THREAD SAFE

27.334.2.2 static 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:

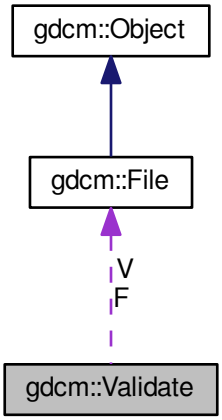
- gdcmUUIDGenerator.h

27.335 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

27.335.1 Detailed Description

Validate class.

27.335.2 Constructor & Destructor Documentation

27.335.2.1 gdcm::Validate::Validate ( )

27.335.2.2 gdcm::Validate::~Validate ( )

27.335.3 Member Function Documentation

27.335.3.1 const File& gdcm::Validate::GetValidatedFile ( ) [inline]

27.335.3.2 void gdcm::Validate::SetFile (File const & *f*) [inline]

27.335.3.3 void gdcm::Validate::Validation ( )

27.335.4 Member Data Documentation

27.335.4.1 const File* gdcm::Validate::F [protected]

27.335.4.2 File gdcm::Validate::V [protected]

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

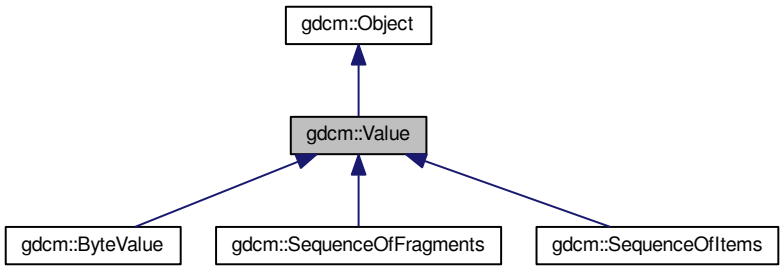
- gdcmValidate.h

27.336 gdcm::Value Class Reference

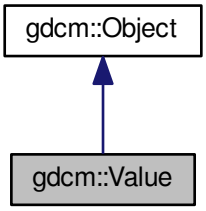
Class to represent the value of a Data Element.

`#include <gdcmValue.h>`

Inheritance diagram for gdcm::Value:



Collaboration diagram for gdcm::Value:



Public Member Functions

- Value ()
- ~Value ()
- virtual void Clear ()=0
- virtual VL GetLength () const =0
- virtual bool operator== (const Value &val) const =0
- virtual void SetLength (VL l)=0

Protected Member Functions

- virtual void SetLengthOnly (VL l)

Friends

- class DataElement

27.336.1 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.

27.336.2 Constructor & Destructor Documentation

27.336.2.1 gdcm::Value::Value ( ) [inline]

27.336.2.2 gdcm::Value::~Value ( ) [inline]

27.336.3 Member Function Documentation

27.336.3.1 virtual void gdcm::Value::Clear ( ) [pure virtual]

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

27.336.3.2 virtual VL gdcm::Value::GetLength ( ) const [pure virtual]

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

Referenced by gdcm::DataSet::InsertDataElement(), and gdcm::DataElement::SetValue().

27.336.3.3 virtual bool gdcm::Value::operator== (const Value & *val*) const [pure virtual]

Implemented in gdcm::SequenceOfFragments, gdcm::SequenceOfItems, and gdcm::ByteValue.

27.336.3.4 virtual void gdcm::Value::SetLength (VL /) [pure virtual]

Implemented in gdcm::ByteValue, gdcm::SequenceOfItems, and gdcm::SequenceOfFragments.

27.336.3.5 virtual void gdcm::Value::SetLengthOnly (VL /) [protected], [virtual]

Reimplemented in gdcm::ByteValue.

27.336.4 Friends And Related Function Documentation

27.336.4.1 friend class DataElement [friend]

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

- gdcmValue.h

27.337 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)

27.337.1 Detailed Description

template<typename TDE, typename TSwap, typename TType = uint8_t>class gdcm::ValueIO< TDE, TSwap, TType >

Class to dispatch template calls.

27.337.2 Member Function Documentation

27.337.2.1 template<typename TDE , typename TSwap , typename TType = uint8_t> static std::istream& gdcm::ValueIO< TDE, TSwap, TType >::Read (std::istream & *is,* Value & *v,* bool *readvalues*) [static]

27.337.2.2 template<typename TDE , typename TSwap , typename TType = uint8_t> static 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

27.338 gdcm::Version Class Reference

major/minor and build version

`#include <gdcmVersion.h>`

Public Member Functions

- Version ()
- ~Version ()
- 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)

27.338.1 Detailed Description

major/minor and build version

27.338.2 Constructor & Destructor Documentation

27.338.2.1 gdcm::Version::Version ( ) [inline]

27.338.2.2 gdcm::Version::~Version ( ) [inline]

27.338.3 Member Function Documentation

27.338.3.1 static int gdcm::Version::GetBuildVersion ( ) [static]

27.338.3.2 static int gdcm::Version::GetMajorVersion ( ) [static]

27.338.3.3 static int gdcm::Version::GetMinorVersion ( ) [static]

27.338.3.4 static const char* gdcm::Version::GetVersion ( ) [static]

27.338.3.5 void gdcm::Version::Print (std::ostream & *os* = `std::cout`) const

Referenced by gdcm::operator<<().

27.338.4 Friends And Related Function Documentation

27.338.4.1 std::ostream& operator<< (std::ostream & *_os,* const Version & *v*) [friend]

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

- gdcmVersion.h

27.339 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)

27.339.1 Detailed Description

Value Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

Examples:

rle2img.cxx.

27.339.2 Member Typedef Documentation

27.339.2.1 typedef uint32_t gdcm::VL::Type

27.339.3 Constructor & Destructor Documentation

27.339.3.1 gdcm::VL::VL (uint32_t *vl* = 0) [inline]

27.339.4 Member Function Documentation

27.339.4.1 VL gdcm::VL::GetLength ( ) const [inline]

Referenced by gdcm::FileMetaInformation::GetFullLength(), and gdcm::Item::Write().

27.339.4.2 static uint16_t gdcm::VL::GetVL16Max ( ) [inline], [static]

27.339.4.3 static uint32_t gdcm::VL::GetVL32Max ( ) [inline], [static]

27.339.4.4 bool gdcm::VL::IsOdd ( ) const [inline]

Return whether or not the VL is odd or not.

Referenced by gdcm::ByteValue::SetLength().

27.339.4.5 bool gdcm::VL::IsUndefined ( ) const [inline]

Referenced by gdcm::ByteValue::SetLength().

27.339.4.6 gdcm::VL::operator uint32_t ( ) const [inline]

27.339.4.7 VL& gdcm::VL::operator++ ( ) [inline]

27.339.4.8 VL gdcm::VL::operator++ (int) [inline]

27.339.4.9 VL& gdcm::VL::operator+= (VL const & *vl*) [inline]

+= operator

27.339.4.10 template<typename TSwap > std::istream& gdcm::VL::Read (std::istream & *is*) [inline]

27.339.4.11 template<typename TSwap > std::istream& gdcm::VL::Read16 (std::istream & *is*) [inline]

27.339.4.12 void gdcm::VL::SetToUndefined ( ) [inline]

27.339.4.13 template<typename TSwap > const std::ostream& gdcm::VL::Write (std::ostream & *os*) const [inline]

Referenced by gdcm::Fragment::Write(), gdcm::SequenceOfItems::Write(), gdcm::Item::Write(), and gdcm::SequenceOfFragments::Write().

27.339.4.14 template<typename TSwap > const std::ostream& gdcm::VL::Write16 (std::ostream & *os*) const [inline]

27.339.5 Friends And Related Function Documentation

27.339.5.1 std::ostream& operator<< (std::ostream & *os,* const VL & *vl*) [friend]

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

- gdcmVL.h

27.340 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,
  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 unsigned int GetNumberOfElementsFromArray (const char *array, unsigned int length)

- static const char * GetVMString (VMType vm)
- static VMType GetVMType (const char *vm)
- static VMType GetVMTypeFromLength (unsigned int 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)

27.340.1 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

27.340.2 Member Enumeration Documentation

27.340.2.1 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
VM7_7n
VM30_30n
VM47_47n
VM_END

27.340.3 Constructor & Destructor Documentation

27.340.3.1 gdcm::VM::VM (VMType *type* = VM0) [inline]

27.340.4 Member Function Documentation

27.340.4.1 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.

27.340.4.2 static unsigned int gdcm::VM::GetIndex (VMType *vm*) [static], [protected]

27.340.4.3 unsigned int gdcm::VM::GetLength ( ) const

27.340.4.4 static unsigned int gdcm::VM::GetNumberOfElementsFromArray (const char * *array,* unsigned int *length*) [static]

27.340.4.5 static 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 gdcm::operator<<().

27.340.4.6 static VMType gdcm::VM::GetVMType (const char * *vm*) [static]

27.340.4.7 static VMType gdcm::VM::GetVMTypeFromLength (unsigned int *length,* unsigned int *size*) [static]

27.340.4.8 static 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

27.340.4.9 gdcm::VM::operator VMType ( ) const [inline]

27.340.5 Friends And Related Function Documentation

27.340.5.1 std::ostream& operator<< (std::ostream & *os,* const VM & *vm*) [friend]

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

- gdcmVM.h

27.341 gdcm::VMToLength< T > Struct Template Reference

`#include <gdcmVM.h>`

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

- gdcmVM.h

27.342 gdcm::VR Class Reference

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.

`#include <gdcmVR.h>`

Public Types

- enum VRType {
  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,
  OW = 16384,
  PN = 32768,
  SH = 65536,
  SL = 131072,
  SQ = 262144,
  SS = 524288,
  ST = 1048576,
  TM = 2097152,
  UI = 4194304,
  UL = 8388608,
  UN = 16777216,
  US = 33554432,
  UT = 67108864,
  OB_OW = OB | OW,
  US_SS = US | SS,
  US_SS_OW = US | SS | 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 | SQ | UN | UT,
  VRASCII = AE | AS | CS | DA | DS | DT | IS | LO | LT | PN | SH | ST | TM | UI | UT,
  VRBINARY = AT | FL | FD | OB | OD | OF | OW | SL | SQ | SS | UL | UN | US,
  VR_VM1 = AS | LT | ST | UT | SQ | OF | OD | OW | OB | UN,
  VRALL = VRASCII | VRBINARY,
  VR_END = UT+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)

27.342.1 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, and GenFakeIdentifyFile.cxx.

27.342.2 Member Enumeration Documentation

27.342.2.1 enum gdcm::VR::VRType

Enumerator

INVALID

AE

AS

AT

CS

DA

DS

DT

FD

FL

IS

LO

LT

OB

OD

OF

OW

PN

SH

SL

SQ

SS

ST

TM

UI

UL

UN

US

UT

OB_OW

US_SS

US_SS_OW

VL16

VL32

VRASCII

VRBINARY

VR_VM1

VRALL

VR_END

Examples:

NewSequence.cs.

27.342.3 Constructor & Destructor Documentation

27.342.3.1 gdcm::VR::VR (VRType *vr* = INVALID) [inline]

27.342.4 Member Function Documentation

27.342.4.1 static bool gdcm::VR::CanDisplay (VRType *vr*) [static]

27.342.4.2 bool gdcm::VR::Compatible (VR const & *vr*) const

27.342.4.3 int gdcm::VR::GetLength ( ) const [inline]

27.342.4.4 static uint32_t gdcm::VR::GetLength (VRType *vr*) [inline], [static]

27.342.4.5 unsigned int gdcm::VR::GetSize ( ) const [inline]

References AE, US_SS, and VRTypeTemplateCase.

27.342.4.6 unsigned int gdcm::VR::GetSizeof ( ) const

27.342.4.7 static const char* gdcm::VR::GetVRString (VRType *vr*) [static]

Referenced by gdcm::operator<<().

27.342.4.8 static const char* gdcm::VR::GetVRStringFromFile (VRType *vr*) [static]

27.342.4.9 static VRType gdcm::VR::GetVRType (const char * *vr*) [static]

27.342.4.10 static VRType gdcm::VR::GetVRTypeFromFile (const char * *vr*) [static]

27.342.4.11 static bool gdcm::VR::IsASCII (VRType *vr*) [static]

27.342.4.12 static bool gdcm::VR::IsASCII2 (VRType *vr*) [static]

27.342.4.13 static bool gdcm::VR::IsBinary (VRType *vr*) [static]

27.342.4.14 static bool gdcm::VR::IsBinary2 (VRType *vr*) [static]

27.342.4.15 bool gdcm::VR::IsDual ( ) const

27.342.4.16 static bool gdcm::VR::IsSwap (const char * *vr*) [static]

27.342.4.17 static bool gdcm::VR::IsValid (const char * *vr*) [static]

27.342.4.18 static bool gdcm::VR::IsValid (const char * *vr1*, VRType *vr2*) [static]

27.342.4.19 bool gdcm::VR::IsVRFile ( ) const

Referenced by gdcm::DataElement::SetVR().

27.342.4.20 gdcm::VR::operator VRType ( ) const [inline]

27.342.4.21 std::istream& gdcm::VR::Read (std::istream & *is*) [inline]

References gdcmDebugMacro, INVALID, and VR_END.

27.342.4.22 const std::ostream& gdcm::VR::Write (std::ostream & *os*) const [inline]

References gdcmAssertAlwaysMacro, and INVALID.

27.342.5 Friends And Related Function Documentation

27.342.5.1 std::ostream& operator<< (std::ostream & *os,* const VR & *vr*) [friend]

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

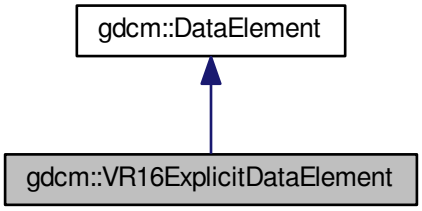
- gdcmVR.h

27.343 gdcm::VR16ExplicitDataElement Class Reference

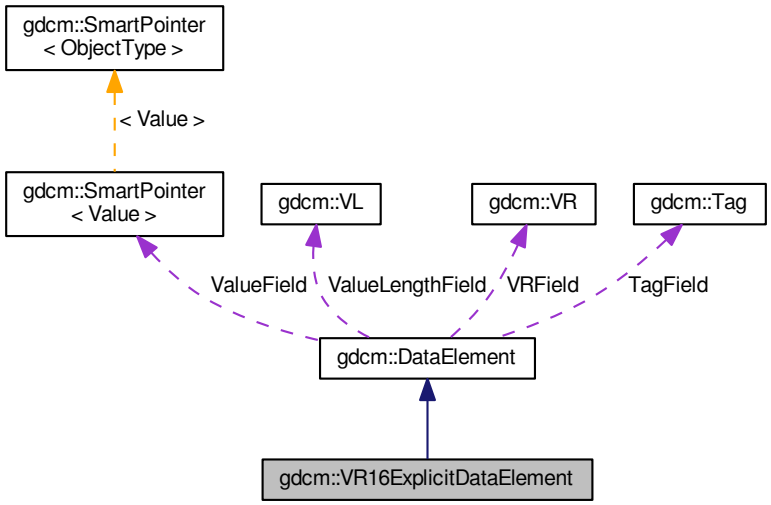
Class to read/write a DataElement as Explicit Data Element.

`#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)

Additional Inherited Members

27.343.1 Detailed Description

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

Note

This class support 16 bits when finding an unkown VR: For instance: Siemens_CT_Sensation64_has_VR_RT.dcm

27.343.2 Member Function Documentation

27.343.2.1 VL gdcm::VR16ExplicitDataElement::GetLength ( ) const

27.343.2.2 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::Read (std::istream & *is*)

27.343.2.3 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (std::istream & *is*)

27.343.2.4 template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadValue (std::istream & *is*, bool *readvalues* = `true`)

27.343.2.5 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

27.344 gdcm::VRToEncoding< T > Struct Template Reference

`#include <gdcmVR.h>`

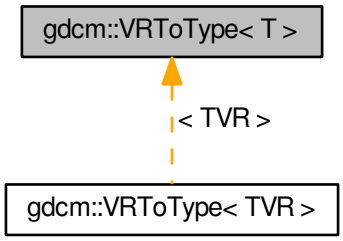
The documentation for this struct was generated from the following file:

- gdcmVR.h

27.345 gdcm::VRToType< T > Struct Template Reference

`#include <gdcmVR.h>`

Inheritance diagram for gdcm::VRToType< T >:



27.345.1 Detailed Description

template<int T>struct gdcm::VRToType< T >

Examples:

DumpGEMSMovieGroup.cxx.

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

- gdcmVR.h

27.346 gdcm::VRVLSize< T > Singleton Reference

`#include <gdcmAttribute.h>`

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

- gdcmAttribute.h

27.347 gdcm::VRVLSize< 0 > Class Template Reference

`#include <gdcmAttribute.h>`

Static Public Member Functions

- static uint16_t Read (std::istream &_is)
- static void Write (std::ostream &os)

27.347.1 Member Function Documentation

27.347.1.1 static uint16_t gdcm::VRVLSize< 0 >::Read (std::istream & *_is*) [inline], [static]

27.347.1.2 static void gdcm::VRVLSize< 0 >::Write (std::ostream & *os*) [inline], [static]

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

- gdcmAttribute.h

27.348 gdcm::VRVLSize< 1 > Class Template Reference

`#include <gdcmAttribute.h>`

Static Public Member Functions

- static uint32_t Read (std::istream &_is)
- static void Write (std::ostream &os)

27.348.1 Member Function Documentation

27.348.1.1 static uint32_t gdcm::VRVLSize< 1 >::Read (std::istream & *_is*) [inline], [static]

27.348.1.2 static void gdcm::VRVLSize< 1 >::Write (std::ostream & *os*) [inline], [static]

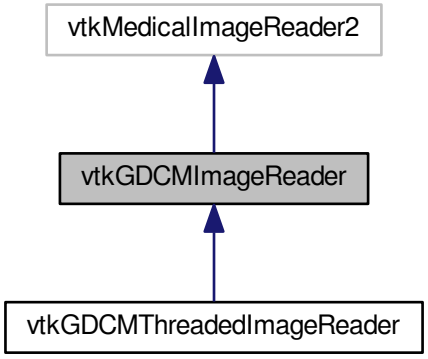
The documentation for this class was generated from the following file:

- gdcmAttribute.h

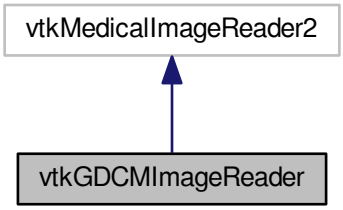
27.349 vtkGDCMImageReader Class Reference

`#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 (LoadOverlays, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LoadIconImage, int)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (ApplyLookupTable, int)
- vtkTypeRevisionMacro (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 (FilePrefix)
- vtkGetStringMacro (FilePattern)
- 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

27.349.1 Detailed Description

Examples:

AWTMedical3.java, Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, HelloVTKWorld.java, MagnifyFile.cxx, MetaImageMD5Activiz.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, offscreenimage.cxx, ReadSeriesIntoVTK.java, RefCounting.cs, and reslicesphere.cxx.

27.349.2 Constructor & Destructor Documentation

27.349.2.1 vtkGDCMImageReader::vtkGDCMImageReader ( ) [protected]

Examples:

HelloActiviz2.cs.

27.349.2.2 vtkGDCMImageReader::~vtkGDCMImageReader ( ) [protected]

27.349.3 Member Function Documentation

27.349.3.1 virtual int vtkGDCMImageReader::CanReadFile (const char * *fname*) [virtual]

Examples:

MetaImageMD5Activiz.cs.

27.349.3.2 void vtkGDCMImageReader::ExecuteData (vtkDataObject * *out*) [protected]

27.349.3.3 void vtkGDCMImageReader::ExecuteInformation ( ) [protected]

27.349.3.4 void vtkGDCMImageReader::FillMedicalImageInformation (const **gdcm::ImageReader** & *reader*) [protected]

27.349.3.5 virtual const char* vtkGDCMImageReader::GetDescriptiveName ( ) [inline], [virtual]

27.349.3.6 virtual const char* vtkGDCMImageReader::GetFileExtensions ( ) [inline], [virtual]

27.349.3.7 vtkImageData* vtkGDCMImageReader::GetIconImage ( )

27.349.3.8 vtkImageData* vtkGDCMImageReader::GetOverlay (int *i*)

27.349.3.9 int vtkGDCMImageReader::LoadSingleFile (const char * *filename,* char * *pointer,* unsigned long & *outlen*) [protected]

27.349.3.10 static **vtkGDCMImageReader*** vtkGDCMImageReader::New ( ) [static]

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, gdcmreslice.cxx, gdcmtexture.cxx, gdcmvolume.cxx, HelloActiviz.cs, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, HelloVTKWorld.cs, MagnifyFile.cxx, MetaImageMD5Activiz.cs, offscreenimage.cxx, RefCounting.cs, and reslicesphere.cxx.

27.349.3.11 virtual void vtkGDCMImageReader::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

Reimplemented in vtkGDCMThreadedImageReader.

27.349.3.12 int vtkGDCMImageReader::RequestDataCompat ( ) [protected]

27.349.3.13 int vtkGDCMImageReader::RequestInformationCompat ( ) [protected]

27.349.3.14 virtual void vtkGDCMImageReader::SetCurve (vtkPolyData * *pd*) [virtual]

27.349.3.15 virtual void vtkGDCMImageReader::SetFileNames (vtkStringArray *) [virtual]

Examples:

gdcmorthoplanes.cxx, HelloActiviz3.cs, HelloActiviz4.cs, HelloActiviz5.cs, MIPViewer.java, MPRViewer.java, MPRViewer2.java, and ReadSeriesIntoVTK.java.

27.349.3.16 void vtkGDCMImageReader::SetFilePattern (const char *) [inline], [protected]

27.349.3.17 void vtkGDCMImageReader::SetFilePrefix (const char *) [inline], [protected]

27.349.3.18 virtual void vtkGDCMImageReader::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]

27.349.3.19 vtkGDCMImageReader::vtkBooleanMacro (**LoadOverlays** , int)

27.349.3.20 vtkGDCMImageReader::vtkBooleanMacro (**LoadIconImage** , int)

27.349.3.21 vtkGDCMImageReader::vtkBooleanMacro (**LossyFlag** , int)

27.349.3.22 vtkGDCMImageReader::vtkBooleanMacro (**ApplyLookupTable** , int)

27.349.3.23 int vtkGDCMImageReader::vtkBooleanMacro (**ApplyYBRToRGB** , int)

27.349.3.24 vtkGDCMImageReader::vtkGetMacro (**LoadOverlays** , int)

27.349.3.25 vtkGDCMImageReader::vtkGetMacro (**LoadIconImage** , int)

27.349.3.26 vtkGDCMImageReader::vtkGetMacro (**LossyFlag** , int)

27.349.3.27 vtkGDCMImageReader::vtkGetMacro (**NumberOfOverlays** , int)

27.349.3.28 vtkGDCMImageReader::vtkGetMacro (**NumberOfIconImages** , int)

27.349.3.29 vtkGDCMImageReader::vtkGetMacro (**ApplyLookupTable** , int)

27.349.3.30 vtkGDCMImageReader::vtkGetMacro (**ApplyYBRToRGB** , int)

27.349.3.31 vtkGDCMImageReader::vtkGetMacro (**ImageFormat** , int)

27.349.3.32 vtkGDCMImageReader::vtkGetMacro (**PlanarConfiguration** , int)

27.349.3.33 vtkGDCMImageReader::vtkGetMacro (**Shift** , double)

27.349.3.34 vtkGDCMImageReader::vtkGetMacro (**Scale** , double)

27.349.3.35 vtkGDCMImageReader::vtkGetObjectMacro (**DirectionCosines** , vtkMatrix4x4)

27.349.3.36 vtkGDCMImageReader::vtkGetObjectMacro (**MedicalImageProperties** , vtkMedicalImageProperties)

27.349.3.37 vtkGDCMImageReader::vtkGetObjectMacro (**FileNames** , vtkStringArray)

27.349.3.38 vtkGDCMImageReader::vtkGetObjectMacro (Curve , vtkPolyData)

27.349.3.39 vtkGDCMImageReader::vtkGetStringMacro (FilePrefix) [protected]

27.349.3.40 vtkGDCMImageReader::vtkGetStringMacro (FilePattern) [protected]

27.349.3.41 vtkGDCMImageReader::vtkGetVector3Macro (ImagePositionPatient , double)

27.349.3.42 vtkGDCMImageReader::vtkGetVector6Macro (ImageOrientationPatient , double)

27.349.3.43 vtkGDCMImageReader::vtkSetMacro (LoadOverlays , int)

27.349.3.44 vtkGDCMImageReader::vtkSetMacro (LoadIconImage , int)

27.349.3.45 vtkGDCMImageReader::vtkSetMacro (LossyFlag , int)

27.349.3.46 vtkGDCMImageReader::vtkSetMacro (ApplyLookupTable , int)

27.349.3.47 vtkGDCMImageReader::vtkSetVector6Macro (ImageOrientationPatient , double) [protected]

27.349.3.48 vtkGDCMImageReader::vtkTypeRevisionMacro (vtkGDCMImageReader , vtkMedicalImageReader2)

27.349.4 Member Data Documentation

27.349.4.1 int vtkGDCMImageReader::ApplyInverseVideo [protected]

27.349.4.2 int vtkGDCMImageReader::ApplyLookupTable [protected]

27.349.4.3 int vtkGDCMImageReader::ApplyPlanarConfiguration [protected]

27.349.4.4 int vtkGDCMImageReader::ApplyShiftScale [protected]

27.349.4.5 int vtkGDCMImageReader::ApplyYBRToRGB [protected]

27.349.4.6 vtkPolyData* vtkGDCMImageReader::Curve [protected]

27.349.4.7 vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines [protected]

27.349.4.8 vtkStringArray* vtkGDCMImageReader::FileNames [protected]

27.349.4.9 int vtkGDCMImageReader::ForceRescale [protected]

27.349.4.10 int vtkGDCMImageReader::IconDataScalarType [protected]

27.349.4.11 int vtkGDCMImageReader::IconImageDataExtent[6] [protected]

27.349.4.12 int vtkGDCMImageReader::IconNumberOfScalarComponents [protected]

27.349.4.13 int vtkGDCMImageReader::ImageFormat [protected]

27.349.4.14 double vtkGDCMImageReader::ImageOrientationPatient[6] [protected]

27.349.4.15 double vtkGDCMImageReader::ImagePositionPatient[3] [protected]

27.349.4.16 int vtkGDCMImageReader::LoadIconImage [protected]

27.349.4.17 int vtkGDCMImageReader::LoadOverlays [protected]

27.349.4.18 int vtkGDCMImageReader::LossyFlag [protected]

27.349.4.19 vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties [protected]

27.349.4.20 int vtkGDCMImageReader::NumberOfIconImages [protected]

27.349.4.21 int vtkGDCMImageReader::NumberOfOverlays [protected]

27.349.4.22 int vtkGDCMImageReader::PlanarConfiguration [protected]

27.349.4.23 double vtkGDCMImageReader::Scale [protected]

27.349.4.24 double vtkGDCMImageReader::Shift [protected]

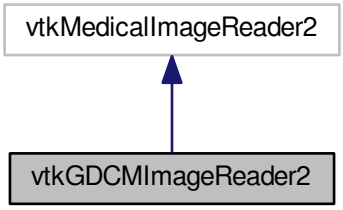
The documentation for this class was generated from the following file:

- vtkGDCMImageReader.h

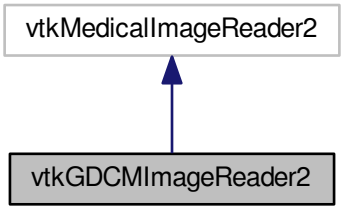
27.350 vtkGDCMImageReader2 Class Reference

`#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 (LoadOverlays, int)
- vtkBooleanMacro (LoadIconImage, int)
- vtkBooleanMacro (LossyFlag, int)
- vtkBooleanMacro (ApplyLookupTable, int)
- int vtkBooleanMacro (ApplyYBRToRGB, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (LoadIconImage, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (NumberOfIconImages, int)
- vtkGetMacro (ApplyLookupTable, int)
- vtkGetMacro (ApplyYBRToRGB, int) vtkSetMacro(ApplyYBRToRGB
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetObjectMacro (Curve, vtkPolyData)
- vtkGetVector3Macro (ImagePositionPatient, double)
- vtkGetVector6Macro (ImageOrientationPatient, double)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (LoadIconImage, int)

- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (ApplyLookupTable, int)
- vtkTypeRevisionMacro (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 (FilePrefix)
- vtkGetStringMacro (FilePattern)
- 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

27.350.1 Detailed Description

Examples:

Compute3DSpacing.cxx.

27.350.2 Constructor & Destructor Documentation

27.350.2.1 vtkGDCMImageReader2::vtkGDCMImageReader2 ( ) [protected]

27.350.2.2 vtkGDCMImageReader2::~vtkGDCMImageReader2 ( ) [protected]

27.350.3 Member Function Documentation

27.350.3.1 virtual int vtkGDCMImageReader2::CanReadFile (const char * *fname*) [virtual]

27.350.3.2 void vtkGDCMImageReader2::FillMedicalImageInformation (const gdcm::ImageReader & *reader*) [protected]

27.350.3.3 virtual const char* vtkGDCMImageReader2::GetDescriptiveName ( ) [inline], [virtual]

27.350.3.4 virtual const char* vtkGDCMImageReader2::GetFileExtensions ( ) [inline], [virtual]

27.350.3.5 vtkImageData* vtkGDCMImageReader2::GetIconImage ( )

27.350.3.6 vtkAlgorithmOutput* vtkGDCMImageReader2::GetIconImagePort ( )

27.350.3.7 vtkImageData* vtkGDCMImageReader2::GetOverlay (int *i*)

27.350.3.8 vtkAlgorithmOutput* vtkGDCMImageReader2::GetOverlayPort (int *index*)

27.350.3.9 int vtkGDCMImageReader2::LoadSingleFile (const char * *filename,* char * *pointer,* unsigned long & *outlen*) [protected]

27.350.3.10 static vtkGDCMImageReader2* vtkGDCMImageReader2::New ( ) [static]

Examples:

Compute3DSpacing.cxx.

27.350.3.11 virtual void vtkGDCMImageReader2::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

27.350.3.12 int vtkGDCMImageReader2::ProcessRequest (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected]

27.350.3.13 int vtkGDCMImageReader2::RequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected]

27.350.3.14 int vtkGDCMImageReader2::RequestDataCompat ( ) [protected]

27.350.3.15 int vtkGDCMImageReader2::RequestInformation (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected]

27.350.3.16 int vtkGDCMImageReader2::RequestInformationCompat ( ) [protected]

27.350.3.17 virtual void vtkGDCMImageReader2::SetCurve (vtkPolyData * *pd*) [virtual]

27.350.3.18 void vtkGDCMImageReader2::SetFilePattern (const char *) [inline], [protected]

27.350.3.19 void vtkGDCMImageReader2::SetFilePrefix (const char *) [inline], [protected]

27.350.3.20 virtual void vtkGDCMImageReader2::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]

27.350.3.21 vtkGDCMImageReader2::vtkBooleanMacro (**LoadOverlays** , int)

27.350.3.22 vtkGDCMImageReader2::vtkBooleanMacro (**LoadIconImage** , int)

27.350.3.23 vtkGDCMImageReader2::vtkBooleanMacro (**LossyFlag** , int)

27.350.3.24 vtkGDCMImageReader2::vtkBooleanMacro (**ApplyLookupTable** , int)

27.350.3.25 int vtkGDCMImageReader2::vtkBooleanMacro (**ApplyYBRToRGB** , int)

27.350.3.26 vtkGDCMImageReader2::vtkGetMacro (**LoadOverlays** , int)

27.350.3.27 vtkGDCMImageReader2::vtkGetMacro (**LoadIconImage** , int)

27.350.3.28 vtkGDCMImageReader2::vtkGetMacro (**LossyFlag** , int)

27.350.3.29 vtkGDCMImageReader2::vtkGetMacro (**NumberOfOverlays** , int)

27.350.3.30 vtkGDCMImageReader2::vtkGetMacro (**NumberOfIconImages** , int)

27.350.3.31 vtkGDCMImageReader2::vtkGetMacro (**ApplyLookupTable** , int)

27.350.3.32 vtkGDCMImageReader2::vtkGetMacro (**ApplyYBRToRGB** , int)

27.350.3.33 vtkGDCMImageReader2::vtkGetMacro (**ImageFormat** , int)

27.350.3.34 vtkGDCMImageReader2::vtkGetMacro (**PlanarConfiguration** , int)

27.350.3.35 vtkGDCMImageReader2::vtkGetMacro (**Shift** , double)

27.350.3.36 vtkGDCMImageReader2::vtkGetMacro (**Scale** , double)

27.350.3.37 vtkGDCMImageReader2::vtkGetObjectMacro (**DirectionCosines** , vtkMatrix4x4)

27.350.3.38 vtkGDCMImageReader2::vtkGetObjectMacro (**Curve** , vtkPolyData)

27.350.3.39 vtkGDCMImageReader2::vtkGetStringMacro (**FilePrefix**) [protected]

27.350.3.40 vtkGDCMImageReader2::vtkGetStringMacro (**FilePattern**) [protected]

27.350.3.41 vtkGDCMImageReader2::vtkGetVector3Macro (**ImagePositionPatient** , double)

27.350.3.42 vtkGDCMImageReader2::vtkGetVector6Macro (ImageOrientationPatient , double)

27.350.3.43 vtkGDCMImageReader2::vtkSetMacro (LoadOverlays , int)

27.350.3.44 vtkGDCMImageReader2::vtkSetMacro (LoadIconImage , int)

27.350.3.45 vtkGDCMImageReader2::vtkSetMacro (LossyFlag , int)

27.350.3.46 vtkGDCMImageReader2::vtkSetMacro (ApplyLookupTable , int)

27.350.3.47 vtkGDCMImageReader2::vtkSetVector6Macro (ImageOrientationPatient , double) [protected]

27.350.3.48 vtkGDCMImageReader2::vtkTypeRevisionMacro (vtkGDCMImageReader2 , vtkMedicalImageReader2)

27.350.4 Member Data Documentation

27.350.4.1 int vtkGDCMImageReader2::ApplyInverseVideo [protected]

27.350.4.2 int vtkGDCMImageReader2::ApplyLookupTable [protected]

27.350.4.3 int vtkGDCMImageReader2::ApplyPlanarConfiguration [protected]

27.350.4.4 int vtkGDCMImageReader2::ApplyShiftScale [protected]

27.350.4.5 int vtkGDCMImageReader2::ApplyYBRToRGB [protected]

27.350.4.6 vtkPolyData* vtkGDCMImageReader2::Curve [protected]

27.350.4.7 vtkMatrix4x4* vtkGDCMImageReader2::DirectionCosines [protected]

27.350.4.8 int vtkGDCMImageReader2::ForceRescale [protected]

27.350.4.9 int vtkGDCMImageReader2::IconDataScalarType [protected]

27.350.4.10 int vtkGDCMImageReader2::IconImageDataExtent[6] [protected]

27.350.4.11 int vtkGDCMImageReader2::IconNumberOfScalarComponents [protected]

27.350.4.12 int vtkGDCMImageReader2::ImageFormat [protected]

27.350.4.13 double vtkGDCMImageReader2::ImageOrientationPatient[6] [protected]

27.350.4.14 double vtkGDCMImageReader2::ImagePositionPatient[3] [protected]

27.350.4.15 int vtkGDCMImageReader2::LoadIconImage [protected]

27.350.4.16 int vtkGDCMImageReader2::LoadOverlays [protected]

27.350.4.17 int vtkGDCMImageReader2::LossyFlag [protected]

27.350.4.18 int vtkGDCMImageReader2::NumberOfIconImages [protected]

27.350.4.19 int vtkGDCMImageReader2::NumberOfOverlays [protected]

27.350.4.20 int vtkGDCMImageReader2::PlanarConfiguration [protected]

27.350.4.21 double vtkGDCMImageReader2::Scale [protected]

27.350.4.22 double vtkGDCMImageReader2::Shift [protected]

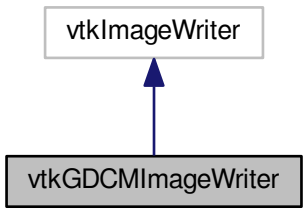
The documentation for this class was generated from the following file:

- vtkGDCMImageReader2.h

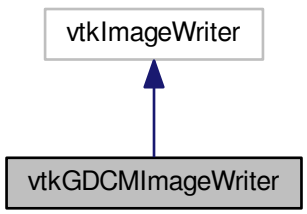
27.351 vtkGDCMImageWriter Class Reference

`#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 (LossyFlag, int)
- vtkBooleanMacro (FileLowerLeft, int)
- vtkGetMacro (LossyFlag, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetMacro (ImageFormat, int)
- vtkGetMacro (FileLowerLeft, int)
- vtkGetMacro (PlanarConfiguration, int)
- vtkGetMacro (CompressionType, int)
- vtkGetObjectMacro (MedicalImageProperties, vtkMedicalImageProperties)
- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetObjectMacro (DirectionCosines, vtkMatrix4x4)
- vtkGetStringMacro (StudyUID)
- vtkGetStringMacro (SeriesUID)
- vtkSetMacro (LossyFlag, int)
- vtkSetMacro (Shift, double)
- vtkSetMacro (Scale, double)
- vtkSetMacro (ImageFormat, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (PlanarConfiguration, int)
- vtkSetMacro (CompressionType, int)
- vtkSetStringMacro (StudyUID)
- vtkSetStringMacro (SeriesUID)
- vtkTypeRevisionMacro (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)

27.351.1 Detailed Description

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakeRTDOSE.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, MagnifyFile.cxx, and RefCounting.cs.

27.351.2 Member Enumeration Documentation

27.351.2.1 enum vtkGDCMImageWriter::CompressionTypes

Enumerator

NO_COMPRESSION

JPEG_COMPRESSION

JPEG2000_COMPRESSION

JPEGLS_COMPRESSION

RLE_COMPRESSION

27.351.3 Constructor & Destructor Documentation

27.351.3.1 vtkGDCMImageWriter::vtkGDCMImageWriter ( ) [protected]

27.351.3.2 vtkGDCMImageWriter::~vtkGDCMImageWriter ( ) [protected]

27.351.4 Member Function Documentation

27.351.4.1 virtual const char* vtkGDCMImageWriter::GetDescriptiveName ( ) [inline], [virtual]

27.351.4.2 virtual const char* vtkGDCMImageWriter::GetFileExtensions ( ) [inline], [virtual]

27.351.4.3 virtual char* vtkGDCMImageWriter::GetFileName ( ) [protected], [virtual]

27.351.4.4 static vtkGDCMImageWriter* vtkGDCMImageWriter::New ( ) [static]

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakeRTDOSE.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloVTKWorld.cs, HelloVTKWorld2.cs, MagnifyFile.cxx, and RefCounting.cs.

27.351.4.5 virtual void vtkGDCMImageWriter::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

27.351.4.6 virtual void vtkGDCMImageWriter::SetDirectionCosines (vtkMatrix4x4 * *matrix*) [virtual]

Examples:

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, and MagnifyFile.cxx.

27.351.4.7 virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (const double *dircos[6]*) [virtual]

27.351.4.8 virtual void vtkGDCMImageWriter::SetFileNames (vtkStringArray *) [virtual]

Examples:

ConvertMultiFrameToSingleFrame.cxx.

27.351.4.9 virtual void vtkGDCMImageWriter::SetMedicalImageProperties (vtkMedicalImageProperties *) [virtual]

Examples:

Convert16BitsTo8Bits.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, and MagnifyFile.cxx.

27.351.4.10 vtkGDCMImageWriter::vtkBooleanMacro (LossyFlag , int)

27.351.4.11 vtkGDCMImageWriter::vtkBooleanMacro (FileLowerLeft , int)

27.351.4.12 vtkGDCMImageWriter::vtkGetMacro (LossyFlag , int)

27.351.4.13 vtkGDCMImageWriter::vtkGetMacro (Shift , double)

27.351.4.14 vtkGDCMImageWriter::vtkGetMacro (Scale , double)

27.351.4.15 vtkGDCMImageWriter::vtkGetMacro (ImageFormat , int)

27.351.4.16 vtkGDCMImageWriter::vtkGetMacro (FileLowerLeft , int)

27.351.4.17 vtkGDCMImageWriter::vtkGetMacro (PlanarConfiguration , int)

27.351.4.18 vtkGDCMImageWriter::vtkGetMacro (CompressionType , int)

27.351.4.19 vtkGDCMImageWriter::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

27.351.4.20 vtkGDCMImageWriter::vtkGetObjectMacro (FileNames , vtkStringArray)

27.351.4.21 vtkGDCMImageWriter::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)

27.351.4.22 vtkGDCMImageWriter::vtkGetStringMacro (StudyUID)

27.351.4.23 vtkGDCMImageWriter::vtkGetStringMacro (SeriesUID)

27.351.4.24 vtkGDCMImageWriter::vtkSetMacro (LossyFlag , int)

27.351.4.25 vtkGDCMImageWriter::vtkSetMacro (Shift , double)

27.351.4.26 vtkGDCMImageWriter::vtkSetMacro (Scale , double)

27.351.4.27 vtkGDCMImageWriter::vtkSetMacro (ImageFormat , int)

27.351.4.28 vtkGDCMImageWriter::vtkSetMacro (FileLowerLeft , int)

27.351.4.29 vtkGDCMImageWriter::vtkSetMacro (PlanarConfiguration , int)

27.351.4.30 vtkGDCMImageWriter::vtkSetMacro (CompressionType , int)

27.351.4.31 vtkGDCMImageWriter::vtkSetStringMacro (StudyUID)

27.351.4.32 vtkGDCMImageWriter::vtkSetStringMacro (SeriesUID)

27.351.4.33 vtkGDCMImageWriter::vtkTypeRevisionMacro (**vtkGDCMImageWriter** , vtkImageWriter)

27.351.4.34 virtual void vtkGDCMImageWriter::Write ( ) [virtual]

Examples:

Convert16BitsTo8Bits.cxx, ConvertMultiFrameToSingleFrame.cxx, ConvertRGBToLuminance.cxx, ConvertSingleBitTo8Bits.cxx, CreateFakeRTDOSE.cxx, gdcmorthoplanes.cxx, HelloActiviz.cs, HelloActiviz2.cs, HelloVTKWorld.cs, HelloVTKWorld.java, HelloVTKWorld2.cs, and MagnifyFile.cxx.

27.351.4.35 int vtkGDCMImageWriter::WriteGDCMData (vtkImageData * *data*, int *timeStep*) [protected]

27.351.4.36 void vtkGDCMImageWriter::WriteSlice (vtkImageData * *data*) [protected]

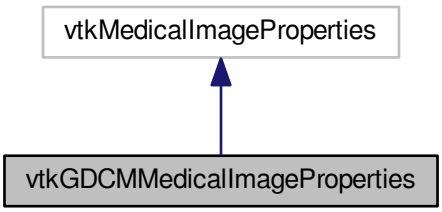
The documentation for this class was generated from the following file:

- vtkGDCMImageWriter.h

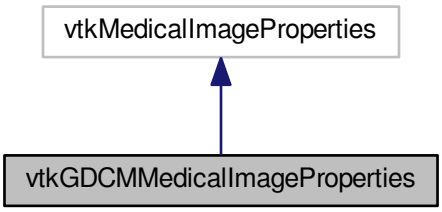
27.352 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

Inheritance diagram for vtkGDCMMedicalImageProperties:



Collaboration diagram for vtkGDCMMedicalImageProperties:



Public Member Functions

- virtual void Clear ()
- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (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

27.352.1 Constructor & Destructor Documentation

27.352.1.1 vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( ) [protected]

27.352.1.2 vtkGDCMMedicalImageProperties::~vtkGDCMMedicalImageProperties ( ) [protected]

27.352.2 Member Function Documentation

27.352.2.1 virtual void vtkGDCMMedicalImageProperties::Clear ( ) [virtual]

27.352.2.2 gdcm::File const& vtkGDCMMedicalImageProperties::GetFile (unsigned int *t*) [protected]

27.352.2.3 static vtkGDCMMedicalImageProperties* vtkGDCMMedicalImageProperties::New ( ) [static]

27.352.2.4 void vtkGDCMMedicalImageProperties::PrintSelf (ostream & *os*, vtkIndent *indent*)

27.352.2.5 void vtkGDCMMedicalImageProperties::PushBackFile (gdcm::File const & *f*) [protected]

27.352.2.6 vtkGDCMMedicalImageProperties::vtkTypeRevisionMacro (vtkGDCMMedicalImageProperties , vtkMedicalImageProperties)

27.352.3 Friends And Related Function Documentation

27.352.3.1 friend class vtkGDCMImageReader [friend]

27.352.3.2 friend class vtkGDCMImageReader2 [friend]

27.352.3.3 friend class vtkGDCMImageWriter [friend]

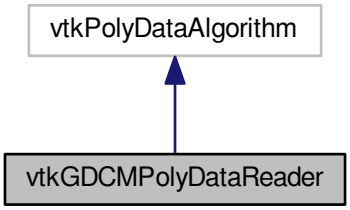
The documentation for this class was generated from the following file:

- vtkGDCMMedicalImageProperties.h

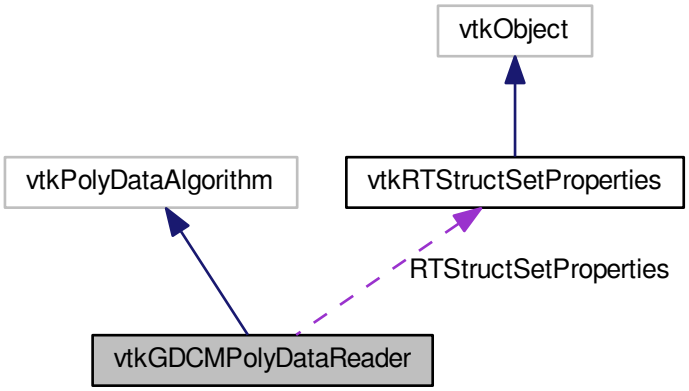
27.353 vtkGDCMPolyDataReader Class Reference

```
#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)
- vtkTypeRevisionMacro (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

27.353.1 Detailed Description

Examples:

gdcmscene.cxx, GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

27.353.2 Constructor & Destructor Documentation

27.353.2.1 vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( ) [protected]

27.353.2.2 vtkGDCMPolyDataReader::∼vtkGDCMPolyDataReader ( ) [protected]

27.353.3 Member Function Documentation

27.353.3.1 void vtkGDCMPolyDataReader::FillMedicalImageInformation (const gdcm::Reader & *reader*) [protected]

27.353.3.2 static vtkGDCMPolyDataReader∗ vtkGDCMPolyDataReader::New ( ) [static]

Examples:

gdcmscene.cxx, GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

27.353.3.3 virtual void vtkGDCMPolyDataReader::PrintSelf (ostream & *os*, vtkIndent *indent*) [virtual]

27.353.3.4 int vtkGDCMPolyDataReader::RequestData (vtkInformation ∗ , vtkInformationVector ∗∗ , vtkInformationVector ∗) [protected]

27.353.3.5 int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (gdcm::Reader const & *reader*, vtkInformationVector ∗ *outputVector*) [protected]

27.353.3.6 int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (gdcm::Reader const & *reader,* vtkInformationVector * *outputVector*) [protected]

27.353.3.7 int vtkGDCMPolyDataReader::RequestInformation (vtkInformation * *vtkNotUsedrequest,* vtkInformationVector ** *vtkNotUsedinputVector,* vtkInformationVector * *outputVector*) [protected]

27.353.3.8 int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const & *reader*) [protected]

27.353.3.9 int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (gdcm::Reader const & *reader*) [protected]

27.353.3.10 vtkGDCMPolyDataReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

27.353.3.11 vtkGDCMPolyDataReader::vtkGetObjectMacro (RTStructSetProperties , vtkRTStructSetProperties)

27.353.3.12 vtkGDCMPolyDataReader::vtkGetStringMacro (FileName)

27.353.3.13 vtkGDCMPolyDataReader::vtkSetStringMacro (FileName)

27.353.3.14 vtkGDCMPolyDataReader::vtkTypeRevisionMacro (vtkGDCMPolyDataReader , vtkPolyDataAlgorithm)

27.353.4 Member Data Documentation

27.353.4.1 char* vtkGDCMPolyDataReader::FileName [protected]

27.353.4.2 vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties [protected]

27.353.4.3 vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties [protected]

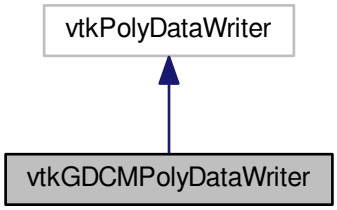
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataReader.h

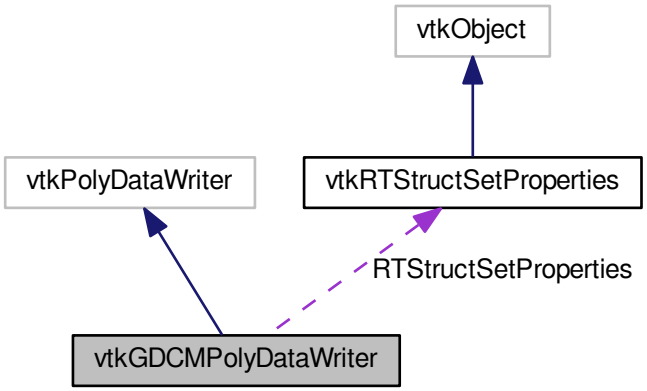
27.354 vtkGDCMPolyDataWriter Class Reference

```
#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)
- vtkTypeRevisionMacro (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

27.354.1 Detailed Description

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

27.354.2 Constructor & Destructor Documentation

27.354.2.1 vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( ) [protected]

27.354.2.2 vtkGDCMPolyDataWriter::~vtkGDCMPolyDataWriter ( ) [protected]

27.354.3 Member Function Documentation

27.354.3.1 void vtkGDCMPolyDataWriter::InitializeRTStructSet (vtkStdString *inDirectory,* vtkStdString *inStructLabel,* vtkStdString *inStructName,* vtkStringArray * *inROINames,* vtkStringArray * *inROIAlgorithmName,* vtkStringArray * *inROIType*)

Examples:

GenerateRTSTRUCT.cxx.

27.354.3.2 static vtkGDCMPolyDataWriter* vtkGDCMPolyDataWriter::New ( ) [static]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

27.354.3.3 virtual void vtkGDCMPolyDataWriter::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

27.354.3.4 virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (vtkMedicalImageProperties * *pd*) [virtual]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

27.354.3.5 void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (int *n*)

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

27.354.3.6 virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (vtkRTStructSetProperties * *pd*) [virtual]

Examples:

GenerateRTSTRUCT.cxx, and rtstructapp.cxx.

27.354.3.7 vtkGDCMPolyDataWriter::vtkTypeRevisionMacro (vtkGDCMPolyDataWriter , vtkPolyDataWriter)

27.354.3.8 void vtkGDCMPolyDataWriter::WriteData ( ) [protected]

27.354.3.9 void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (gdcm::File & *file,* int *num*) [protected]

27.354.3.10 void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (gdcm::File & *file*) [protected]

27.354.4 Member Data Documentation

27.354.4.1 vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties [protected]

27.354.4.2 vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties [protected]

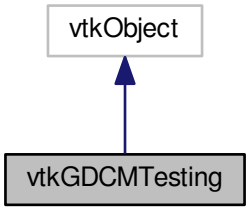
The documentation for this class was generated from the following file:

- vtkGDCMPolyDataWriter.h

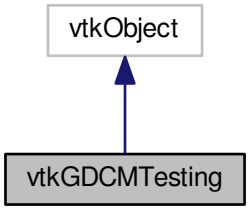
27.355 vtkGDCMTesting Class Reference

`#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)
- vtkTypeRevisionMacro (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 ()

27.355.1 Detailed Description

Examples:

HelloActiviz5.cs, HelloVTKWorld2.cs, MetaImageMD5Activiz.cs, ReadSeriesIntoVTK.java, and RefCounting.cs.

27.355.2 Member Typedef Documentation

27.355.2.1 typedef const char* const(* vtkGDCMTesting::MD5MetaImagesType)[3]

27.355.3 Constructor & Destructor Documentation

27.355.3.1 vtkGDCMTesting::vtkGDCMTesting ( ) [protected]

27.355.3.2 vtkGDCMTesting::~vtkGDCMTesting ( ) [protected]

27.355.4 Member Function Documentation

27.355.4.1 static const char* vtkGDCMTesting::GetGDCMDataRoot ( ) [static]

Examples:

HelloActiviz5.cs, and ReadSeriesIntoVTK.java.

27.355.4.2 static const char* const* vtkGDCMTesting::GetMD5MetaImage (unsigned int *file*) [static]

27.355.4.3 static const char* vtkGDCMTesting::GetMHDMD5FromFile (const char * *filepath*) [static]

Examples:

MetaImageMD5Activiz.cs.

27.355.4.4 static unsigned int vtkGDCMTesting::GetNumberOfMD5MetaImages ( ) [static]

27.355.4.5 static const char* vtkGDCMTesting::GetRAWMD5FromFile (const char * *filepath*) [static]

Examples:

MetaImageMD5Activiz.cs.

27.355.4.6 static const char* vtkGDCMTesting::GetVTKDataRoot ( ) [static]

Examples:

HelloActiviz5.cs, and HelloVTKWorld2.cs.

27.355.4.7 static vtkGDCMTesting* vtkGDCMTesting::New ( ) [static]

Examples:

RefCounting.cs.

27.355.4.8 void vtkGDCMTesting::PrintSelf (ostream & *os,* vtkIndent *indent*)

27.355.4.9 vtkGDCMTesting::vtkTypeRevisionMacro (vtkGDCMTesting , vtkObject)

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

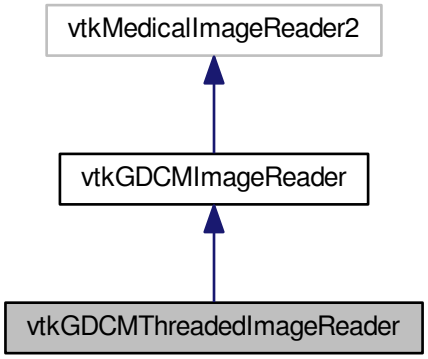
- vtkGDCMTesting.h

27.356 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 (Shift, double)
- vtkSetMacro (Scale, double)
- vtkSetMacro (UseShiftScale, int)
- vtkTypeRevisionMacro (vtkGDCMThreadedImageReader, vtkGDCMImageReader)

Static Public Member Functions

- static vtkGDCMThreadedImageReader * New ()

Protected Member Functions

- vtkGDCMThreadedImageReader ()
- ~vtkGDCMThreadedImageReader ()
- void ExecuteData (vtkDataObject *out)
- void ExecuteInformation ()
- void ReadFiles (unsigned int nfiles, const char *filenames[])
- void RequestDataCompat ()

Additional Inherited Members

27.356.1 Constructor & Destructor Documentation

27.356.1.1 vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader ( ) [protected]

27.356.1.2 vtkGDCMThreadedImageReader::~vtkGDCMThreadedImageReader ( ) [protected]

27.356.2 Member Function Documentation

27.356.2.1 void vtkGDCMThreadedImageReader::ExecuteData (vtkDataObject * *out*) [protected]

27.356.2.2 void vtkGDCMThreadedImageReader::ExecuteInformation ( ) [protected]

27.356.2.3 static vtkGDCMThreadedImageReader* vtkGDCMThreadedImageReader::New ( ) [static]

27.356.2.4 virtual void vtkGDCMThreadedImageReader::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

Reimplemented from vtkGDCMImageReader.

27.356.2.5 void vtkGDCMThreadedImageReader::ReadFiles (unsigned int *nfiles,* const char * *filenames[]*) [protected]

27.356.2.6 void vtkGDCMThreadedImageReader::RequestDataCompat ( ) [protected]

27.356.2.7 vtkGDCMThreadedImageReader::vtkBooleanMacro (UseShiftScale , int)

27.356.2.8 vtkGDCMThreadedImageReader::vtkGetMacro (UseShiftScale , int)

27.356.2.9 vtkGDCMThreadedImageReader::vtkSetMacro (Shift , double)

27.356.2.10 vtkGDCMThreadedImageReader::vtkSetMacro (Scale , double)

27.356.2.11 vtkGDCMThreadedImageReader::vtkSetMacro (UseShiftScale , int)

27.356.2.12 vtkGDCMThreadedImageReader::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader , vtkGDCMImageReader)

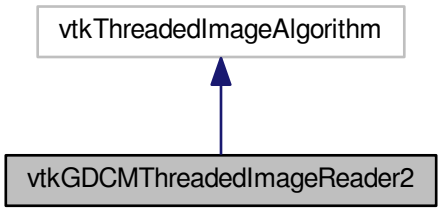
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader.h

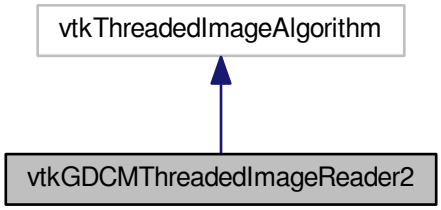
27.357 vtkGDCMThreadedImageReader2 Class Reference

```
#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 (FileLowerLeft, int)
- vtkGetMacro (NumberOfOverlays, int)
- vtkGetMacro (DataScalarType, int)
- vtkGetMacro (NumberOfScalarComponents, int)
- vtkGetMacro (LoadOverlays, int)
- vtkGetMacro (Shift, double)
- vtkGetMacro (Scale, double)
- vtkGetMacro (UseShiftScale, int)

- vtkGetObjectMacro (FileNames, vtkStringArray)
- vtkGetVector3Macro (DataOrigin, double)
- vtkGetVector3Macro (DataSpacing, double)
- vtkGetVector6Macro (DataExtent, int)
- vtkSetMacro (FileLowerLeft, int)
- vtkSetMacro (DataScalarType, int)
- vtkSetMacro (NumberOfScalarComponents, int)
- vtkSetMacro (LoadOverlays, int)
- vtkSetMacro (Shift, double)
- vtkSetMacro (Scale, double)
- vtkSetMacro (UseShiftScale, int)
- vtkSetVector3Macro (DataOrigin, double)
- vtkSetVector3Macro (DataSpacing, double)
- vtkSetVector6Macro (DataExtent, int)
- vtkTypeRevisionMacro (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)

27.357.1 Constructor & Destructor Documentation

27.357.1.1 vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2 ( ) [protected]

27.357.1.2 vtkGDCMThreadedImageReader2::~vtkGDCMThreadedImageReader2 ( ) [protected]

27.357.2 Member Function Documentation

27.357.2.1 virtual const char* vtkGDCMThreadedImageReader2::GetFileName (int *i* = 0) [virtual]

27.357.2.2 static vtkGDCMThreadedImageReader2* vtkGDCMThreadedImageReader2::New ( ) [static]

27.357.2.3 virtual void vtkGDCMThreadedImageReader2::PrintSelf (ostream & *os,* vtkIndent *indent*) [virtual]

27.357.2.4 int vtkGDCMThreadedImageReader2::RequestInformation (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected]

27.357.2.5 virtual void vtkGDCMThreadedImageReader2::SetFileName (const char * *filename*) [virtual]

27.357.2.6 virtual void vtkGDCMThreadedImageReader2::SetFileNames (vtkStringArray *) [virtual]

27.357.2.7 int vtkGDCMThreadedImageReader2::SplitExtent (int *splitExt[6],* int *startExt[6],* int *num,* int *total*)

27.357.2.8 void vtkGDCMThreadedImageReader2::ThreadedRequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector,* vtkImageData *** *inData,* vtkImageData ** *outData,* int *outExt[6],* int *id*) [protected]

27.357.2.9 vtkGDCMThreadedImageReader2::vtkBooleanMacro (FileLowerLeft , int)

27.357.2.10 vtkGDCMThreadedImageReader2::vtkBooleanMacro (LoadOverlays , int)

27.357.2.11 vtkGDCMThreadedImageReader2::vtkBooleanMacro (UseShiftScale , int)

27.357.2.12 vtkGDCMThreadedImageReader2::vtkGetMacro (FileLowerLeft , int)

27.357.2.13 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfOverlays , int)

27.357.2.14 vtkGDCMThreadedImageReader2::vtkGetMacro (DataScalarType , int)

27.357.2.15 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfScalarComponents , int)

27.357.2.16 vtkGDCMThreadedImageReader2::vtkGetMacro (LoadOverlays , int)

27.357.2.17 vtkGDCMThreadedImageReader2::vtkGetMacro (Shift , double)

27.357.2.18 vtkGDCMThreadedImageReader2::vtkGetMacro (Scale , double)

27.357.2.19 vtkGDCMThreadedImageReader2::vtkGetMacro (UseShiftScale , int)

27.357.2.20 vtkGDCMThreadedImageReader2::vtkGetObjectMacro (FileNames , vtkStringArray)

27.357.2.21 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataOrigin , double)

27.357.2.22 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataSpacing , double)

27.357.2.23 vtkGDCMThreadedImageReader2::vtkGetVector6Macro (DataExtent , int)

27.357.2.24 vtkGDCMThreadedImageReader2::vtkSetMacro (FileLowerLeft , int)

27.357.2.25 vtkGDCMThreadedImageReader2::vtkSetMacro (DataScalarType , int)

27.357.2.26 vtkGDCMThreadedImageReader2::vtkSetMacro (NumberOfScalarComponents , int)

27.357.2.27 vtkGDCMThreadedImageReader2::vtkSetMacro (LoadOverlays , int)

27.357.2.28 vtkGDCMThreadedImageReader2::vtkSetMacro (Shift , double)

27.357.2.29 vtkGDCMThreadedImageReader2::vtkSetMacro (Scale , double)

27.357.2.30 vtkGDCMThreadedImageReader2::vtkSetMacro (UseShiftScale , int)

27.357.2.31 vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataOrigin , double)

27.357.2.32 vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataSpacing , double)

27.357.2.33 vtkGDCMThreadedImageReader2::vtkSetVector6Macro (DataExtent , int)

27.357.2.34 vtkGDCMThreadedImageReader2::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader2 , vtkThreadedImageAlgorithm)

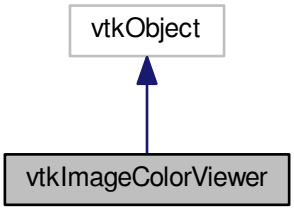
The documentation for this class was generated from the following file:

- vtkGDCMThreadedImageReader2.h

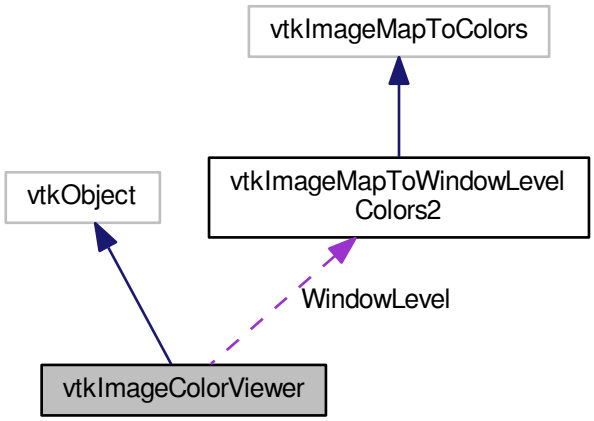
27.358 vtkImageColorViewer Class Reference

`#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 void GetSliceRange (int range[2])
- virtual void GetSliceRange (int &min, int &max)
- virtual int * GetSliceRange ()
- 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 ()
- VTK_LEGACY (int GetWholeZMin())
- VTK_LEGACY (int GetWholeZMax())
- VTK_LEGACY (int GetZSlice())
- VTK_LEGACY (void SetZSlice(int))
- vtkBooleanMacro (OffScreenRendering, int)
- vtkGetMacro (SliceOrientation, int)
- vtkGetMacro (Slice, int)
- vtkGetObjectMacro (RenderWindow, vtkRenderWindow)
- vtkGetObjectMacro (Renderer, vtkRenderer)
- vtkGetObjectMacro (ImageActor, vtkImageActor)
- vtkGetObjectMacro (WindowLevel, vtkImageMapToWindowLevelColors2)
- vtkGetObjectMacro (InteractorStyle, vtkInteractorStyleImage)
- vtkTypeRevisionMacro (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

27.358.1 Detailed Description

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

27.358.2 Member Enumeration Documentation

27.358.2.1 anonymous enum

Enumerator

SLICE_ORIENTATION_YZ

SLICE_ORIENTATION_XZ

SLICE_ORIENTATION_XY

27.358.3 Constructor & Destructor Documentation

27.358.3.1 vtkImageColorViewer::vtkImageColorViewer ( ) [protected]

27.358.3.2 vtkImageColorViewer::~vtkImageColorViewer ( ) [protected]

27.358.4 Member Function Documentation

27.358.4.1 virtual void vtkImageColorViewer::AddInput (vtkImageData * *input*) [virtual]

27.358.4.2 virtual void vtkImageColorViewer::AddInputConnection (vtkAlgorithmOutput * *input*) [virtual]

27.358.4.3 virtual double vtkImageColorViewer::GetColorLevel ( ) [virtual]

27.358.4.4 virtual double vtkImageColorViewer::GetColorWindow ( ) [virtual]

27.358.4.5 virtual vtkImageData* vtkImageColorViewer::GetInput ( ) [virtual]

27.358.4.6 virtual int vtkImageColorViewer::GetOffScreenRendering ( ) [virtual]

27.358.4.7 double vtkImageColorViewer::GetOverlayVisibility ( )

27.358.4.8 virtual int* vtkImageColorViewer::GetPosition ( ) [virtual]

27.358.4.9 virtual int* vtkImageColorViewer::GetSize ( ) [virtual]

27.358.4.10 virtual int vtkImageColorViewer::GetSliceMax ( ) [virtual]

27.358.4.11 virtual int vtkImageColorViewer::GetSliceMin ( ) [virtual]

27.358.4.12 virtual void vtkImageColorViewer::GetSliceRange (int *range[2]*) [inline], [virtual]

27.358.4.13 virtual void vtkImageColorViewer::GetSliceRange (int & *min,* int & *max*) [virtual]

27.358.4.14 virtual int* vtkImageColorViewer::GetSliceRange ( ) [virtual]

27.358.4.15 virtual const char* vtkImageColorViewer::GetWindowName ( ) [virtual]

27.358.4.16 virtual void vtkImageColorViewer::InstallPipeline ( ) [protected], [virtual]

27.358.4.17 static vtkImageColorViewer* vtkImageColorViewer::New ( ) [static]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

27.358.4.18 void vtkImageColorViewer::PrintSelf (ostream & *os,* vtkIndent *indent*)

27.358.4.19 virtual void vtkImageColorViewer::Render (void) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

27.358.4.20 virtual void vtkImageColorViewer::SetColorLevel (double *s*) [virtual]

27.358.4.21 virtual void vtkImageColorViewer::SetColorWindow (double *s*) [virtual]

27.358.4.22 virtual void vtkImageColorViewer::SetDisplayId (void * *a*) [virtual]

27.358.4.23 virtual void vtkImageColorViewer::SetInput (vtkImageData * *in*) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

27.358.4.24 virtual void vtkImageColorViewer::SetInputConnection (vtkAlgorithmOutput * *input*) [virtual]

27.358.4.25 virtual void vtkImageColorViewer::SetOffScreenRendering (int) [virtual]

27.358.4.26 void vtkImageColorViewer::SetOverlayVisibility (double *vis*)

27.358.4.27 virtual void vtkImageColorViewer::SetParentId (void * *a*) [virtual]

27.358.4.28 virtual void vtkImageColorViewer::SetPosition (int *a,* int *b*) [virtual]

27.358.4.29 virtual void vtkImageColorViewer::SetPosition (int *a[2]*) [inline], [virtual]

References SetPosition().

Referenced by SetPosition().

27.358.4.30 virtual void vtkImageColorViewer::SetRenderer (vtkRenderer * *arg*) [virtual]

27.358.4.31 virtual void vtkImageColorViewer::SetRenderWindow (vtkRenderWindow * *arg*) [virtual]

27.358.4.32 virtual void vtkImageColorViewer::SetSize (int *a,* int *b*) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

27.358.4.33 virtual void vtkImageColorViewer::SetSize (int *a[2]*) [inline], [virtual]

References SetSize().

Referenced by SetSize().

27.358.4.34 virtual void vtkImageColorViewer::SetSlice (int *s*) [virtual]

27.358.4.35 virtual void vtkImageColorViewer::SetSliceOrientation (int *orientation*) [virtual]

27.358.4.36 virtual void vtkImageColorViewer::SetSliceOrientationToXY ( ) [inline], [virtual]

References SLICE_ORIENTATION_XY.

27.358.4.37 virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( ) [inline], [virtual]

References SLICE_ORIENTATION_XZ.

27.358.4.38 virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( ) [inline], [virtual]

References SLICE_ORIENTATION_YZ.

27.358.4.39 virtual void vtkImageColorViewer::SetupInteractor (vtkRenderWindowInteractor *) [virtual]

Examples:

gdcmrtionplan.cxx, and gdcmrtplan.cxx.

27.358.4.40 virtual void vtkImageColorViewer::SetWindowId (void * *a*) [virtual]

27.358.4.41 virtual void vtkImageColorViewer::UnInstallPipeline ( ) [protected], [virtual]

27.358.4.42 virtual void vtkImageColorViewer::UpdateDisplayExtent ( ) [virtual]

27.358.4.43 virtual void vtkImageColorViewer::UpdateOrientation ( ) [protected], [virtual]

27.358.4.44 vtkImageColorViewer::VTK_LEGACY (int *GetWholeZMin()*)

27.358.4.45 vtkImageColorViewer::VTK_LEGACY (int *GetWholeZMax()*)

27.358.4.46 vtkImageColorViewer::VTK_LEGACY (int *GetZSlice()*)

27.358.4.47 vtkImageColorViewer::VTK_LEGACY (void *SetZSliceint*)

27.358.4.48 vtkImageColorViewer::vtkBooleanMacro (OffScreenRendering , int)

27.358.4.49 vtkImageColorViewer::vtkGetMacro (**SliceOrientation** , int)

27.358.4.50 vtkImageColorViewer::vtkGetMacro (**Slice** , int)

27.358.4.51 vtkImageColorViewer::vtkGetObjectMacro (**RenderWindow** , vtkRenderWindow)

27.358.4.52 vtkImageColorViewer::vtkGetObjectMacro (**Renderer** , vtkRenderer)

27.358.4.53 vtkImageColorViewer::vtkGetObjectMacro (**ImageActor** , vtkImageActor)

27.358.4.54 vtkImageColorViewer::vtkGetObjectMacro (**WindowLevel** , **vtkImageMapToWindowLevelColors2**)

27.358.4.55 vtkImageColorViewer::vtkGetObjectMacro (**InteractorStyle** , vtkInteractorStyleImage)

27.358.4.56 vtkImageColorViewer::vtkTypeRevisionMacro (**vtkImageColorViewer** , vtkObject)

27.358.5 Friends And Related Function Documentation

27.358.5.1 friend class vtkImageColorViewerCallback [friend]

27.358.6 Member Data Documentation

27.358.6.1 int vtkImageColorViewer::FirstRender [protected]

27.358.6.2 vtkImageActor* vtkImageColorViewer::ImageActor [protected]

27.358.6.3 vtkRenderWindowInteractor* vtkImageColorViewer::Interactor [protected]

27.358.6.4 vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle [protected]

27.358.6.5 vtkImageActor* vtkImageColorViewer::OverlayImageActor [protected]

27.358.6.6 vtkRenderer* vtkImageColorViewer::Renderer [protected]

27.358.6.7 vtkRenderWindow* vtkImageColorViewer::RenderWindow [protected]

27.358.6.8 int vtkImageColorViewer::Slice [protected]

27.358.6.9 int vtkImageColorViewer::SliceOrientation [protected]

27.358.6.10 vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel [protected]

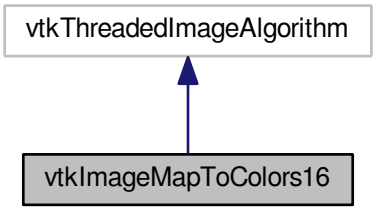
The documentation for this class was generated from the following file:

- vtkImageColorViewer.h

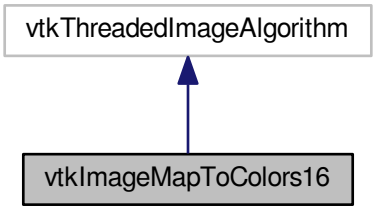
27.359 vtkImageMapToColors16 Class Reference

`#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 (OutputFormat, int)
- vtkGetMacro (ActiveComponent, int)
- vtkGetMacro (PassAlphaToOutput, int)
- vtkGetObjectMacro (LookupTable, vtkScalarsToColors)
- vtkSetMacro (OutputFormat, int)
- vtkSetMacro (ActiveComponent, int)
- vtkSetMacro (PassAlphaToOutput, int)
- vtkTypeRevisionMacro (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

27.359.1 Constructor & Destructor Documentation

27.359.1.1 vtkImageMapToColors16::vtkImageMapToColors16 ( ) [protected]

27.359.1.2 vtkImageMapToColors16::~vtkImageMapToColors16 ( ) [protected]

27.359.2 Member Function Documentation

27.359.2.1 virtual unsigned long vtkImageMapToColors16::GetMTime ( ) [virtual]

27.359.2.2 static vtkImageMapToColors16* vtkImageMapToColors16::New ( ) [static]

27.359.2.3 void vtkImageMapToColors16::PrintSelf (ostream & *os*, vtkIndent *indent*)

27.359.2.4 virtual int vtkImageMapToColors16::RequestData (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*) [protected], [virtual]

27.359.2.5 virtual int vtkImageMapToColors16::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected], [virtual]

27.359.2.6 virtual void vtkImageMapToColors16::SetLookupTable (vtkScalarsToColors *) [virtual]

27.359.2.7 void vtkImageMapToColors16::SetOutputFormatToLuminance ( ) [inline]

27.359.2.8 void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( ) [inline]

27.359.2.9 void vtkImageMapToColors16::SetOutputFormatToRGB ( ) [inline]

27.359.2.10 void vtkImageMapToColors16::SetOutputFormatToRGBA ( ) [inline]

27.359.2.11 void vtkImageMapToColors16::ThreadedRequestData (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*, vtkImageData *** *inData*, vtkImageData ** *outData*, int *extent[6]*, int *id*) [protected]

27.359.2.12 vtkImageMapToColors16::vtkBooleanMacro (PassAlphaToOutput , int)

27.359.2.13 vtkImageMapToColors16::vtkGetMacro (OutputFormat , int)

27.359.2.14 vtkImageMapToColors16::vtkGetMacro (ActiveComponent , int)

27.359.2.15 vtkImageMapToColors16::vtkGetMacro (PassAlphaToOutput , int)

27.359.2.16 vtkImageMapToColors16::vtkGetObjectMacro (LookupTable , vtkScalarsToColors)

27.359.2.17 vtkImageMapToColors16::vtkSetMacro (OutputFormat , int)

27.359.2.18 vtkImageMapToColors16::vtkSetMacro (ActiveComponent , int)

27.359.2.19 vtkImageMapToColors16::vtkSetMacro (PassAlphaToOutput , int)

27.359.2.20 vtkImageMapToColors16::vtkTypeRevisionMacro (vtkImageMapToColors16 , vtkThreadedImageAlgorithm)

27.359.3 Member Data Documentation

27.359.3.1 int vtkImageMapToColors16::ActiveComponent [protected]

27.359.3.2 int vtkImageMapToColors16::DataWasPassed [protected]

27.359.3.3 vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]

27.359.3.4 int vtkImageMapToColors16::OutputFormat [protected]

27.359.3.5 int vtkImageMapToColors16::PassAlphaToOutput [protected]

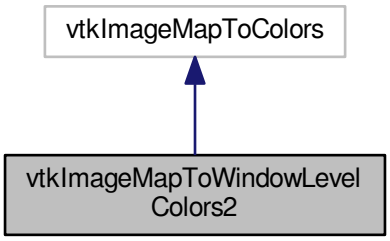
The documentation for this class was generated from the following file:

- vtkImageMapToColors16.h

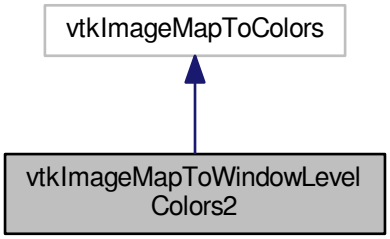
27.360 vtkImageMapToWindowLevelColors2 Class Reference

`#include <vtkImageMapToWindowLevelColors2.h>`

Inheritance diagram for vtkImageMapToWindowLevelColors2:



Collaboration diagram for vtkImageMapToWindowLevelColors2:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkGetMacro (Window, double)
- vtkGetMacro (Level, double)
- vtkSetMacro (Window, double)
- vtkSetMacro (Level, double)
- vtkTypeRevisionMacro (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

27.360.1 Constructor & Destructor Documentation

27.360.1.1 vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( ) [protected]

27.360.1.2 vtkImageMapToWindowLevelColors2::~vtkImageMapToWindowLevelColors2 ( ) [protected]

27.360.2 Member Function Documentation

27.360.2.1 static vtkImageMapToWindowLevelColors2* vtkImageMapToWindowLevelColors2::New ( ) [static]

27.360.2.2 void vtkImageMapToWindowLevelColors2::PrintSelf (ostream & *os,* vtkIndent *indent*)

27.360.2.3 virtual int vtkImageMapToWindowLevelColors2::RequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector*) [protected], [virtual]

27.360.2.4 virtual int vtkImageMapToWindowLevelColors2::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected], [virtual]

27.360.2.5 void vtkImageMapToWindowLevelColors2::ThreadedRequestData (vtkInformation * *request,* vtkInformationVector ** *inputVector,* vtkInformationVector * *outputVector,* vtkImageData *** *inData,* vtkImageData ** *outData,* int *extent[6],* int *id*) [protected]

27.360.2.6 vtkImageMapToWindowLevelColors2::vtkGetMacro (Window , double)

27.360.2.7 vtkImageMapToWindowLevelColors2::vtkGetMacro (Level , double)

27.360.2.8 vtkImageMapToWindowLevelColors2::vtkSetMacro (Window , double)

27.360.2.9 vtkImageMapToWindowLevelColors2::vtkSetMacro (Level , double)

27.360.2.10 vtkImageMapToWindowLevelColors2::vtkTypeRevisionMacro (vtkImageMapToWindowLevelColors2 , vtkImageMapToColors)

27.360.3 Member Data Documentation

27.360.3.1 double vtkImageMapToWindowLevelColors2::Level [protected]

27.360.3.2 double vtkImageMapToWindowLevelColors2::Window [protected]

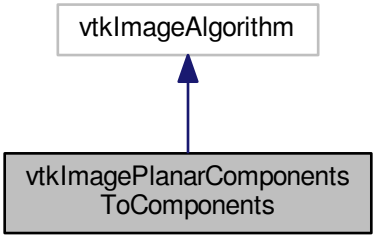
The documentation for this class was generated from the following file:

- vtkImageMapToWindowLevelColors2.h

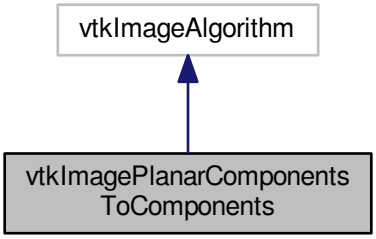
27.361 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (vtkImagePlanarComponentsToComponents, vtkImageAlgorithm)

Static Public Member Functions

- static vtkImagePlanarComponentsToComponents * New ()

Protected Member Functions

- vtkImagePlanarComponentsToComponents ()
- ~vtkImagePlanarComponentsToComponents ()
- virtual int RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)

27.361.1 Constructor & Destructor Documentation

27.361.1.1 vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ( ) [protected]

27.361.1.2 vtkImagePlanarComponentsToComponents::~vtkImagePlanarComponentsToComponents ( ) [inline], [protected]

27.361.2 Member Function Documentation

27.361.2.1 static vtkImagePlanarComponentsToComponents* vtkImagePlanarComponentsToComponents::New ( ) [static]

27.361.2.2 void vtkImagePlanarComponentsToComponents::PrintSelf (ostream & *os*, vtkIndent *indent*)

27.361.2.3 virtual int vtkImagePlanarComponentsToComponents::RequestData (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected], [virtual]

27.361.2.4 vtkImagePlanarComponentsToComponents::vtkTypeRevisionMacro (vtkImagePlanarComponentsToComponents , vtkImageAlgorithm)

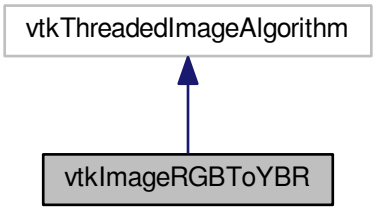
The documentation for this class was generated from the following file:

- vtkImagePlanarComponentsToComponents.h

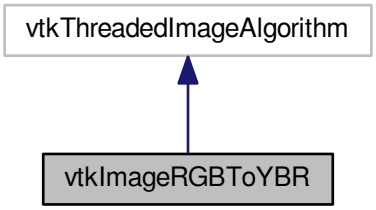
27.362 vtkImageRGBToYBR Class Reference

`#include <vtkImageRGBToYBR.h>`

Inheritance diagram for vtkImageRGBToYBR:



Collaboration diagram for vtkImageRGBToYBR:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (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)

27.362.1 Constructor & Destructor Documentation

27.362.1.1 vtkImageRGBToYBR::vtkImageRGBToYBR ( ) [protected]

27.362.1.2 vtkImageRGBToYBR::~vtkImageRGBToYBR ( ) [inline], [protected]

27.362.2 Member Function Documentation

27.362.2.1 static vtkImageRGBToYBR* vtkImageRGBToYBR::New ( ) [static]

27.362.2.2 void vtkImageRGBToYBR::PrintSelf (ostream & *os*, vtkIndent *indent*)

27.362.2.3 void vtkImageRGBToYBR::ThreadedExecute (vtkImageData * *inData*, vtkImageData * *outData*, int *ext[6]*, int *id*) [protected]

27.362.2.4 vtkImageRGBToYBR::vtkTypeRevisionMacro (vtkImageRGBToYBR , vtkThreadedImageAlgorithm)

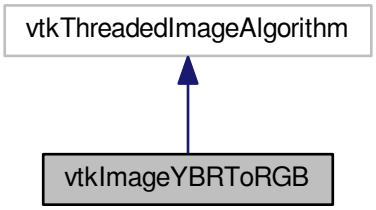
The documentation for this class was generated from the following file:

- vtkImageRGBToYBR.h

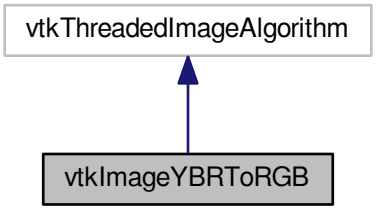
27.363 vtkImageYBRToRGB Class Reference

`#include <vtkImageYBRToRGB.h>`

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



Public Member Functions

- void PrintSelf (ostream &os, vtkIndent indent)
- vtkTypeRevisionMacro (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)

27.363.1 Constructor & Destructor Documentation

27.363.1.1 vtkImageYBRToRGB::vtkImageYBRToRGB ( ) [protected]

27.363.1.2 vtkImageYBRToRGB::~vtkImageYBRToRGB ( ) [inline], [protected]

27.363.2 Member Function Documentation

27.363.2.1 static vtkImageYBRToRGB* vtkImageYBRToRGB::New ( ) [static]

27.363.2.2 void vtkImageYBRToRGB::PrintSelf (ostream & *os*, vtkIndent *indent*)

27.363.2.3 void vtkImageYBRToRGB::ThreadedExecute (vtkImageData * *inData*, vtkImageData * *outData*, int *ext[6]*, int *id*) [protected]

27.363.2.4 vtkImageYBRToRGB::vtkTypeRevisionMacro (vtkImageYBRToRGB , vtkThreadedImageAlgorithm)

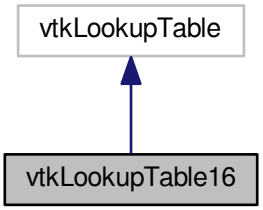
The documentation for this class was generated from the following file:

- vtkImageYBRToRGB.h

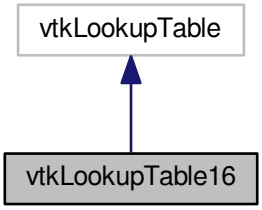
27.364 vtkLookupTable16 Class Reference

`#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)
- vtkTypeRevisionMacro (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

27.364.1 Constructor & Destructor Documentation

27.364.1.1 vtkLookupTable16::vtkLookupTable16 (int *sze* = 256, int *ext* = 256) [protected]

27.364.1.2 vtkLookupTable16::~vtkLookupTable16 ( ) [protected]

27.364.2 Member Function Documentation

27.364.2.1 void vtkLookupTable16::Build ( )

27.364.2.2 unsigned short* vtkLookupTable16::GetPointer (const vtkIdType *id*) [inline]

27.364.2.3 void vtkLookupTable16::MapScalarsThroughTable2 (void * *input*, unsigned char * *output*, int *inputDataType*, int *numberOfValues*, int *inputIncrement*, int *outputFormat*) [protected]

27.364.2.4 static vtkLookupTable16* vtkLookupTable16::New ( ) [static]

27.364.2.5 void vtkLookupTable16::PrintSelf (ostream & *os*, vtkIndent *indent*)

27.364.2.6 void vtkLookupTable16::SetNumberOfTableValues (vtkIdType *number*)

27.364.2.7 vtkLookupTable16::vtkTypeRevisionMacro (vtkLookupTable16 , vtkLookupTable)

27.364.2.8 unsigned char * vtkLookupTable16::WritePointer (const vtkIdType *id*, const int *number*) [inline]

References Table16.

27.364.3 Member Data Documentation

27.364.3.1 vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]

Referenced by WritePointer().

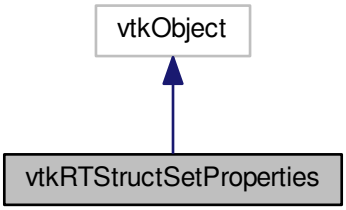
The documentation for this class was generated from the following file:

- vtkLookupTable16.h

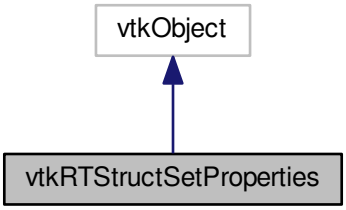
27.365 vtkRTStructSetProperties Class Reference

`#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 *ROI↩ GenerationAlgorithm, 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 (StructureSetLabel)
- vtkGetStringMacro (StructureSetName)
- vtkGetStringMacro (StructureSetDate)
- vtkGetStringMacro (StructureSetTime)
- vtkGetStringMacro (SOPInstanceUID)
- vtkGetStringMacro (StudyInstanceUID)
- vtkGetStringMacro (SeriesInstanceUID)

- vtkGetStringMacro (ReferenceSeriesInstanceUID)
- vtkGetStringMacro (ReferenceFrameOfReferenceUID)
- vtkSetStringMacro (StructureSetLabel)
- vtkSetStringMacro (StructureSetName)
- vtkSetStringMacro (StructureSetDate)
- vtkSetStringMacro (StructureSetTime)
- vtkSetStringMacro (SOPInstanceUID)
- vtkSetStringMacro (StudyInstanceUID)
- vtkSetStringMacro (SeriesInstanceUID)
- vtkSetStringMacro (ReferenceSeriesInstanceUID)
- vtkSetStringMacro (ReferenceFrameOfReferenceUID)
- vtkTypeRevisionMacro (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

27.365.1 Detailed Description

Examples:

GenerateRTSTRUCT.cxx.

27.365.2 Constructor & Destructor Documentation

27.365.2.1 vtkRTStructSetProperties::vtkRTStructSetProperties ( ) [protected]

27.365.2.2 vtkRTStructSetProperties::~vtkRTStructSetProperties ( ) [protected]

27.365.3 Member Function Documentation

27.365.3.1 void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (vtkIdType *pdnum,* const char * *classuid,* const char * *instanceuid*)

27.365.3.2 void vtkRTStructSetProperties::AddReferencedFrameOfReference (const char * *classuid,* const char * *instanceuid*)

27.365.3.3 void vtkRTStructSetProperties::AddStructureSetROI (int *roinumber,* const char * *reframerefuid,* const char * *roiname,* const char * *ROIGenerationAlgorithm,* const char * *ROIDescription* = 0)

27.365.3.4 void vtkRTStructSetProperties::AddStructureSetROIObservation (int *refnumber,* int *observationnumber,* const char * *rtroiinterpretedtype,* const char * *roiinterpreter,* const char * *roiobservationlabel* = 0)

27.365.3.5 virtual void vtkRTStructSetProperties::Clear ( ) [virtual]

27.365.3.6 virtual void vtkRTStructSetProperties::DeepCopy (**vtkRTStructSetProperties** * *p*) [virtual]

27.365.3.7 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (vtkIdType *pdnum,* vtkIdType *id*)

27.365.3.8 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType *pdnum,* vtkIdType *id*)

27.365.3.9 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )

27.365.3.10 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (vtkIdType *pdnum*)

27.365.3.11 vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )

27.365.3.12 vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )

27.365.3.13 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (vtkIdType *id*)

27.365.3.14 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (vtkIdType *id*)

27.365.3.15 int vtkRTStructSetProperties::GetStructureSetObservationNumber (vtkIdType *id*)

27.365.3.16 const char* vtkRTStructSetProperties::GetStructureSetROIDescription (vtkIdType *id*)

27.365.3.17 const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (vtkIdType)

27.365.3.18 const char* vtkRTStructSetProperties::GetStructureSetROIName (vtkIdType)

27.365.3.19 int vtkRTStructSetProperties::GetStructureSetROINumber (vtkIdType *id*)

27.365.3.20 const char* vtkRTStructSetProperties::GetStructureSetROIObservationLabel (vtkIdType *id*)

27.365.3.21 const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (vtkIdType)

27.365.3.22 const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (vtkIdType *id*)

27.365.3.23 static vtkRTStructSetProperties* vtkRTStructSetProperties::New ( ) [static]

Examples:

GenerateRTSTRUCT.cxx.

27.365.3.24 void vtkRTStructSetProperties::PrintSelf (ostream & *os,* vtkIndent *indent*)

27.365.3.25 vtkRTStructSetProperties::vtkGetStringMacro (StructureSetLabel)

27.365.3.26 vtkRTStructSetProperties::vtkGetStringMacro (StructureSetName)

27.365.3.27 vtkRTStructSetProperties::vtkGetStringMacro (StructureSetDate)

27.365.3.28 vtkRTStructSetProperties::vtkGetStringMacro (StructureSetTime)

27.365.3.29 vtkRTStructSetProperties::vtkGetStringMacro (SOPInstanceUID)

27.365.3.30 vtkRTStructSetProperties::vtkGetStringMacro (StudyInstanceUID)

27.365.3.31 vtkRTStructSetProperties::vtkGetStringMacro (SeriesInstanceUID)

27.365.3.32 vtkRTStructSetProperties::vtkGetStringMacro (ReferenceSeriesInstanceUID)

27.365.3.33 vtkRTStructSetProperties::vtkGetStringMacro (ReferenceFrameOfReferenceUID)

27.365.3.34 vtkRTStructSetProperties::vtkSetStringMacro (StructureSetLabel)

27.365.3.35 vtkRTStructSetProperties::vtkSetStringMacro (StructureSetName)

27.365.3.36 vtkRTStructSetProperties::vtkSetStringMacro (StructureSetDate)

27.365.3.37 vtkRTStructSetProperties::vtkSetStringMacro (StructureSetTime)

27.365.3.38 vtkRTStructSetProperties::vtkSetStringMacro (SOPInstanceUID)

27.365.3.39 vtkRTStructSetProperties::vtkSetStringMacro (StudyInstanceUID)

27.365.3.40 vtkRTStructSetProperties::vtkSetStringMacro (SeriesInstanceUID)

27.365.3.41 vtkRTStructSetProperties::vtkSetStringMacro (ReferenceSeriesInstanceUID)

27.365.3.42 vtkRTStructSetProperties::vtkSetStringMacro (ReferenceFrameOfReferenceUID)

27.365.3.43 vtkRTStructSetProperties::vtkTypeRevisionMacro (vtkRTStructSetProperties , vtkObject)

27.365.4 Member Data Documentation

27.365.4.1 vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals [protected]

27.365.4.2 char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID [protected]

27.365.4.3 char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID [protected]

27.365.4.4 char* vtkRTStructSetProperties::SeriesInstanceUID [protected]

27.365.4.5 char* vtkRTStructSetProperties::SOPInstanceUID [protected]

27.365.4.6 char* vtkRTStructSetProperties::StructureSetDate [protected]

27.365.4.7 char* vtkRTStructSetProperties::StructureSetLabel [protected]

27.365.4.8 char* vtkRTStructSetProperties::StructureSetName [protected]

27.365.4.9 char* vtkRTStructSetProperties::StructureSetTime [protected]

27.365.4.10 char* vtkRTStructSetProperties::StudyInstanceUID [protected]

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

- vtkRTStructSetProperties.h

27.366 gdcm::Waveform Class Reference

Waveform class.

`#include <gdcmWaveform.h>`

Public Member Functions

- Waveform ()

27.366.1 Detailed Description

Waveform class.

27.366.2 Constructor & Destructor Documentation

27.366.2.1 gdcm::Waveform::Waveform ( ) [inline]

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

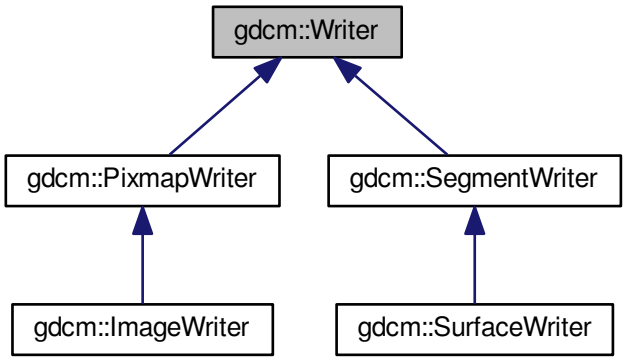
- gdcmWaveform.h

27.367 gdcm::Writer Class Reference

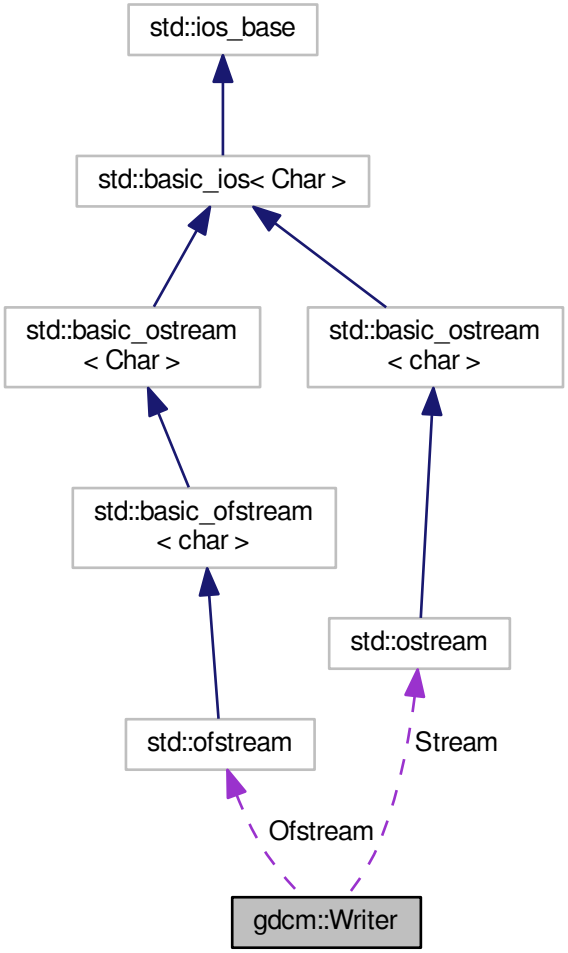
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.

`#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

- std::ostream * GetStreamPtr () const
- void SetWriteDataSetOnly (bool b)

Protected Attributes

- std::ofstream * Ofstream
- std::ostream * Stream

Friends

- class StreamImageWriter

27.367.1 Detailed Description

Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only.

Detailled 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 (garanteed 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, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_Stream_Image_Writer.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

27.367.2 Constructor & Destructor Documentation

27.367.2.1 gdcm::Writer::Writer ( )

27.367.2.2 virtual gdcm::Writer::~Writer ( ) [virtual]

27.367.3 Member Function Documentation

27.367.3.1 void gdcm::Writer::CheckFileMetaInformationOff ( ) [inline]

Examples:

CreateFakeRTDOSE.cxx, FixBrokenJ2K.cxx, and HelloWorld.cxx.

27.367.3.2 void gdcm::Writer::CheckFileMetaInformationOn ( ) [inline]

27.367.3.3 File& gdcm::Writer::GetFile ( ) [inline]

Examples:

CreateJPIPDataSet.cxx, EncapsulateFileInRawData.cxx, Extracting_All_Resolution.cxx, Fake_Image_Using_↩ Stream_Image_Writer.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GetSubSequenceData.cxx, iU22tomultisc.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, rle2img.cxx, and StreamImageReaderTest.cxx.

27.367.3.4 std::ostream* gdcm::Writer::GetStreamPtr ( ) const [inline], [protected]

27.367.3.5 void gdcm::Writer::SetCheckFileMetaInformation (bool *b*) [inline]

Undocumented function, do not use (= leave default)

Examples:

GenAllVR.cxx, GenFakeIdentifyFile.cxx, and PatchFile.cxx.

27.367.3.6 void gdcm::Writer::SetFile (const File & *f*) [inline]

Set/Get the DICOM file (DataSet + Header)

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, CreateFakeRTDOSE.cxx, DuplicatePCDE.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, NewSequence.cs, PatchFile.cxx, pmsct_rgb1.cxx, and rle2img.cxx.

27.367.3.7 void gdcm::Writer::SetFileName (const char * *filename_native*)

Set the filename of DICOM file to write:

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CompressImage.cxx, CreateARGBImage.cxx, CreateCMYKImage.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, csa2img.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenFakeImage.cxx, GenLongSeqs.cxx, GenSeqs.cxx, GetSubSequenceData.cxx, HelloVizWorld.cxx, HelloWorld.cxx, iU22tomultisc.cxx, LargeVRDSExplicit.cxx, MergeTwoFiles.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, and rle2img.cxx.

27.367.3.8 void gdcm::Writer::SetStream (std::ostream & *output_stream*) [inline]

Set user ostream buffer.

27.367.3.9 void gdcm::Writer::SetWriteDataSetOnly (bool *b*) [inline], [protected]

27.367.3.10 virtual bool gdcm::Writer::Write ( ) [virtual]

Main function to tell the writer to write.

Reimplemented in gdcm::PixmapWriter, gdcm::ImageWriter, gdcm::SurfaceWriter, and gdcm::SegmentWriter.

Examples:

ChangePrivateTags.cxx, ChangeSequenceUltrasound.cxx, ClinicalTrialAnnotate.cxx, CreateFakeRTDOSE.cxx, CreateJPIPDataSet.cxx, DuplicatePCDE.cxx, EncapsulateFileInRawData.cxx, FixBrokenJ2K.cxx, FixJAIBugJPEGLS.cxx, GenAllVR.cxx, GenFakeIdentifyFile.cxx, GenLongSeqs.cxx, GenSeqs.cxx, HelloWorld.cxx, LargeVRDSExplicit.cxx, PatchFile.cxx, pmsct_rgb1.cxx, QIDO-RS.cxx, and rle2img.cxx.

27.367.4 Friends And Related Function Documentation

27.367.4.1 friend class StreamImageWriter [friend]

27.367.5 Member Data Documentation

27.367.5.1 std::ofstream* gdcm::Writer::Ofstream [protected]

27.367.5.2 std::ostream* gdcm::Writer::Stream [protected]

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

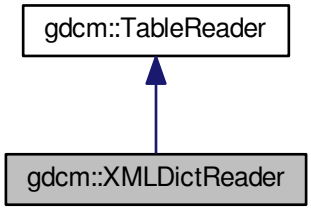
- gdcmWriter.h

27.368 gdcm::XMLDictReader Class Reference

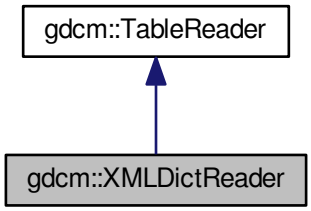
Class for representing a XMLDictReader.

`#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)

Protected Member Functions

- void HandleDescription (const char **atts)
- void HandleEntry (const char **atts)

27.368.1 Detailed Description

Class for representing a XMLDictReader.

Note

bla Will read the DICOMV3.xml file

27.368.2 Constructor & Destructor Documentation

27.368.2.1 gdcm::XMLDictReader::XMLDictReader ( )

27.368.2.2 gdcm::XMLDictReader::~XMLDictReader ( ) [inline]

27.368.3 Member Function Documentation

27.368.3.1 void gdcm::XMLDictReader::CharacterDataHandler (const char * *data*, int *length*) [virtual]

Reimplemented from gdcm::TableReader.

27.368.3.2 void gdcm::XMLDictReader::EndElement (const char * *name*) [virtual]

Reimplemented from gdcm::TableReader.

27.368.3.3 const Dict& gdcm::XMLDictReader::GetDict ( ) [inline]

27.368.3.4 void gdcm::XMLDictReader::HandleDescription (const char ** *atts*) [protected]

27.368.3.5 void gdcm::XMLDictReader::HandleEntry (const char ** *atts*) [protected]

27.368.3.6 void gdcm::XMLDictReader::StartElement (const char * *name*, const char ** *atts*) [virtual]

Reimplemented from gdcm::TableReader.

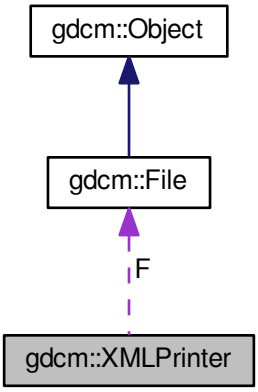
The documentation for this class was generated from the following file:

- gdcmXMLDictReader.h

27.369 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

27.369.1 Member Enumeration Documentation

27.369.1.1 enum gdcm::XMLPrinter::PrintStyles

Enumerator

OnlyUUID

LOADBULKDATA

27.369.2 Constructor & Destructor Documentation

27.369.2.1 gdcm::XMLPrinter::XMLPrinter ( )

27.369.2.2 virtual gdcm::XMLPrinter::~XMLPrinter ( ) [virtual]

27.369.3 Member Function Documentation

27.369.3.1 PrintStyles gdcm::XMLPrinter::GetPrintStyle ( ) const [inline]

27.369.3.2 virtual void gdcm::XMLPrinter::HandleBulkData (const char * *uuid*, const TransferSyntax & *ts*, const char * *bulkdata*, size_t *bulklen*) [virtual]

Virtual function mecanism to allow application programmer to override the default mecanism for BulkData handling. By default GDCM will simply discard the BulkData and only write the UUID

27.369.3.3 void gdcm::XMLPrinter::Print (std::ostream & *os*)

27.369.3.4 VR gdcm::XMLPrinter::PrintDataElement (std::ostream & *os*, const Dicts & *dicts*, const DataSet & *ds*, const DataElement & *de*, const TransferSyntax & *ts*) [protected]

27.369.3.5 void gdcm::XMLPrinter::PrintDataSet (const DataSet & *ds*, const TransferSyntax & *ts*, std::ostream & *os*)

27.369.3.6 void gdcm::XMLPrinter::PrintSQ (const SequenceOfItems * *sqi*, const TransferSyntax & *ts*, std::ostream & *os*) [protected]

27.369.3.7 void gdcm::XMLPrinter::SetFile (File const & *f*) [inline]

27.369.3.8 void gdcm::XMLPrinter::SetStyle (PrintStyles *ps*) [inline]

27.369.4 Member Data Documentation

27.369.4.1 const File* gdcm::XMLPrinter::F [protected]

27.369.4.2 PrintStyles gdcm::XMLPrinter::PrintStyle [protected]

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

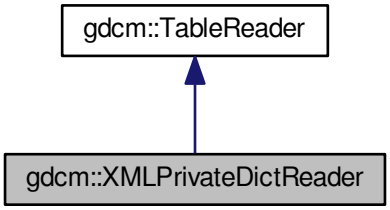
- gdcmXMLPrinter.h

27.370 gdcm::XMLPrivateDictReader Class Reference

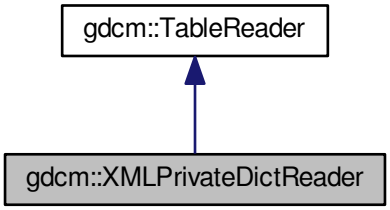
Class for representing a XMLPrivateDictReader.

`#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)

Protected Member Functions

- void HandleDescription (const char **atts)
- void HandleEntry (const char **atts)

27.370.1 Detailed Description

Class for representing a XMLPrivateDictReader.

Note

bla Will read the Private.xml file

27.370.2 Constructor & Destructor Documentation

27.370.2.1 gdcm::XMLPrivateDictReader::XMLPrivateDictReader ( )

27.370.2.2 gdcm::XMLPrivateDictReader::~XMLPrivateDictReader ( ) [inline]

27.370.3 Member Function Documentation

27.370.3.1 void gdcm::XMLPrivateDictReader::CharacterDataHandler (const char * *data*, int *length*) [virtual]

Reimplemented from gdcm::TableReader.

27.370.3.2 void gdcm::XMLPrivateDictReader::EndElement (const char * *name*) [virtual]

Reimplemented from gdcm::TableReader.

27.370.3.3 const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ( ) [inline]

27.370.3.4 void gdcm::XMLPrivateDictReader::HandleDescription (const char ** *atts*) [protected]

27.370.3.5 void gdcm::XMLPrivateDictReader::HandleEntry (const char ** *atts*) [protected]

27.370.3.6 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 28

File Documentation

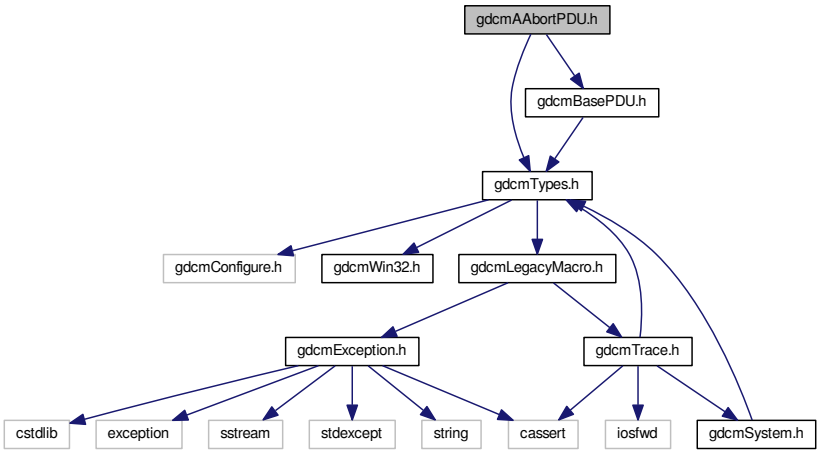
28.1 gdcm2pnm.dox File Reference

28.2 gdcm2vtk.dox File Reference

28.3 gdcmAAbortPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAbortPDU.h:



Classes

- class gdcm::network::AAbortPDU

  AAbortPDU Table 9-26 A-ABORT PDU FIELDS.

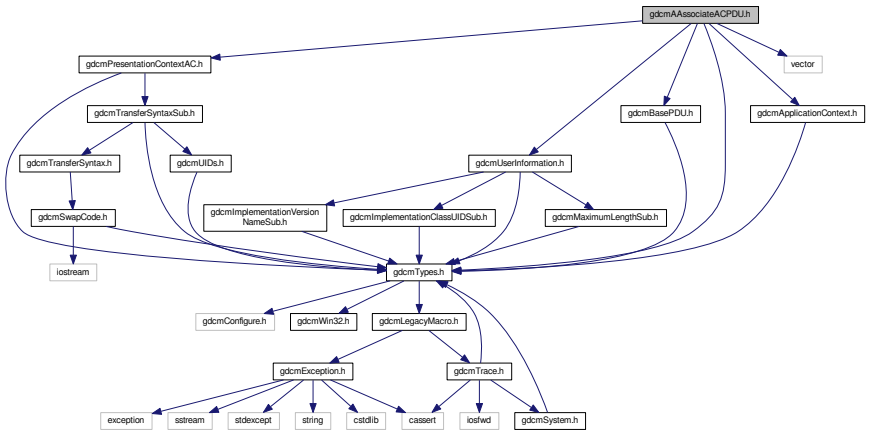
Namespaces

- gdcm
- gdcm::network

28.4 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 Table 9-17 ASSOCIATE-AC PDU fields.

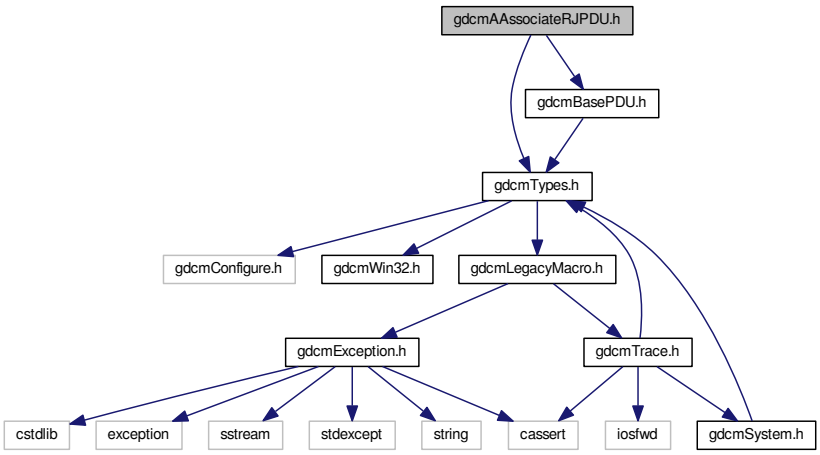
Namespaces

- gdcm
- gdcm::network

28.5 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAssociateRJPDU.h:



Classes

- class gdcm::network::AAssociateRJPDU

  AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.

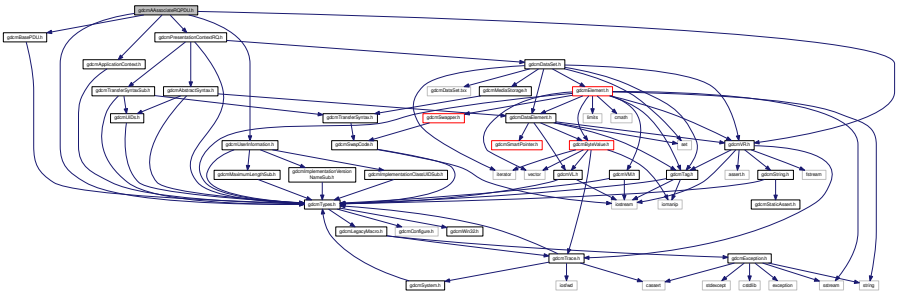
Namespaces

- gdcm
- gdcm::network

28.6 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 Table 9-11 ASSOCIATE-RQ PDU fields.

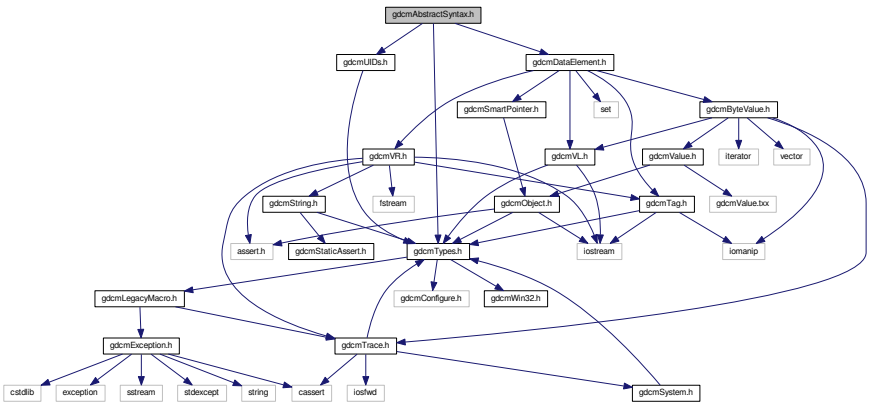
Namespaces

- gdcm
- gdcm::network

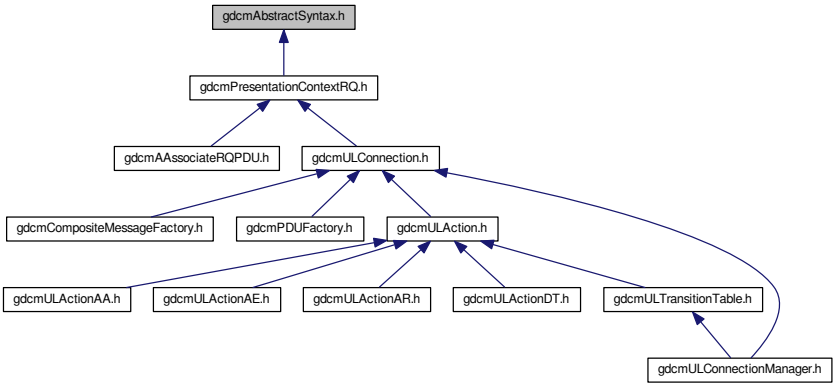
28.7 gdcmAbstractSyntax.h File Reference

```
#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 Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

Namespaces

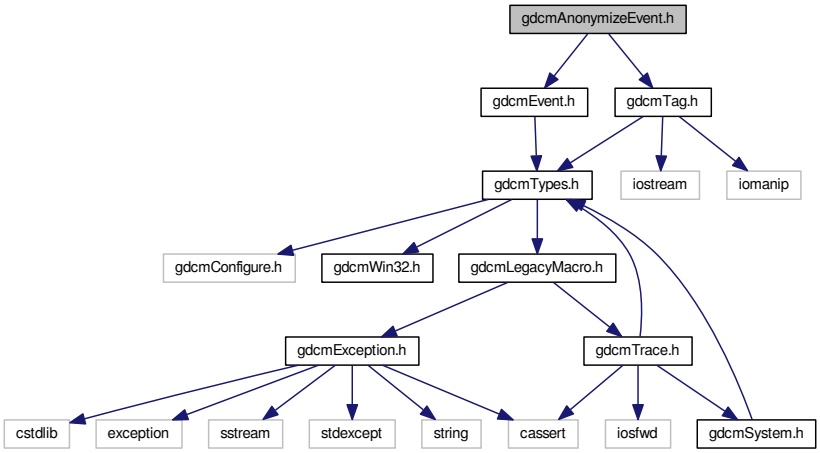
- gdcm
- gdcm::network

28.8 gdcmanon.dox File Reference

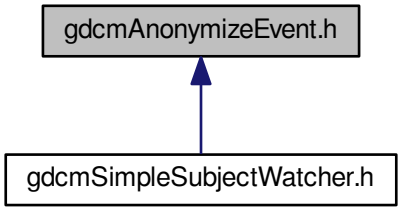
28.9 gdcmAnonymizeEvent.h File Reference

```
#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 Special type of event triggered during the Anonymization process.

Namespaces

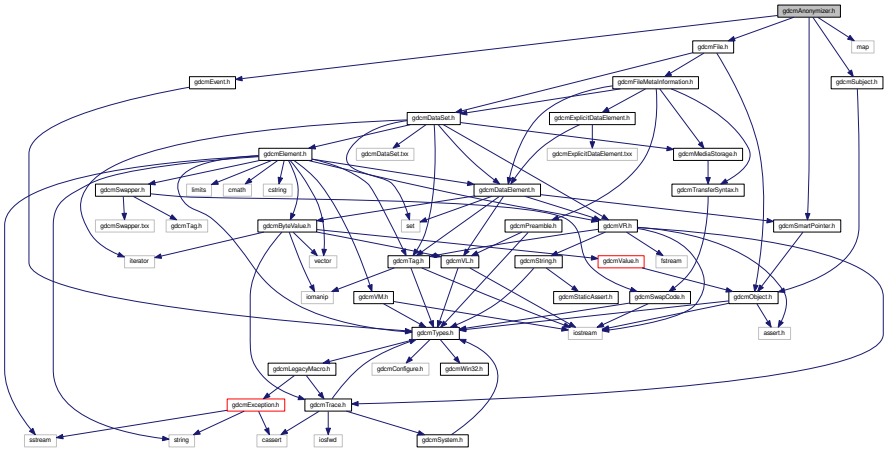
- gdcm

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

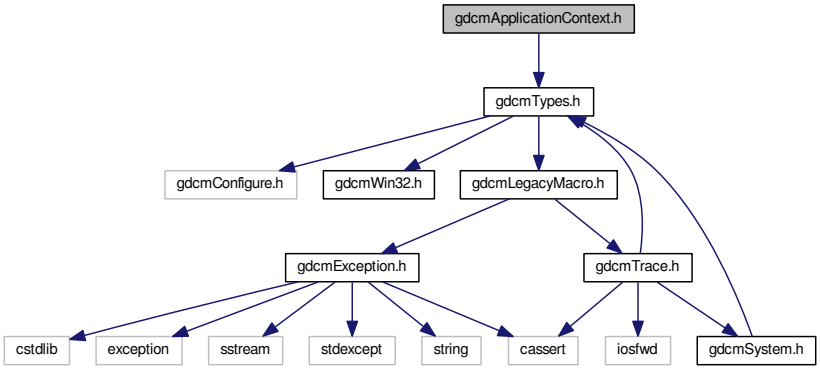
Namespaces

- gdcm

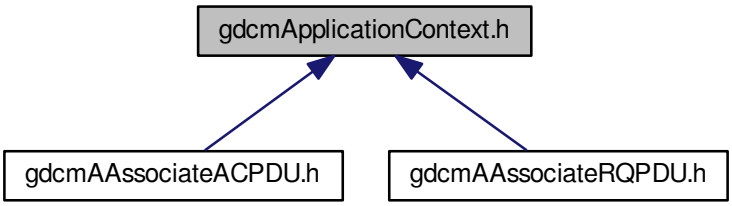
28.11 gdcmApplicationContext.h File Reference

```
#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 Table 9-12 APPLICATION CONTEXT ITEM FIELDS.

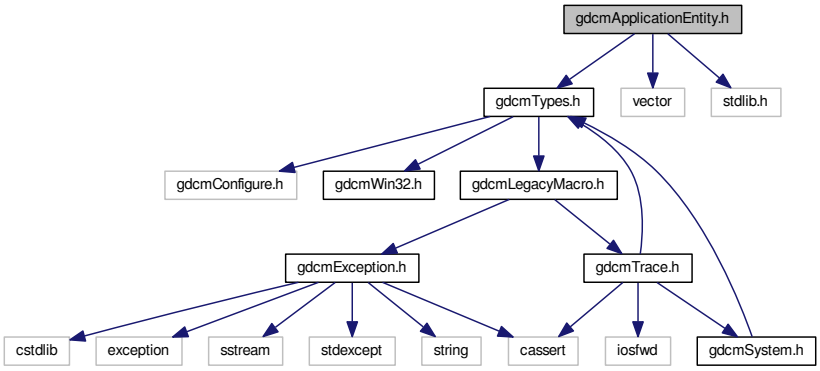
Namespaces

- gdcm
- gdcm::network

28.12 gdcmApplicationEntity.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <stdlib.h>
```

Include dependency graph for gdcmApplicationEntity.h:



Classes

- class gdcm::ApplicationEntity

  ApplicationEntity.

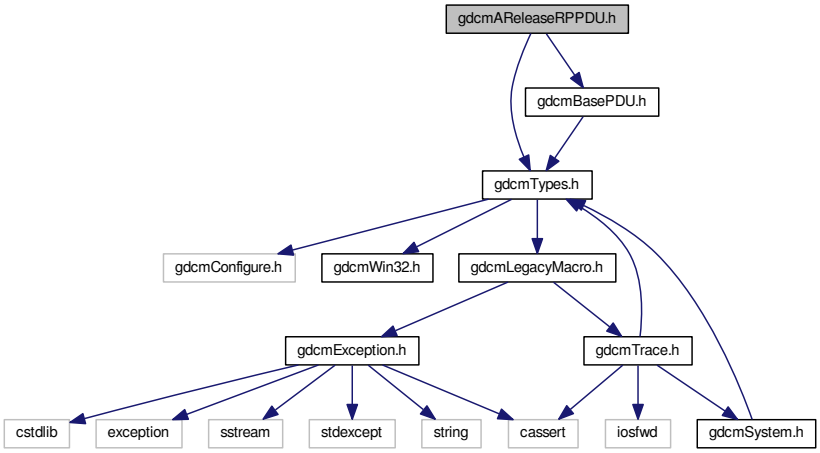
Namespaces

- gdcm

28.13 gdcmAReleaseRPPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRPPDU.h:



Classes

- class gdcm::network::AReleaseRPPDU

  AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.

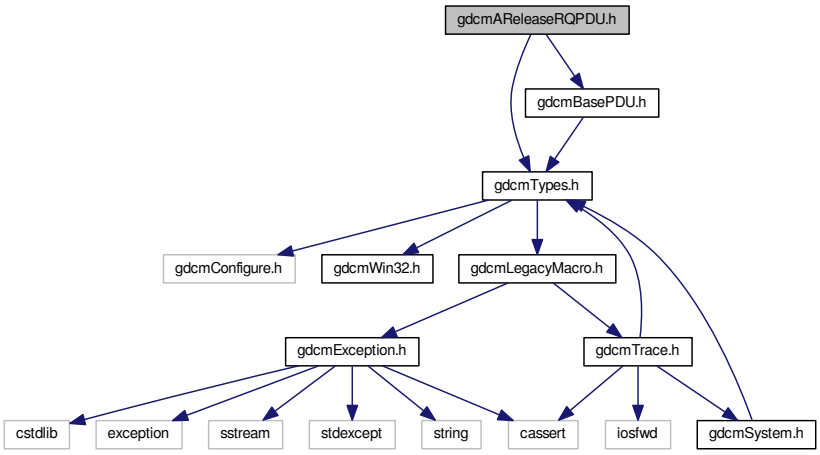
Namespaces

- gdcm
- gdcm::network

28.14 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAReleaseRQPDU.h:



Classes

- class gdcm::network::AReleaseRQPDU

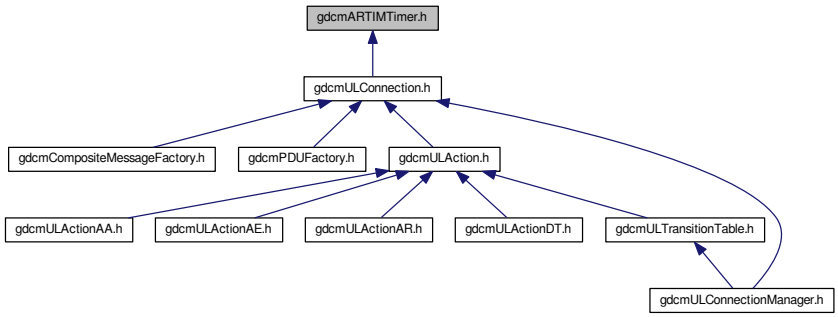
  AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.

Namespaces

- gdcm
- gdcm::network

28.15 gdcmARTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::network::ARTIMTimer

  ARTIMTimer This file contains the code for the ARTIM timer.

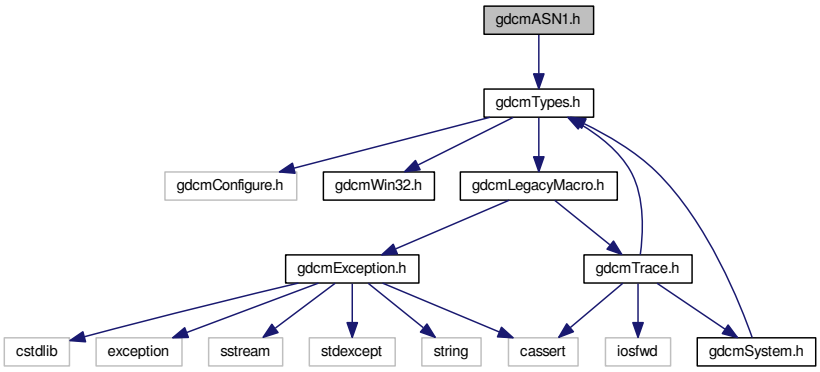
Namespaces

- gdcm
- gdcm::network

28.16 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



Classes

- class gdcm::ASN1

  Class for ASN1.

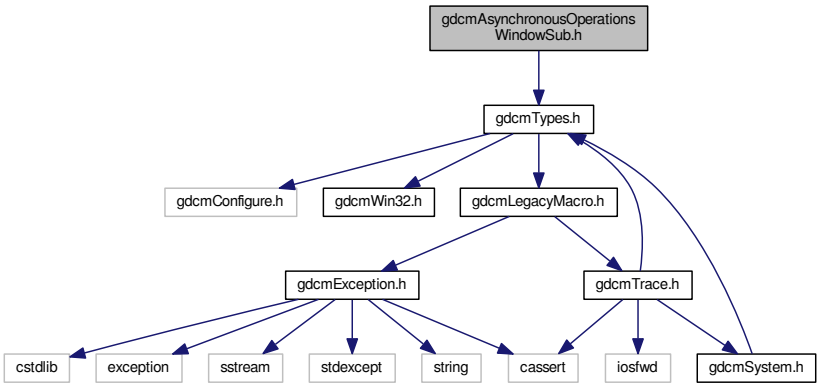
Namespaces

- gdcm

28.17 gdcmAsynchronousOperationsWindowSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



Classes

- class gdcm::network::AsynchronousOperationsWindowSub

  AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIE↩ LDS (A-ASSOCIATE-RQ)

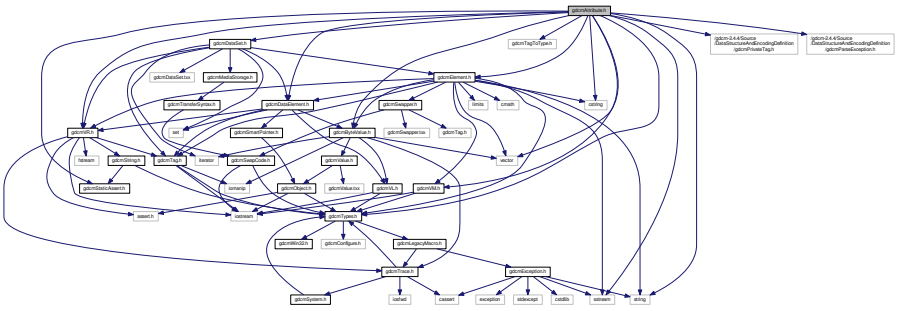
Namespaces

- gdcm
- gdcm::network

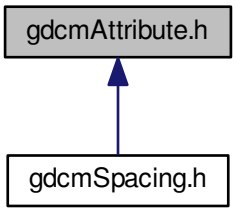
28.18 gdcmAttribute.h File Reference

```
#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 instanciation 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 >
- singleton gdcm::VRVLSize< T >
- class gdcm::VRVLSize< 0 >
- class gdcm::VRVLSize< 1 >

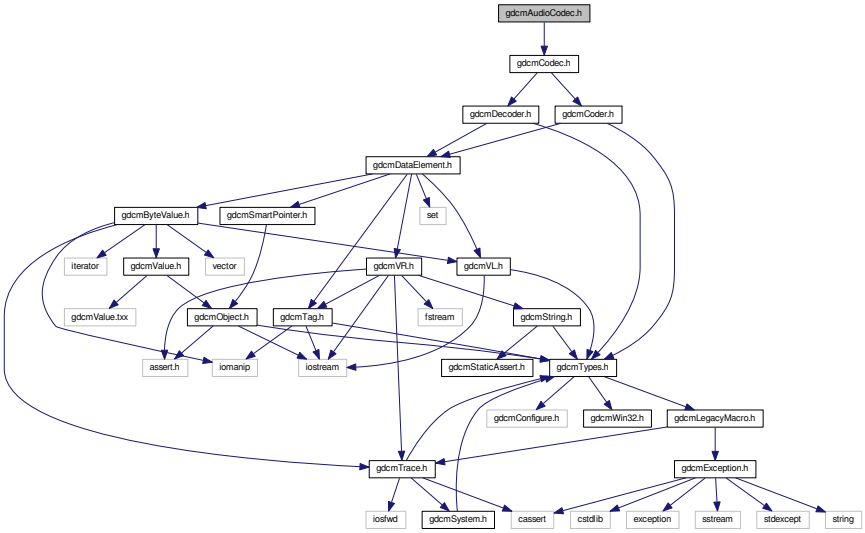
Namespaces

- gdcm

28.19 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



Classes

- class gdcm::AudioCodec

  AudioCodec.

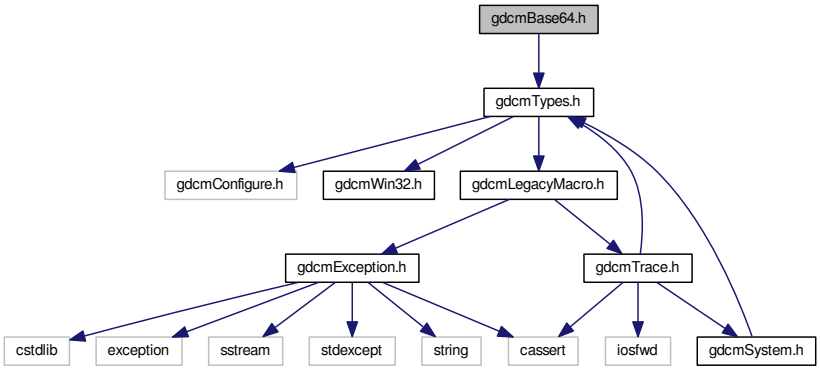
Namespaces

- gdcm

28.20 gdcmBase64.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmBase64.h:



Classes

- class gdcm::Base64

  Class for Base64.

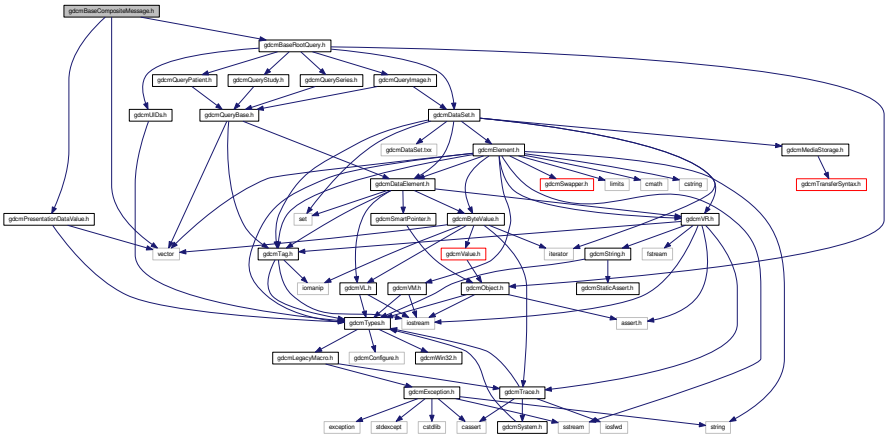
Namespaces

- gdcm

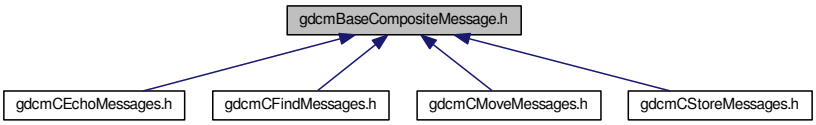
28.21 gdcmBaseCompositeMessage.h File Reference

```
#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 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.

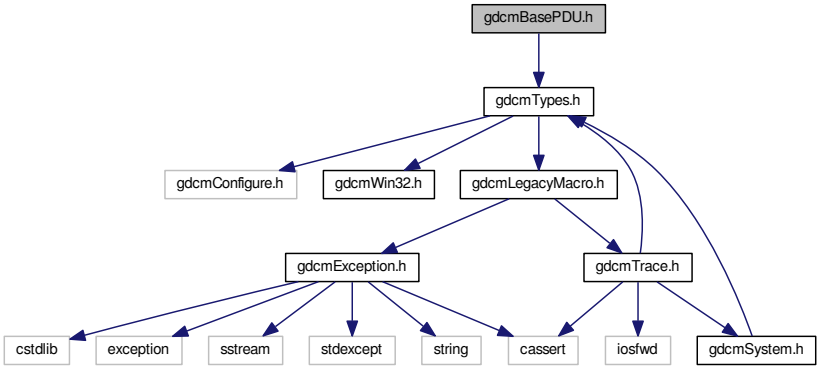
Namespaces

- gdcm
- gdcm::network

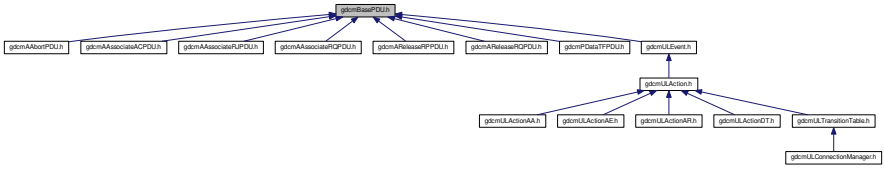
28.22 gdcmBasePDU.h File Reference

```
#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 base class for PDUs.

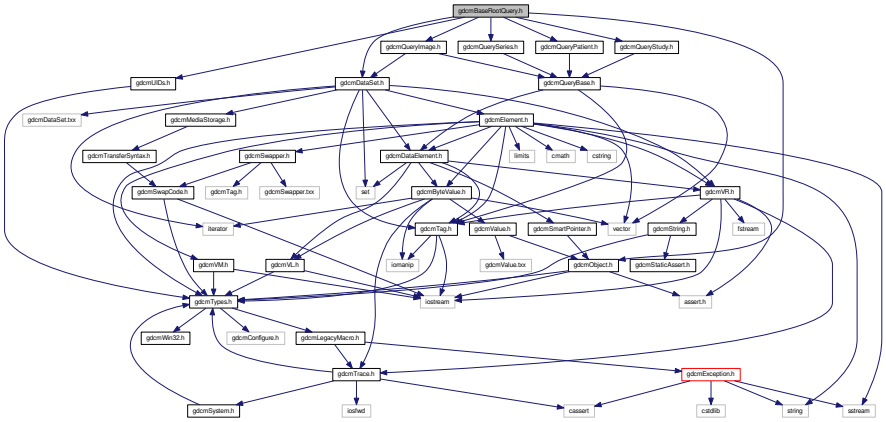
Namespaces

- gdcm
- gdcm::network

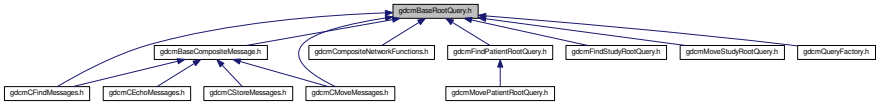
28.23 gdcmBaseRootQuery.h File Reference

```
#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmObject.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 contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.

Namespaces

- gdcm

Enumerations

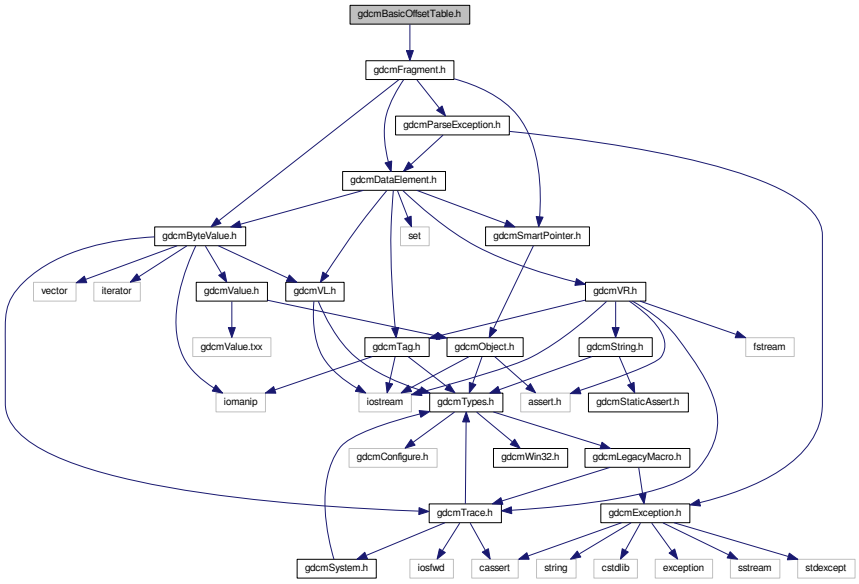
- enum gdcm::EQueryLevel {
  gdcm::ePatient = 0,
  gdcm::eStudy = 1,
  gdcm::eSeries = 2,
  gdcm::eImage = 3 }

- enum gdcm::EQueryType {
  gdcm::eFind = 0,
  gdcm::eMove }

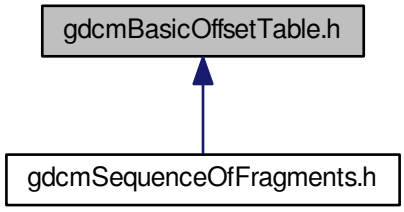
28.24 gdcmBasicOffsetTable.h File Reference

```
#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

- gdcm

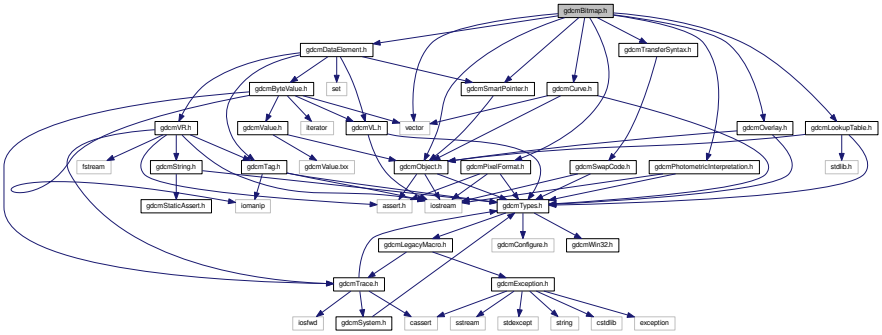
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const BasicOffsetTable &val)

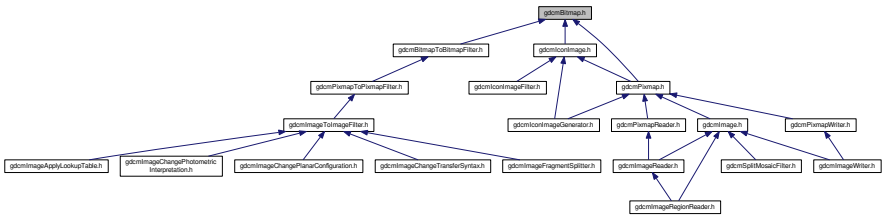
28.25 gdcmBitmap.h File Reference

```
#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 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)

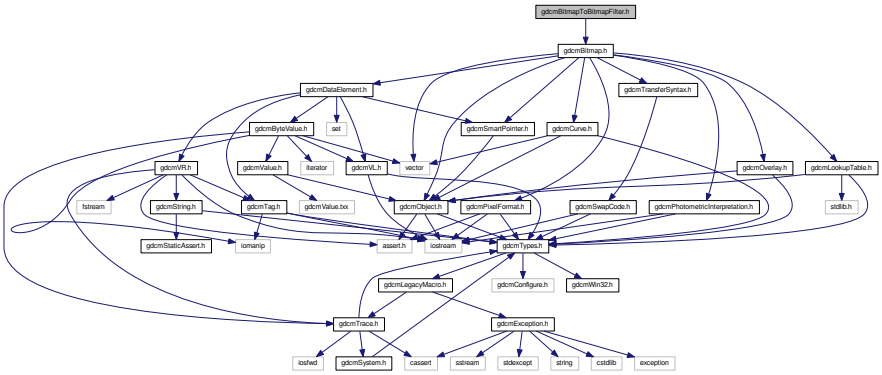
Namespaces

- gdcm

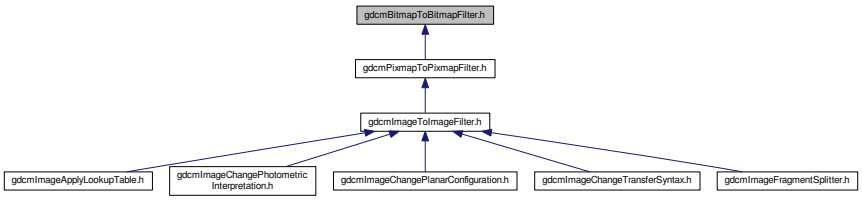
28.26 gdcmBitmapToBitmapFilter.h File Reference

```
#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 Super class for all filter taking an image and producing an output image.

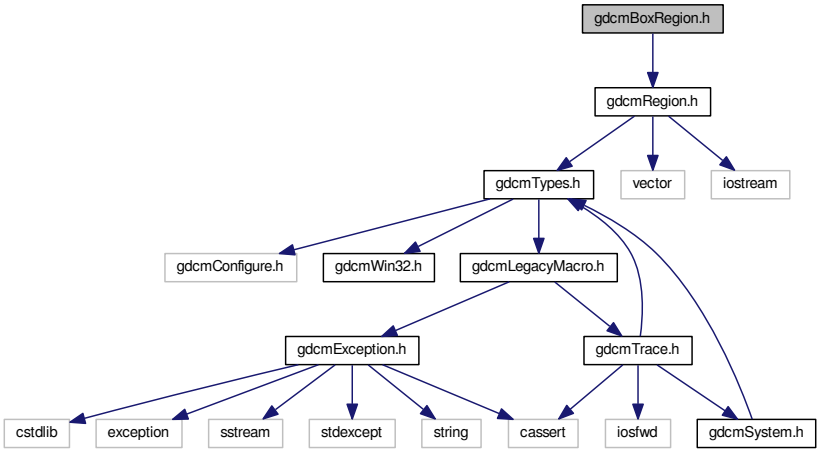
Namespaces

- gdcm

28.27 gdcmBoxRegion.h File Reference

```
#include "gdcmRegion.h"
```

Include dependency graph for gdcmBoxRegion.h:



Classes

- class gdcm::BoxRegion

  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)

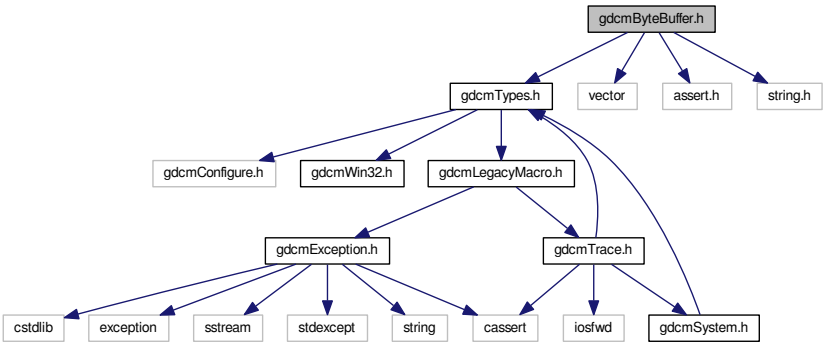
Namespaces

- gdcm

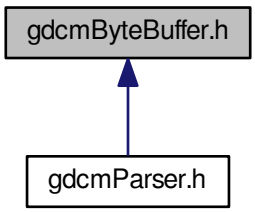
28.28 gdcmByteBuffer.h File Reference

```
#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.

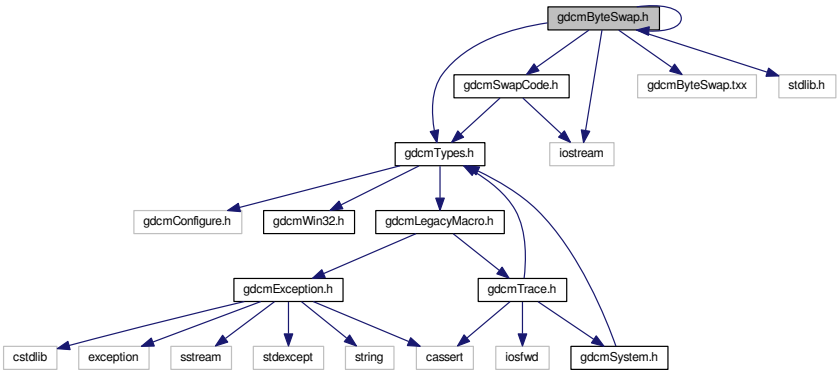
Namespaces

- gdcm

28.29 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.

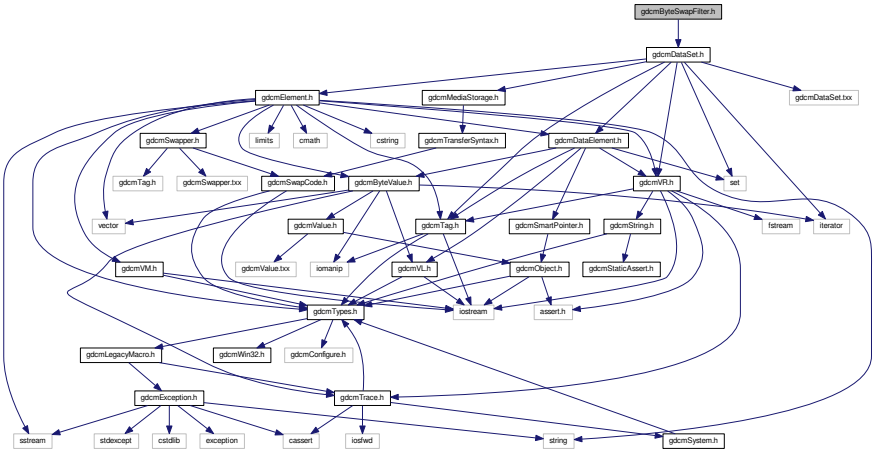
Namespaces

- gdcm

28.30 gdcmByteSwapFilter.h File Reference

```
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmByteSwapFilter.h:



Classes

- class gdcm::ByteSwapFilter

  ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??

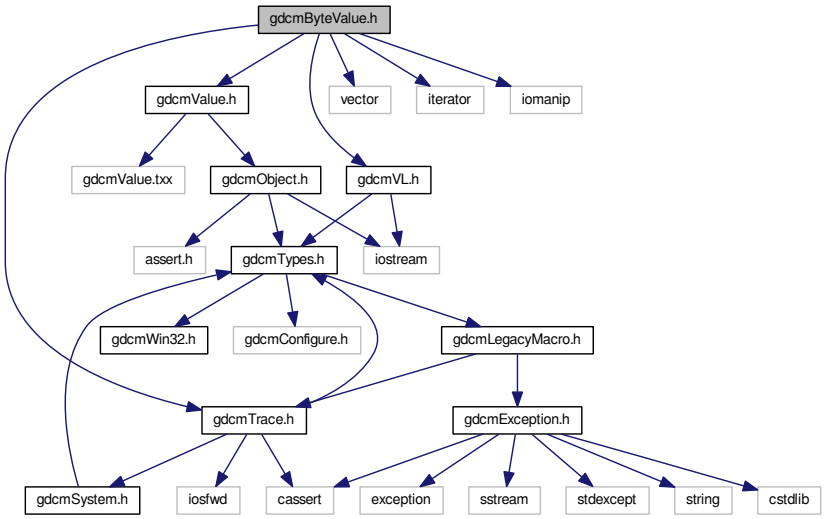
Namespaces

- gdcm

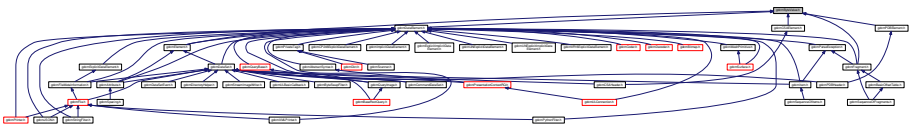
28.31 gdcmByteValue.h File Reference

```
#include "gdcmValue.h"
#include "gdcmTrace.h"
#include "gdcmVL.h"
#include <vector>
#include <iterator>
#include <iomanip>
```

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)

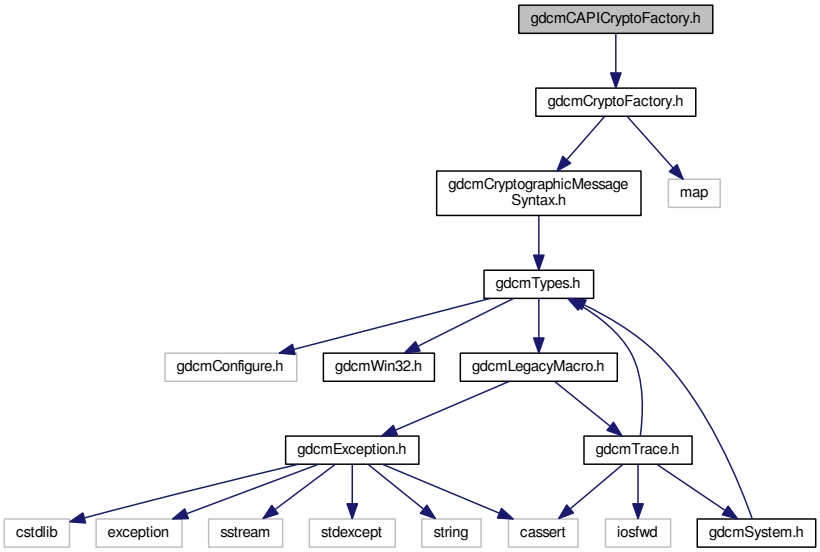
Namespaces

- gdcm

28.32 gdcmCAPICryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
```

Include dependency graph for gdcmCAPICryptoFactory.h:



Classes

- class gdcm::CAPICryptoFactory

Namespaces

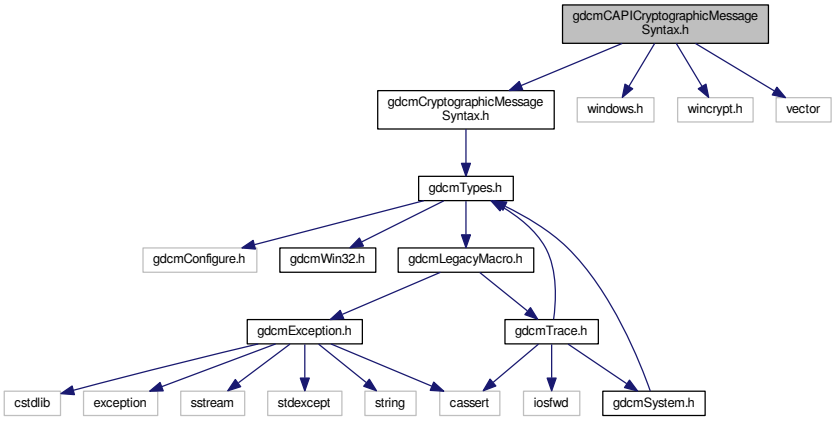
- gdcm

28.33 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

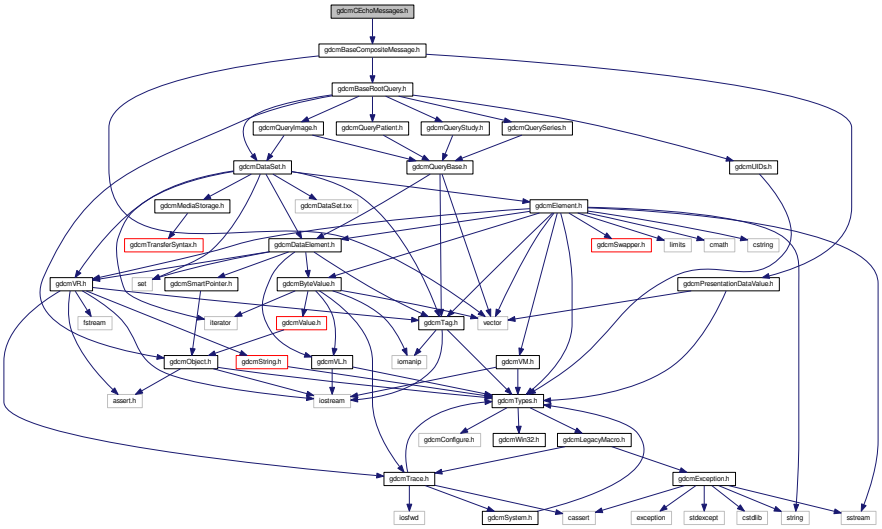
Namespaces

- gdcm

28.34 gdcmCEchoMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```

Include dependency graph for gdcmCEchoMessages.h:



Classes

- class gdcm::network::CEchoRQ

  CEchoRQ this file defines the messages for the cecho action.

- class gdcm::network::CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.

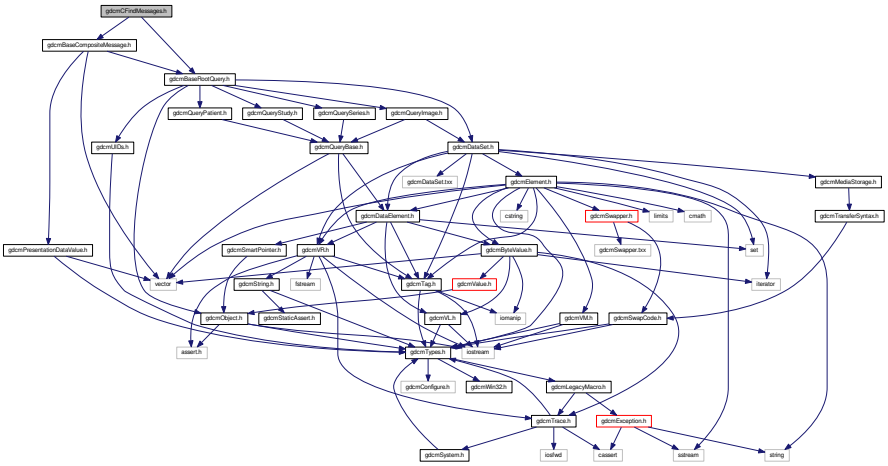
Namespaces

- gdcm
- gdcm::network

28.35 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 this file defines the messages for the cfind action.

- class gdcm::network::CFindRSP

  CFindRSP this file defines the messages for the cfind action.

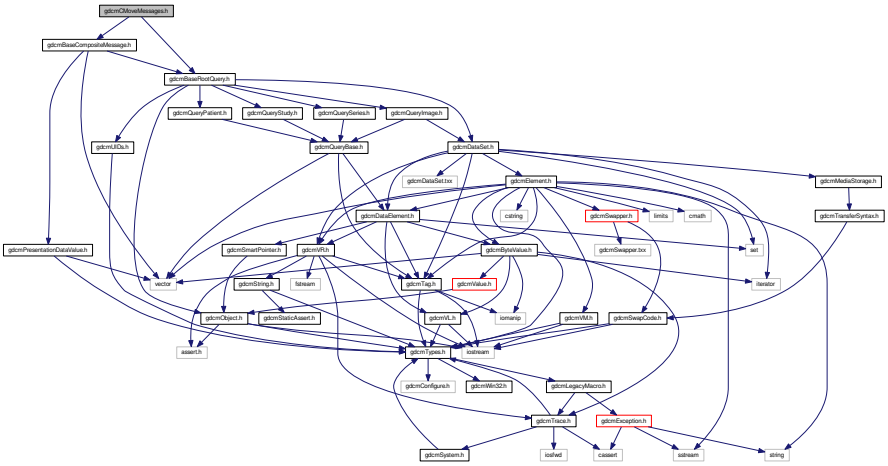
Namespaces

- gdcm
- gdcm::network

28.36 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 this file defines the messages for the cmove action.

- class gdcm::network::CMoveRSP

  CMoveRSP this file defines the messages for the cmove action.

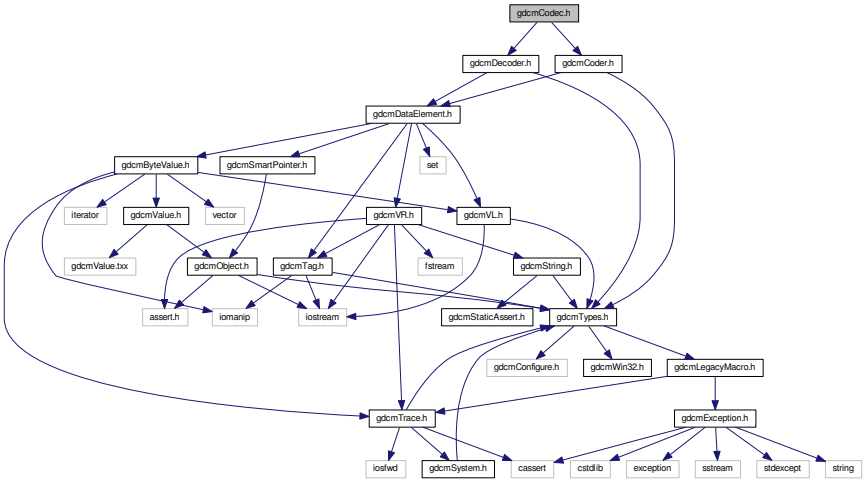
Namespaces

- gdcm
- gdcm::network

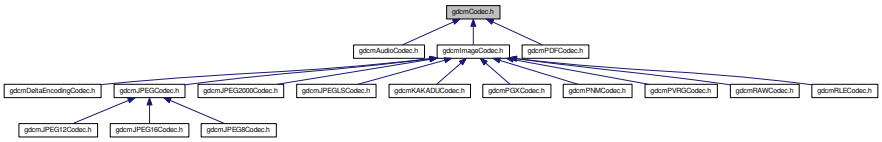
28.37 gdcmCodec.h File Reference

```
#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.

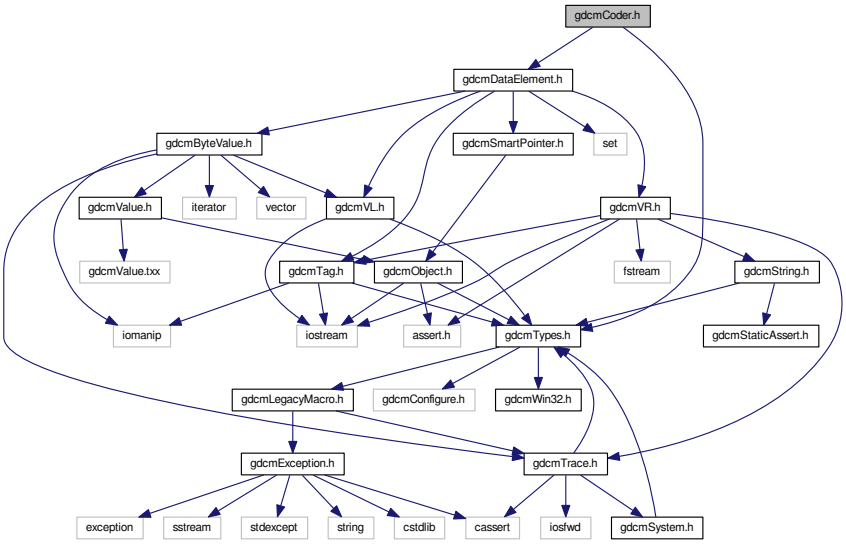
Namespaces

- gdcm

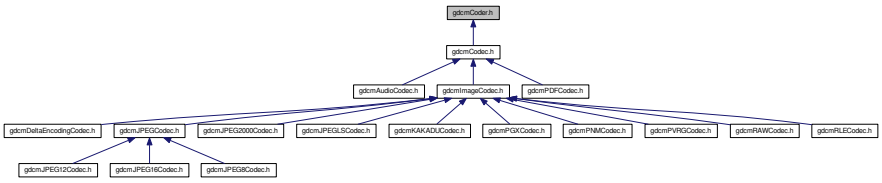
28.38 gdcmCoder.h File Reference

```
#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.

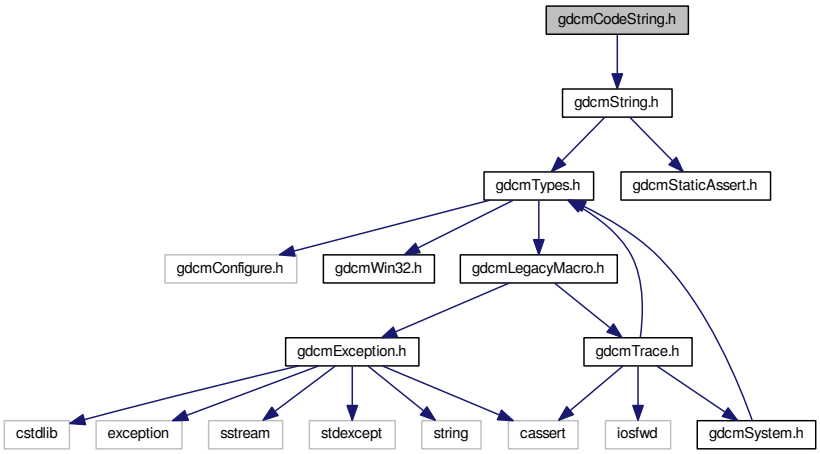
Namespaces

- gdcm

28.39 gdcmCodeString.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmCodeString.h:



Classes

- class gdcm::CodeString

  CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct.

Namespaces

- gdcm

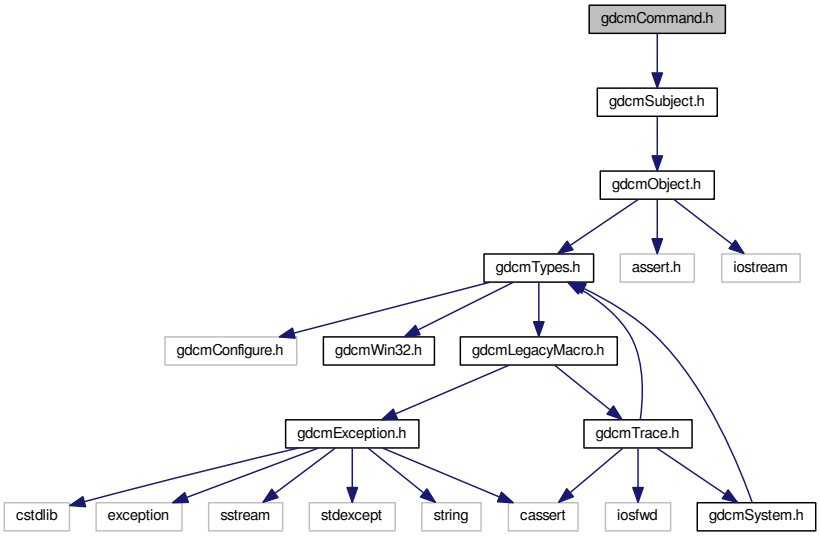
Functions

- 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)

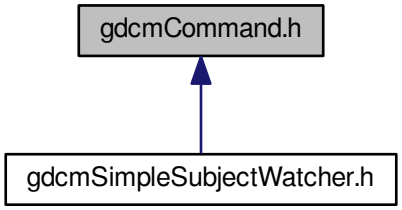
28.40 gdcmCommand.h File Reference

```
#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.

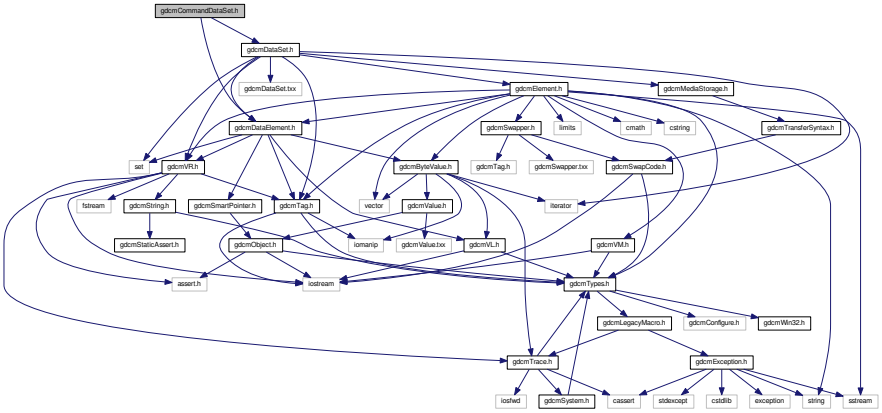
Namespaces

- gdcm

28.41 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

- gdcm

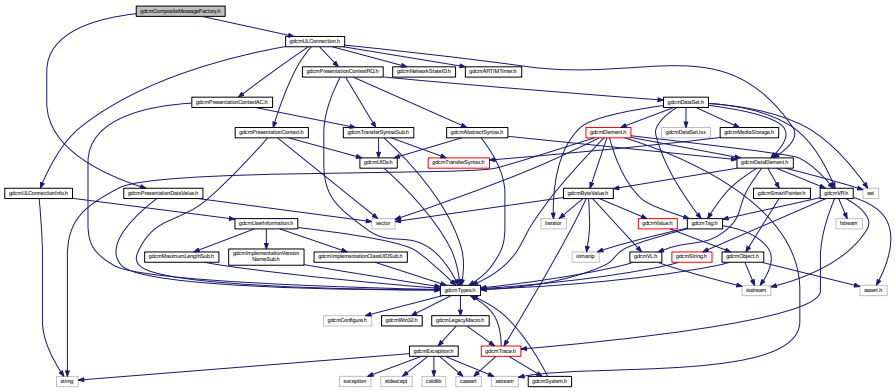
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CommandDataSet &val)

28.42 gdcmCompositeMessageFactory.h File Reference

```
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnection.h"
```

Include dependency graph for gdcmCompositeMessageFactory.h:



Classes

- class gdcm::network::CompositeMessageFactory

  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).

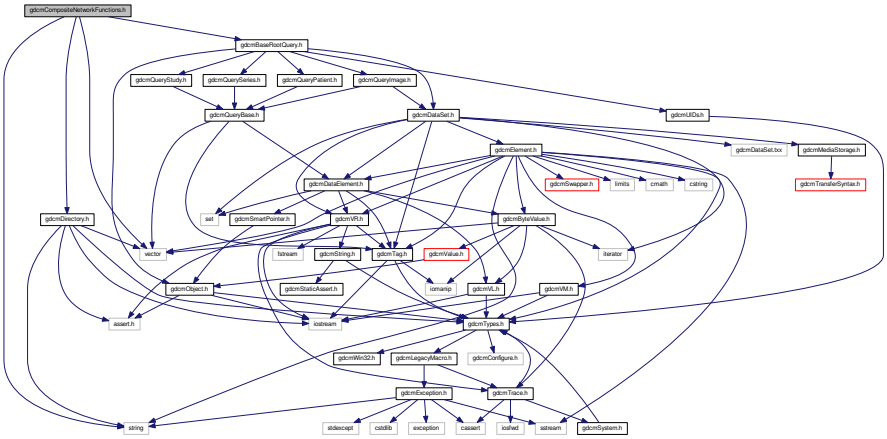
Namespaces

- gdcm
- gdcm::network

28.43 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 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:

Namespaces

- gdcm

28.44 gdcmConstCharWrapper.h File Reference

Classes

- class gdcm::ConstCharWrapper

  Do not use me.

Namespaces

- gdcm

28.45 gdcmconv.dox File Reference

28.46 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.

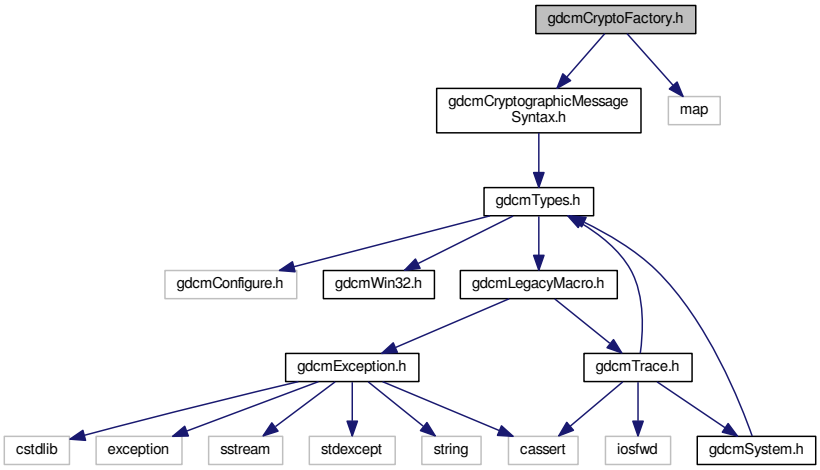
Namespaces

- gdcm

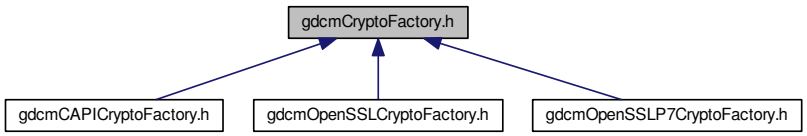
28.47 gdcmCryptoFactory.h File Reference

```
#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.

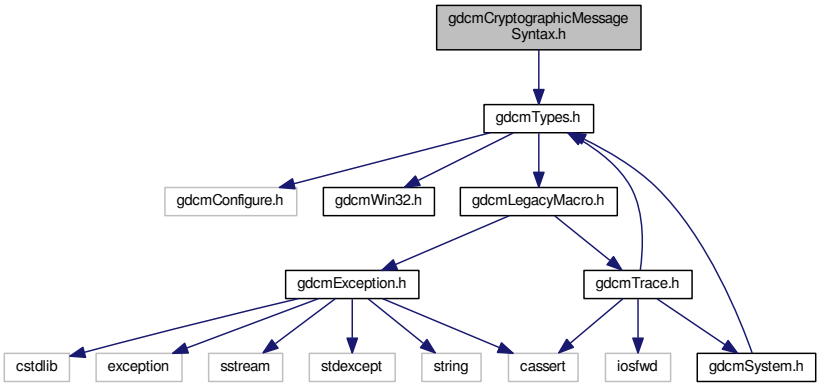
Namespaces

- gdcm

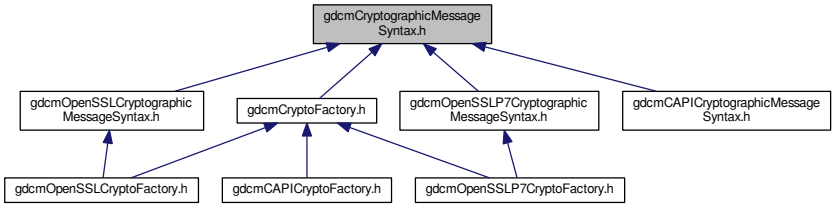
28.48 gdcmCryptographicMessageSyntax.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::CryptographicMessageSyntax

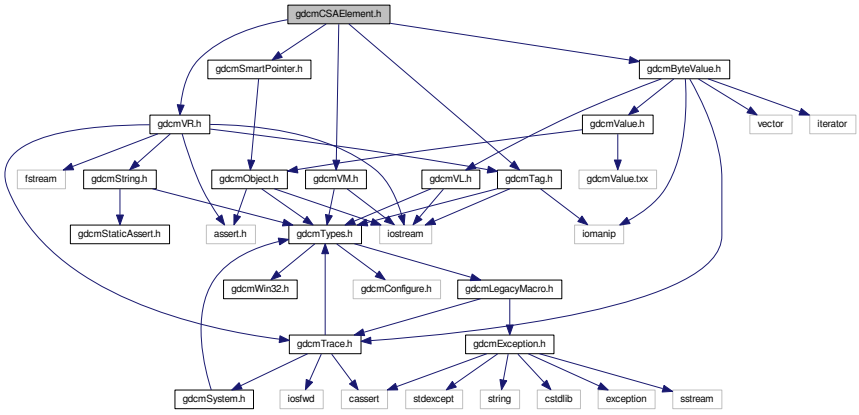
Namespaces

- gdcm

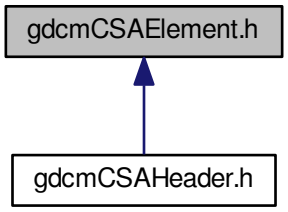
28.49 gdcmCSAElement.h File Reference

```
#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

- gdcm

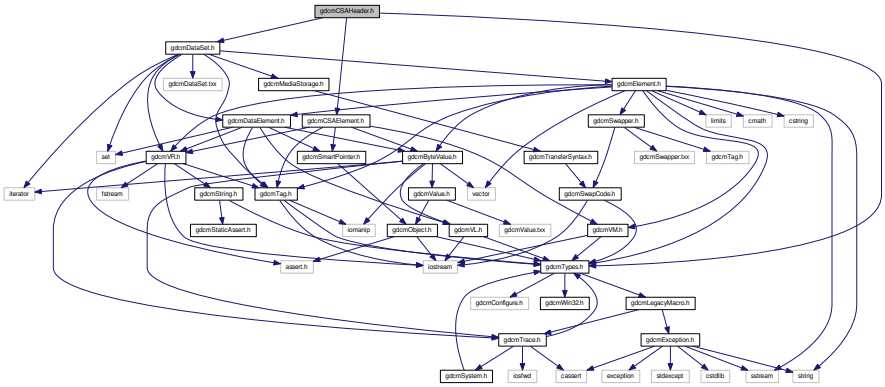
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)

28.50 gdcmCSAHeader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmCSAElement.h"
```

Include dependency graph for gdcmCSAHeader.h:



Classes

- class gdcm::CSAHeader

  Class for CSAHeader.

Namespaces

- gdcm

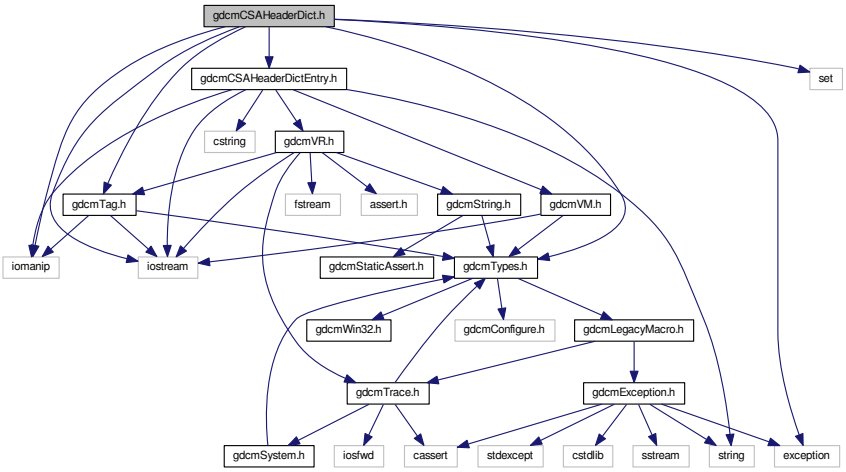
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)

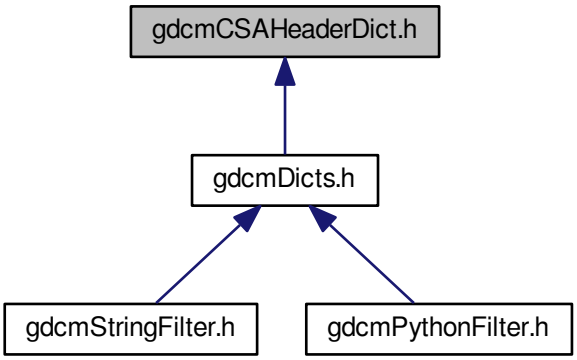
28.51 gdcmCSAHeaderDict.h File Reference

```
#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

- gdcm

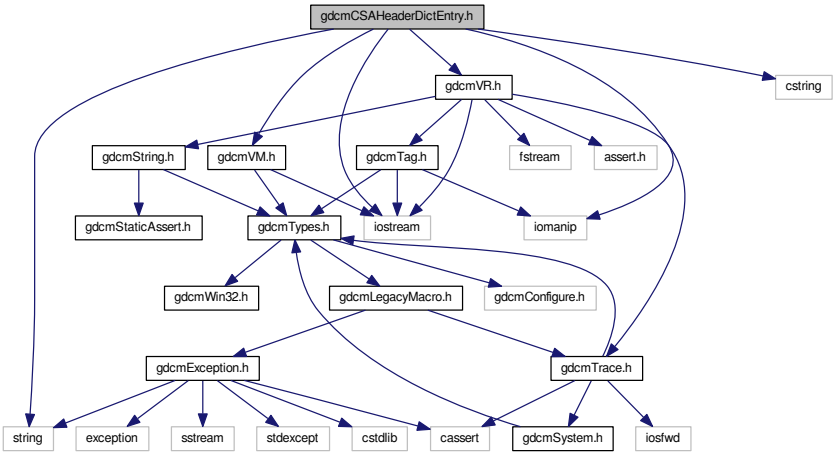
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)

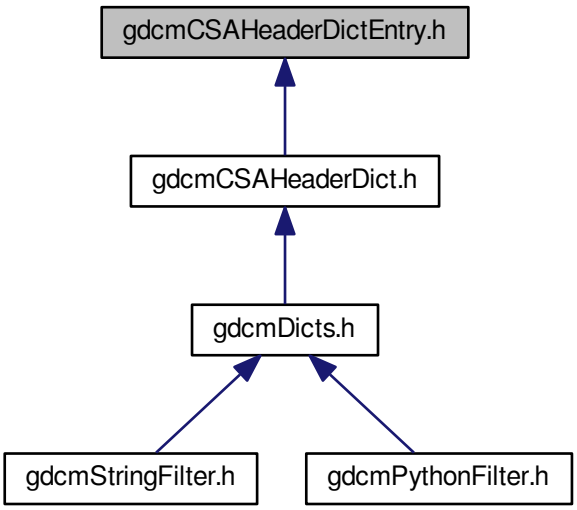
28.52 gdcmCSAHeaderDictEntry.h File Reference

```
#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 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.

Namespaces

- gdcm

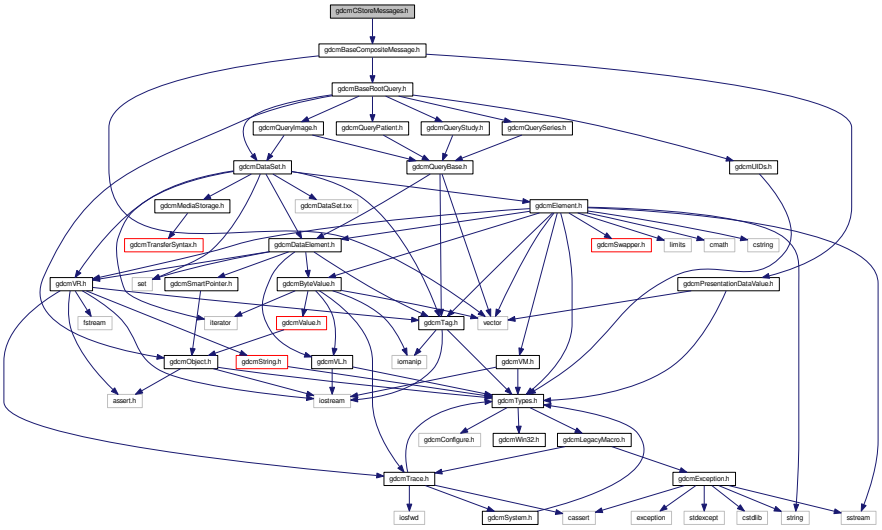
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)

28.53 gdcmCStoreMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```

Include dependency graph for gdcmCStoreMessages.h:



Classes

- class gdcm::network::CStoreRQ

  CStoreRQ this file defines the messages for the cecho action.

- class gdcm::network::CStoreRSP

  CStoreRSP this file defines the messages for the cecho action.

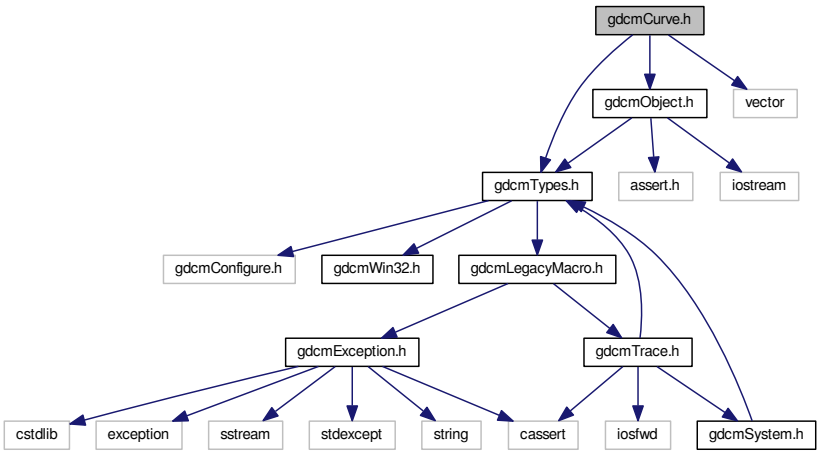
Namespaces

- gdcm
- gdcm::network

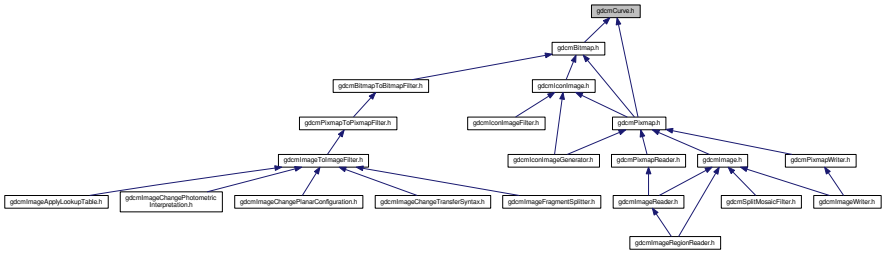
28.54 gdcmCurve.h File Reference

```
#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 WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

Namespaces

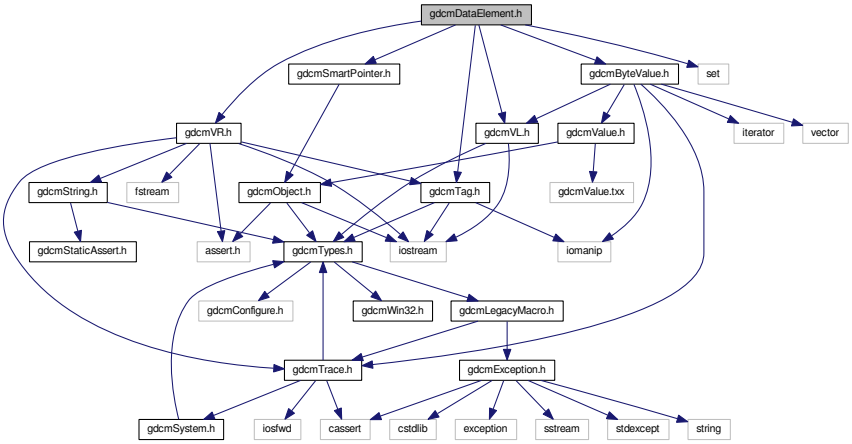
- gdcm

28.55 gdcmDataElement.h File Reference

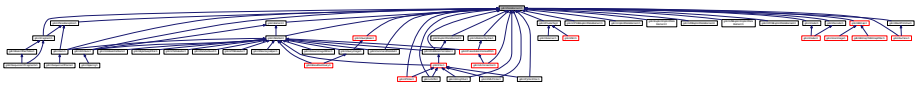
```
#include "gdcmTag.h"
```

```
#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

- gdcm

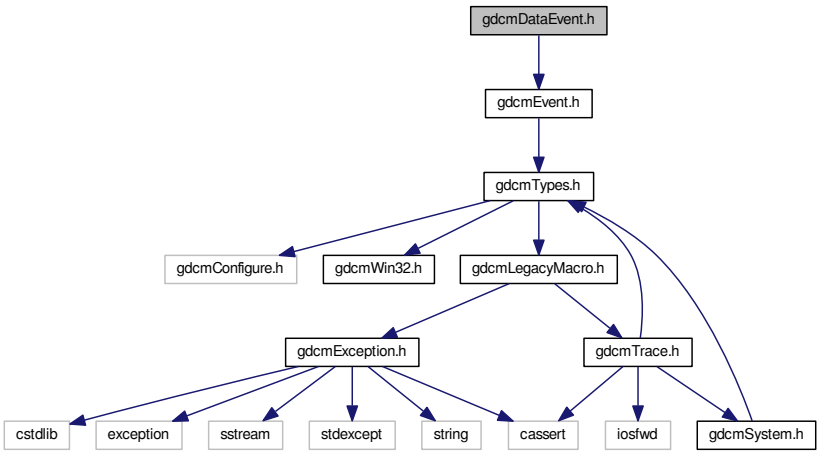
Functions

- bool gdcm::operator!= (const DataElement &lhs, const DataElement &rhs)
- std::ostream & gdcm::operator<< (std::ostream &os, const DataElement &val)

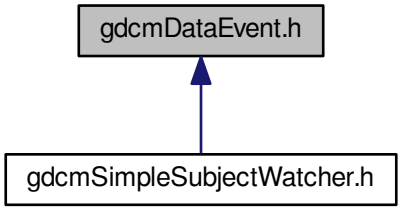
28.56 gdcmDataEvent.h File Reference

```
#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.

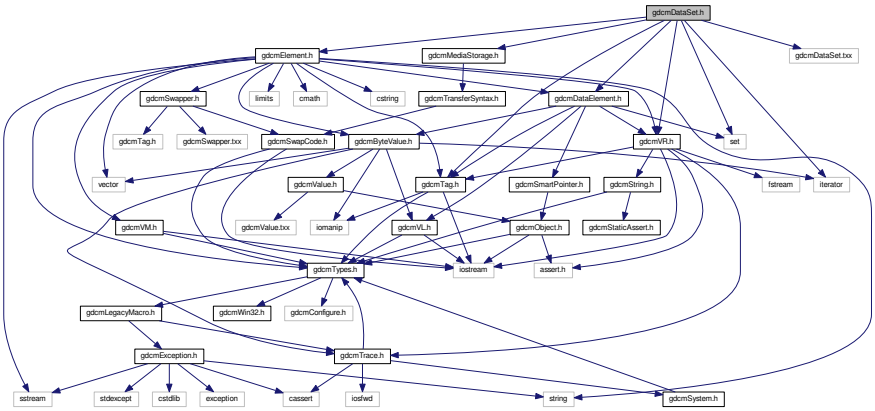
Namespaces

- gdcm

28.57 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) A Data Set represents an instance of a real world Information Object.

Namespaces

- gdcm

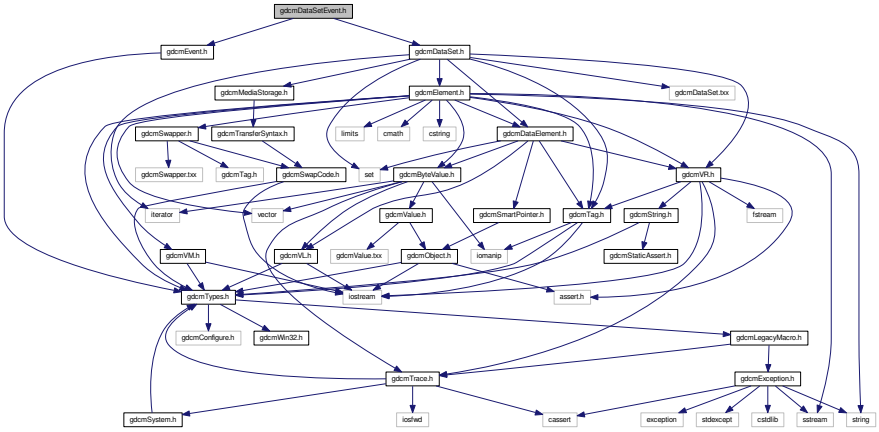
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)

28.58 gdcmDataSetEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDataSetEvent.h:



Classes

- class gdcm::DataSetEvent

  DataSetEvent Special type of event triggered during the DataSet store/move process.

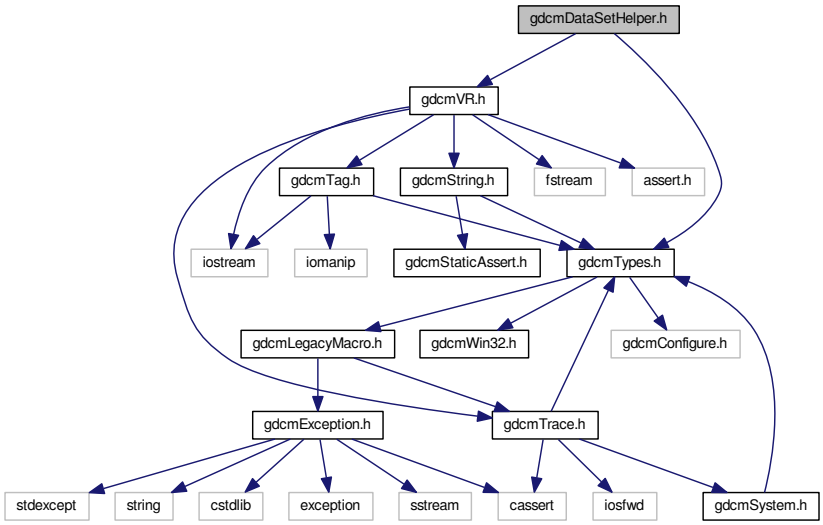
Namespaces

- gdcm

28.59 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)

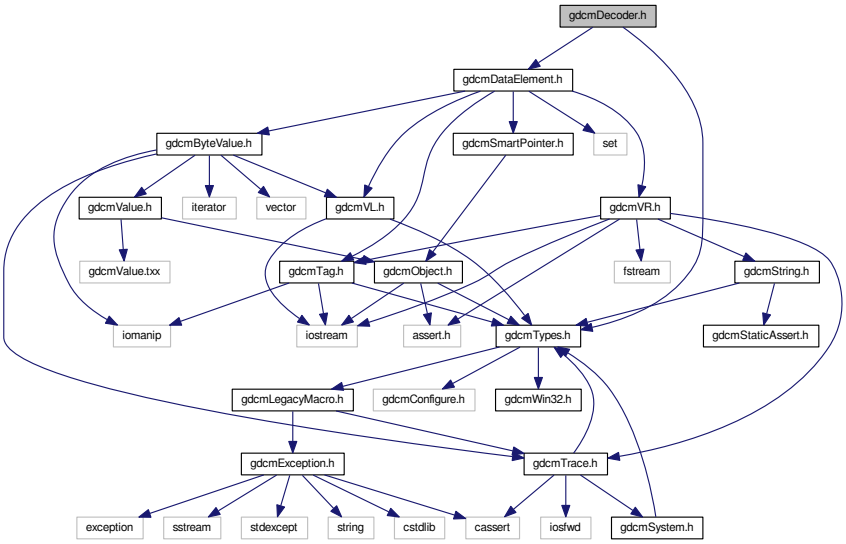
Namespaces

- gdcm

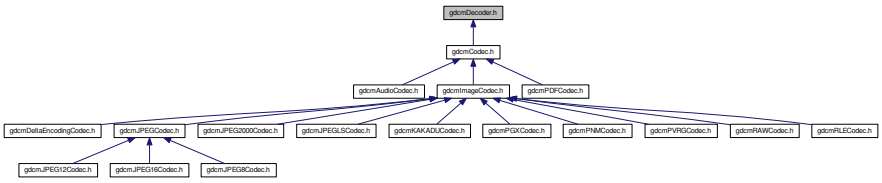
28.60 gdcmDecoder.h File Reference

```
#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.

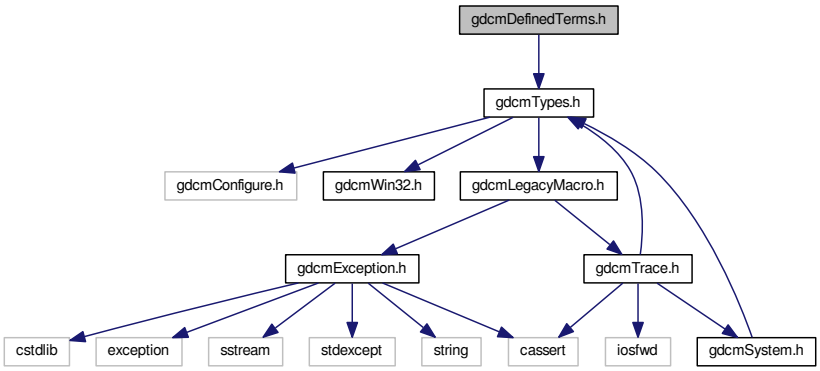
Namespaces

- gdcm

28.61 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.

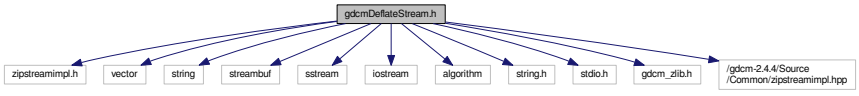
Namespaces

- gdcm

28.62 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

Include dependency graph for gdcmDeflateStream.h:

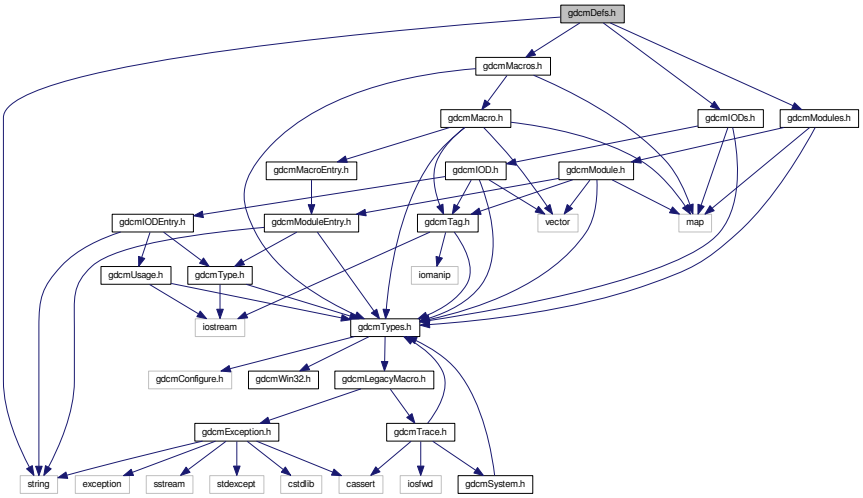


28.63 gdcmDefs.h File Reference

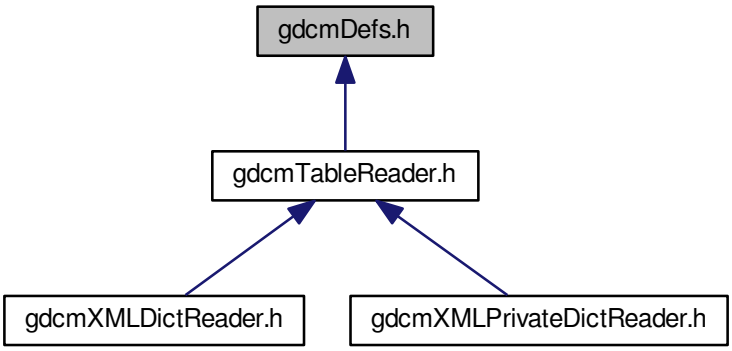
```
#include "gdcmModules.h"
```

```
#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'.

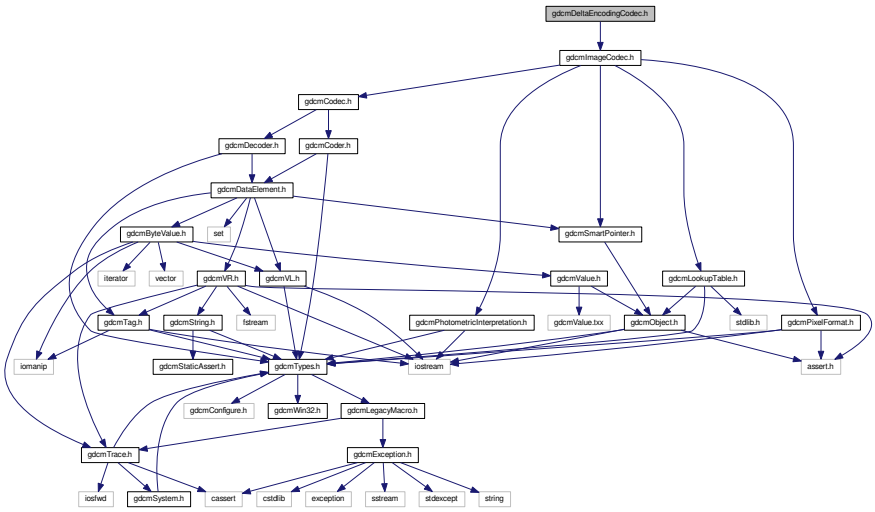
Namespaces

- gdcm

28.64 gdcmDeltaEncodingCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmDeltaEncodingCodec.h:



Classes

- class gdcm::DeltaEncodingCodec

  DeltaEncodingCodec compression used by some private vendor.

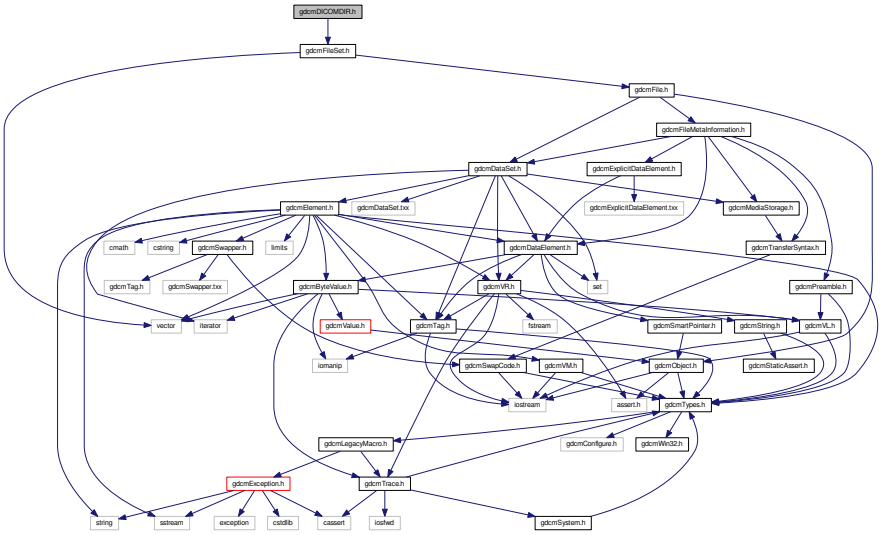
Namespaces

- gdcm

28.65 gdcmDICOMDIR.h File Reference

```
#include "gdcmFileSet.h"
```

Include dependency graph for gdcmDICOMDIR.h:



Classes

- class gdcm::DICOMDIR

  DICOMDIR class.

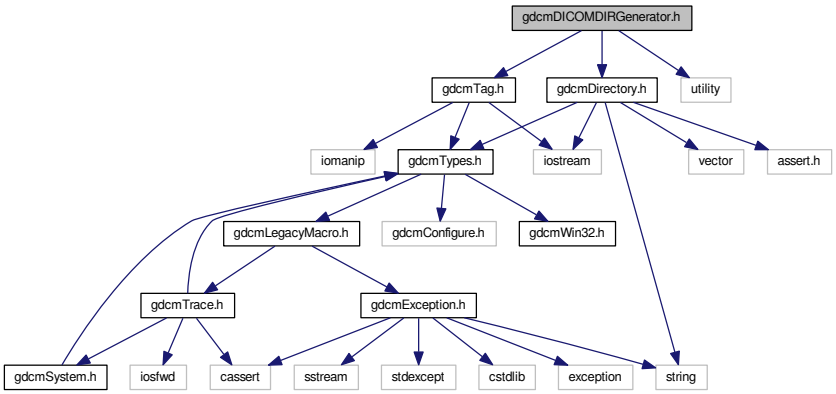
Namespaces

- gdcm

28.66 gdcmDICOMDIRGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <utility>
```

Include dependency graph for gdcmDICOMDIRGenerator.h:



Classes

- class gdcm::DICOMDIRGenerator

  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.

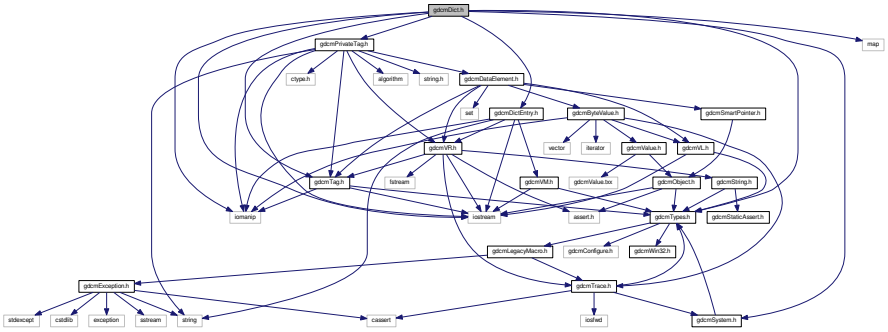
Namespaces

- gdcm

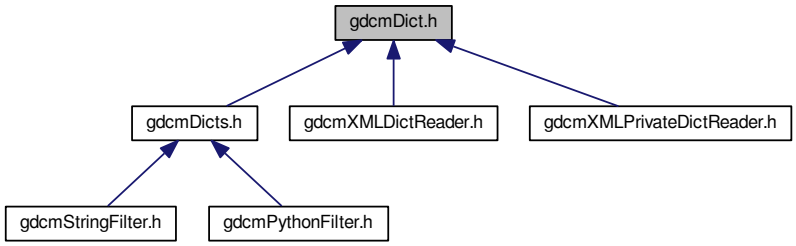
28.67 gdcmDict.h File Reference

```
#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

- gdcm

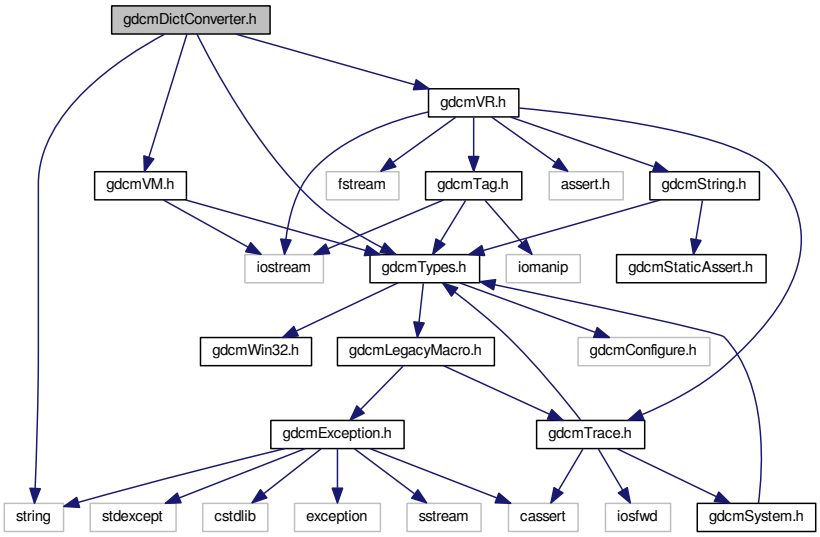
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)
- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)

28.68 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:

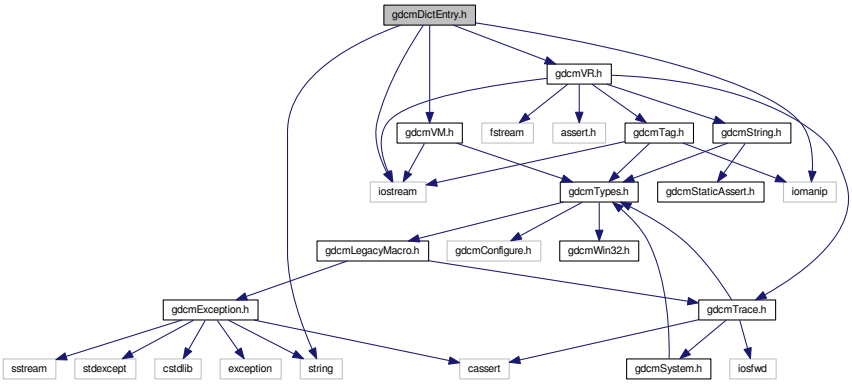
Namespaces

- gdcm

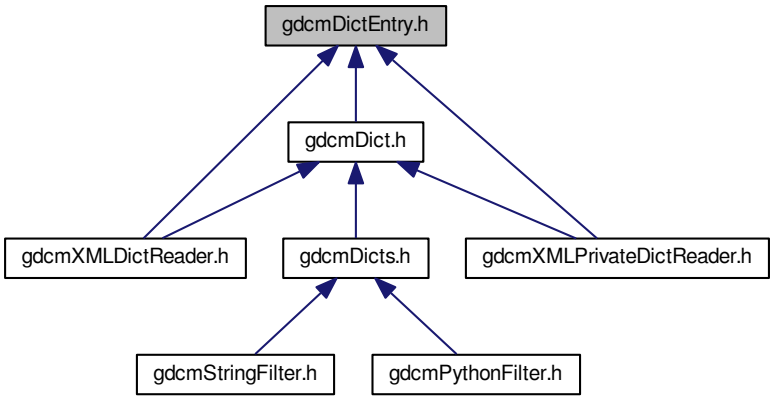
28.69 gdcmDictEntry.h File Reference

```
#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 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.

Namespaces

- gdcm

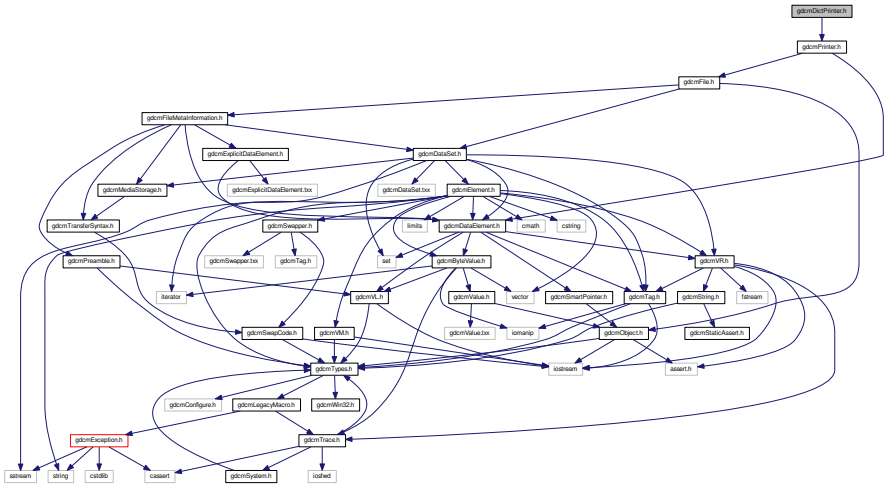
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const DictEntry &val)

28.70 gdcmDictPrinter.h File Reference

```
#include "gdcmPrinter.h"
```

Include dependency graph for gdcmDictPrinter.h:



Classes

- class gdcm::DictPrinter

  DictPrinter class.

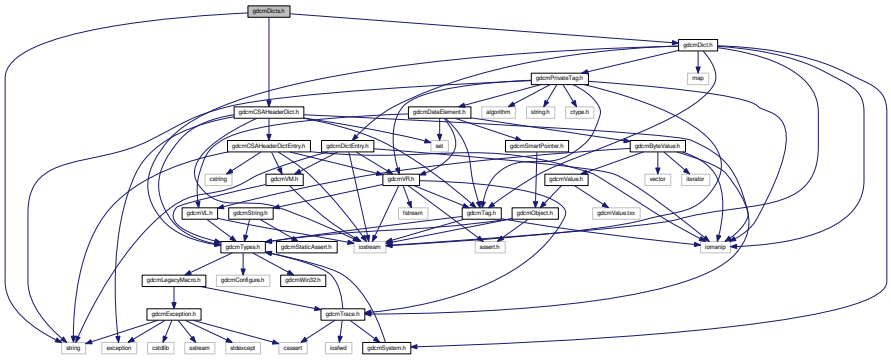
Namespaces

- gdcm

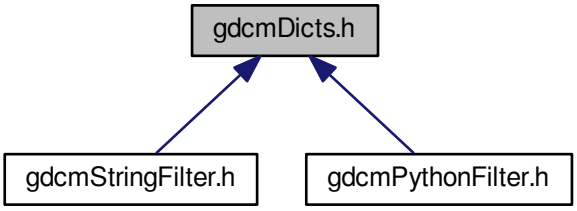
28.71 gdcmDicts.h File Reference

```
#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

- gdcm

Functions

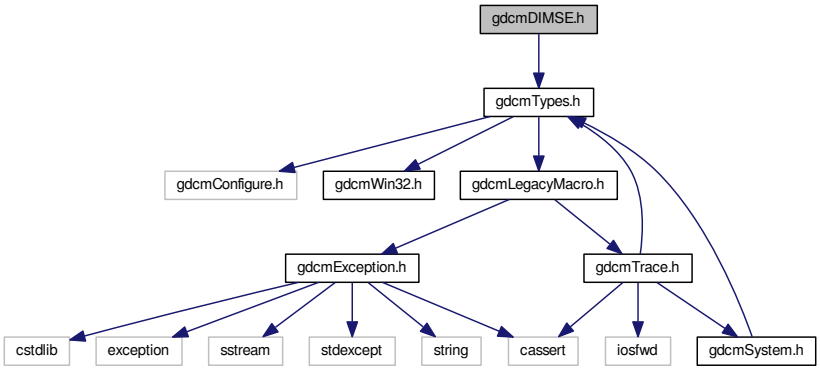
- std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)

28.72 gdcmdiff.dox File Reference

28.73 gdcmDIMSE.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDIMSE.h:



Classes

- class gdcm::network::CEchoRQ

  CEchoRQ this file defines the messages for the cecho action.

- class gdcm::network::CEchoRSP

  CEchoRSP this file defines the messages for the cecho action.

- class gdcm::network::CFind
- class gdcm::network::DIMSE

  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)

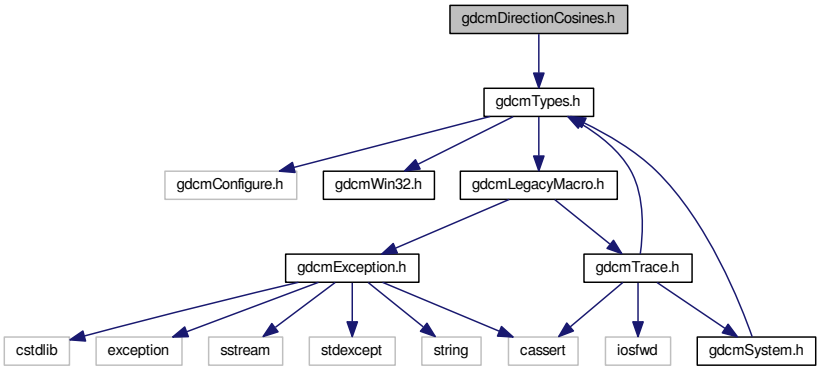
Namespaces

- gdcm
- gdcm::network

28.74 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDirectionCosines.h:



Classes

- class gdcm::DirectionCosines

  class to handle DirectionCosines

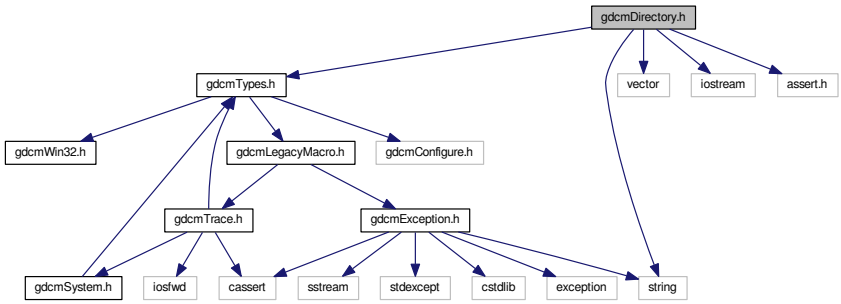
Namespaces

- gdcm

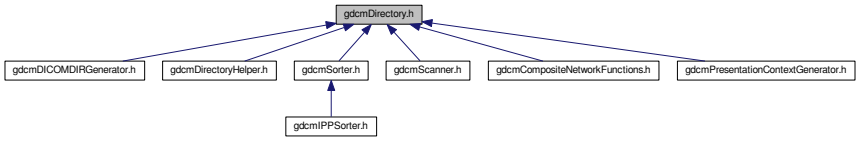
28.75 gdcmDirectory.h File Reference

```
#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

- gdcm

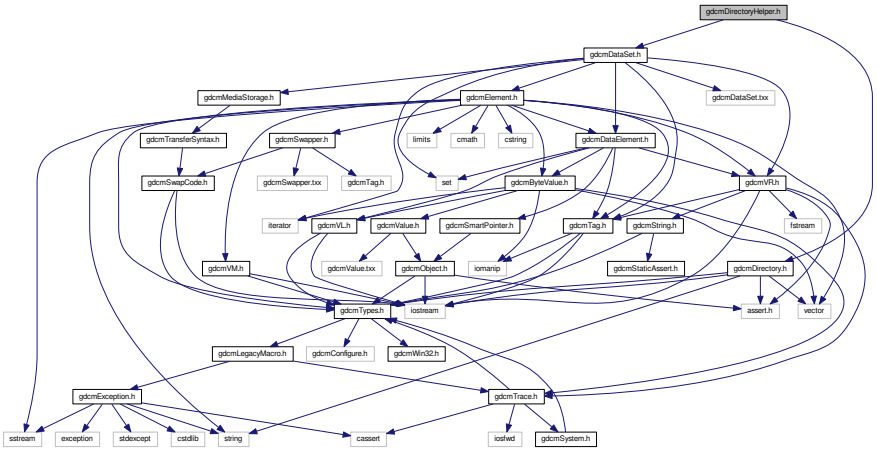
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Directory &d)

28.76 gdcmDirectoryHelper.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDirectoryHelper.h:



Classes

- class gdcm::DirectoryHelper

  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.

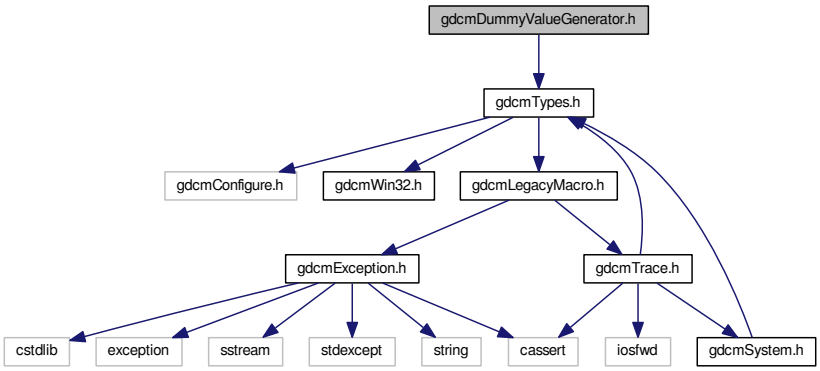
Namespaces

- gdcm

28.77 gdcmDummyValueGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDummyValueGenerator.h:



Classes

- class gdcm::DummyValueGenerator

  Class for generating dummy value.

Namespaces

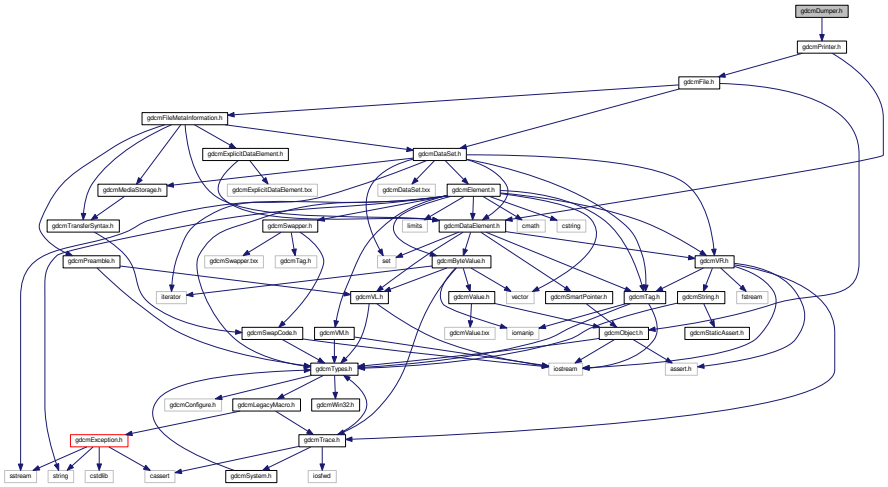
- gdcm

28.78 gdcmdump.dox File Reference

28.79 gdcmDumper.h File Reference

```
#include "gdcmPrinter.h"
```

Include dependency graph for gdcmDumper.h:



Classes

- class gdcm::Dumper

  Codec class.

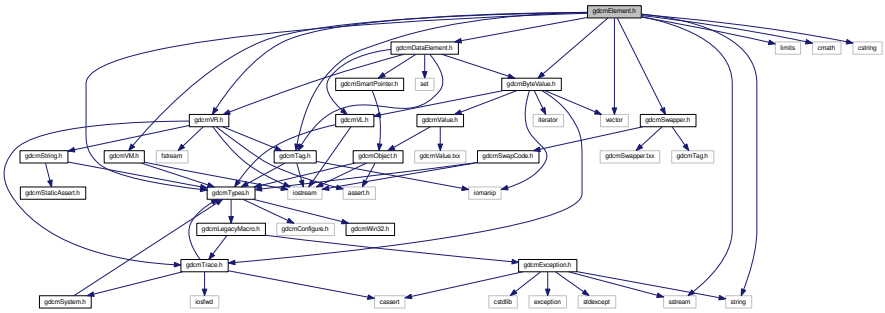
Namespaces

- gdcm

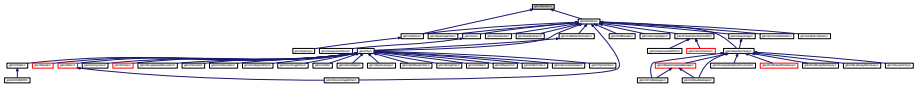
28.80 gdcmElement.h File Reference

```
#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_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 >
- singleton gdcm::EncodingImplementation< T >

  EncodingImplementation.
- class gdcm::EncodingImplementation< VR::VRASCII >
- class gdcm::EncodingImplementation< VR::VRBINARY >
- struct gdcm::ignore_char

Namespaces

- gdcm

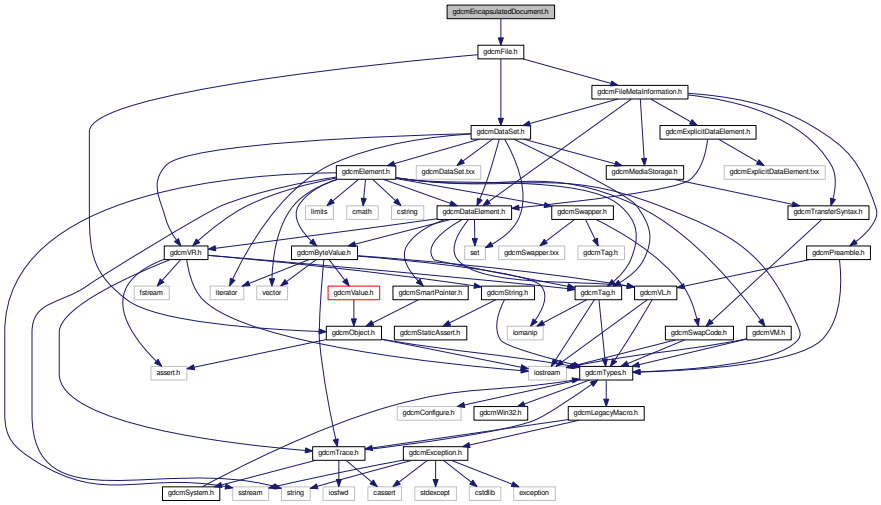
Functions

- ignore_char const gdcm::backslash ('\\')
- std::istream & gdcm::operator>> (std::istream &in, ignore_char const &ic)
- template<typename Float >
  std::string gdcm::to_string (Float data)

28.81 gdcmEncapsulatedDocument.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmEncapsulatedDocument.h:



Classes

- class gdcm::EncapsulatedDocument

  EncapsulatedDocument.

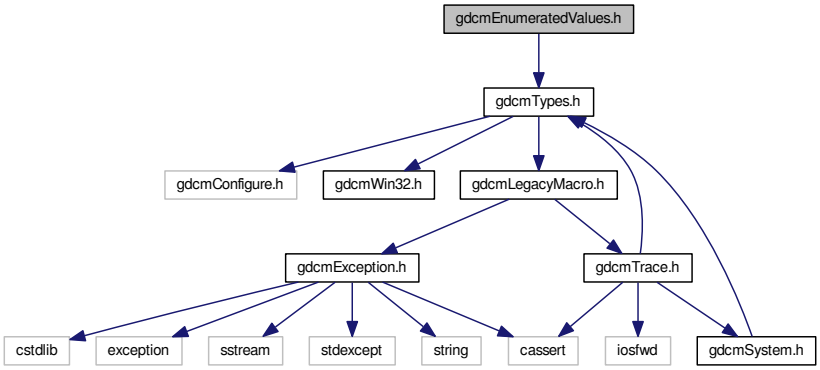
Namespaces

- gdcm

28.82 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:

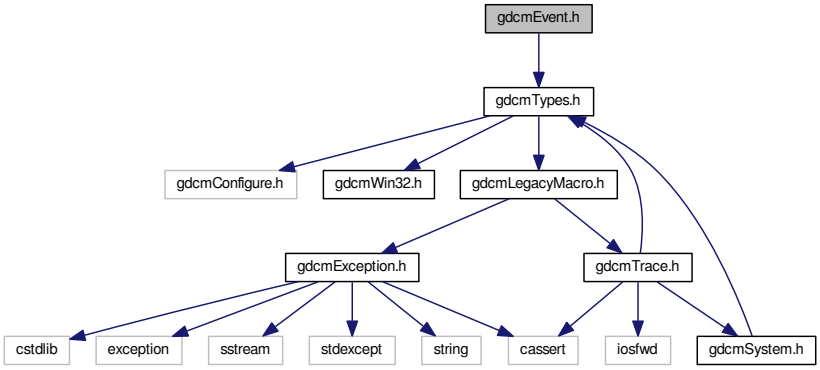
Namespaces

- gdcm

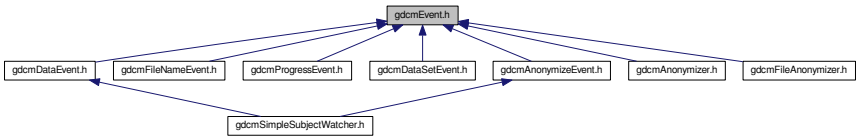
28.83 gdcmEvent.h File Reference

```
#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

- gdcm

Macros

- #define gdcmEventMacro(classname, super)

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, Event &e)

  Generic inserter operator for Event and its subclasses.

28.83.1 Macro Definition Documentation

28.83.1.1 #define gdcmEventMacro(*classname, super*)

Value:

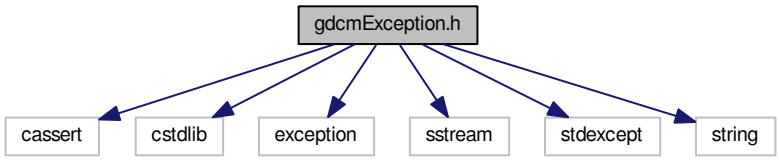
```
 \
class  classname : public super { \
  public: \
    typedef classname Self; \
    typedef super Superclass; \
    classname() {} \
```

```
    virtual ~classname() {} \
    virtual const char * GetEventName() const { return #classname; } \
    virtual bool CheckEvent(const ::gdcm::Event* e) const \
      { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdcm::Event* MakeObject() const \
      { return new Self; } \
    classname(const Self&s) : super(s){}; \
  private: \
    void operator=(const Self&); \
}
```

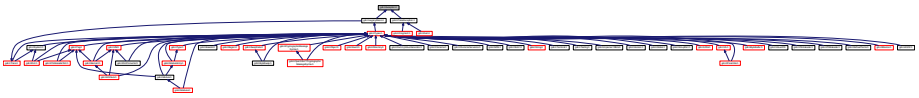
28.84 gdcmException.h File Reference

```
#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

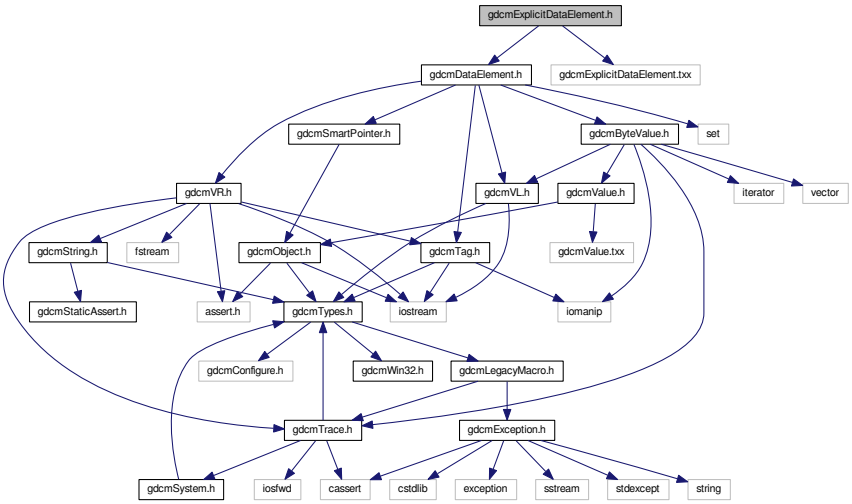
- gdcm

28.85 gdcmExplicitDataElement.h File Reference

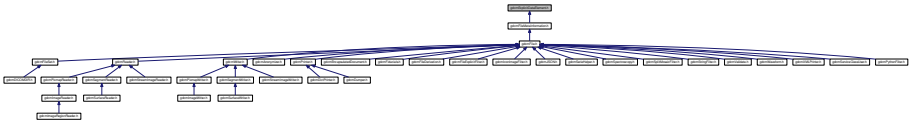
```
#include "gdcmDataElement.h"
```

```
#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.

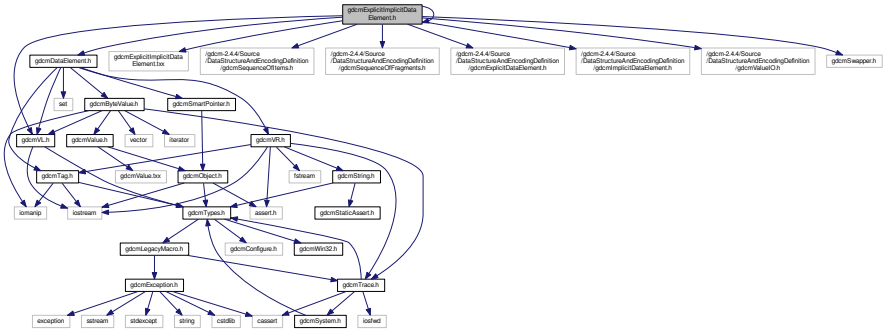
Namespaces

- gdcm

28.86 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.

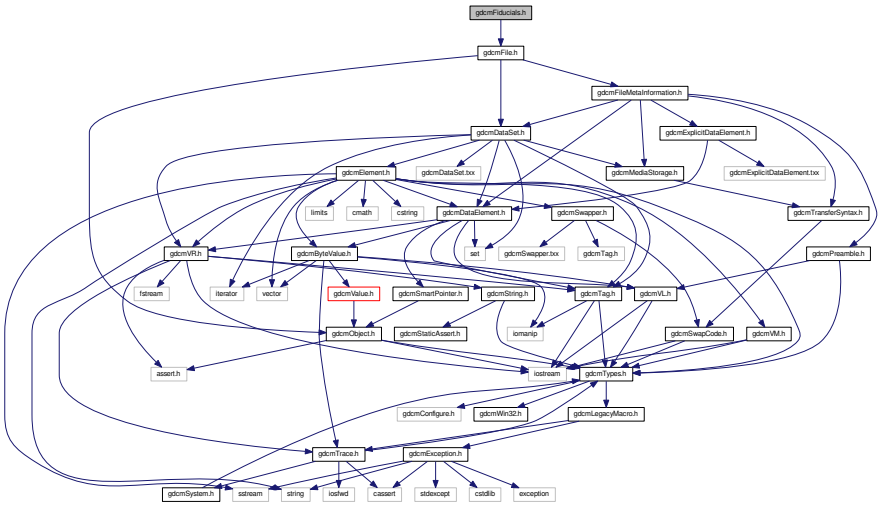
Namespaces

- gdcm

28.87 gdcmFiducials.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFiducials.h:



Classes

- class gdcm::Fiducials

  Fiducials.

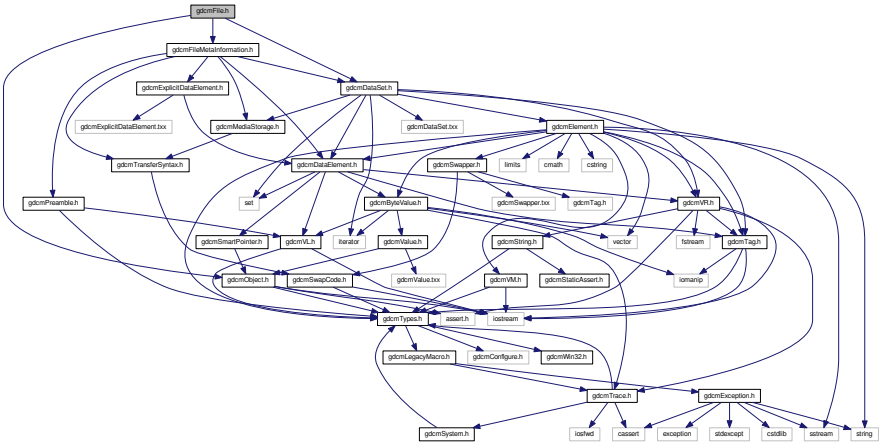
Namespaces

- gdcm

28.88 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 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.

Namespaces

- gdcm

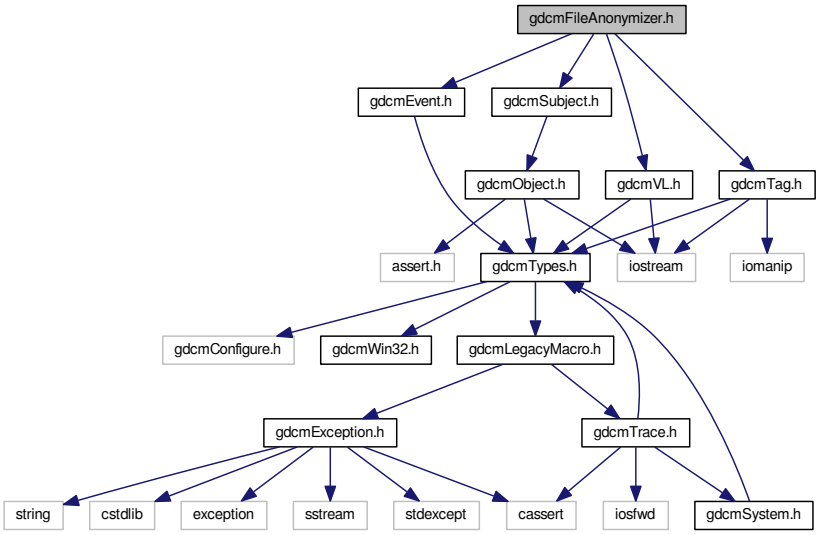
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const File &val)

28.89 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.

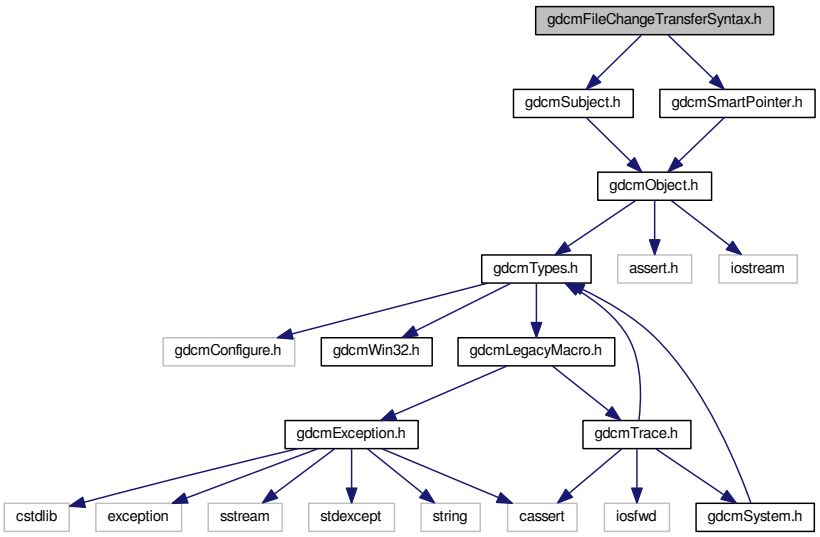
Namespaces

- gdcm

28.90 gdcmFileChangeTransferSyntax.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



Classes

- class gdcm::FileChangeTransferSyntax

  FileChangeTransferSyntax.

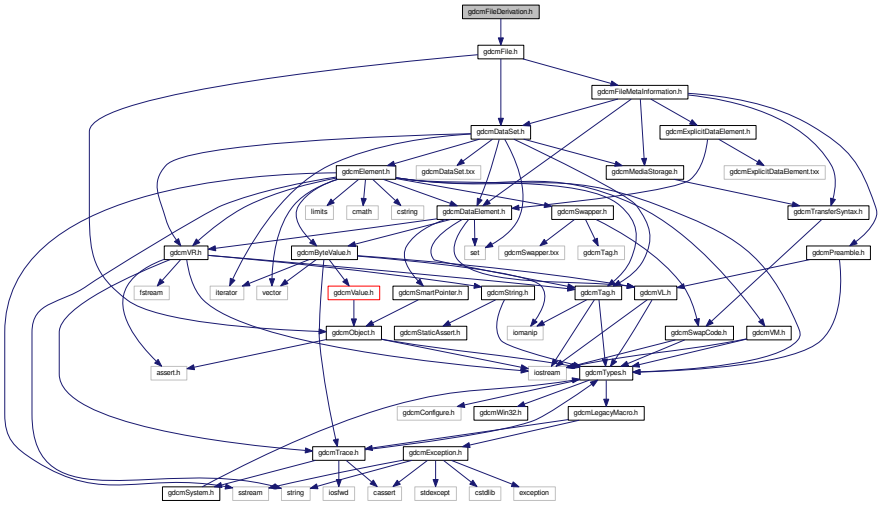
Namespaces

- gdcm

28.91 gdcmFileDerivation.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFileDerivation.h:



Classes

- class gdcm::FileDerivation

  FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation Code Sequence.

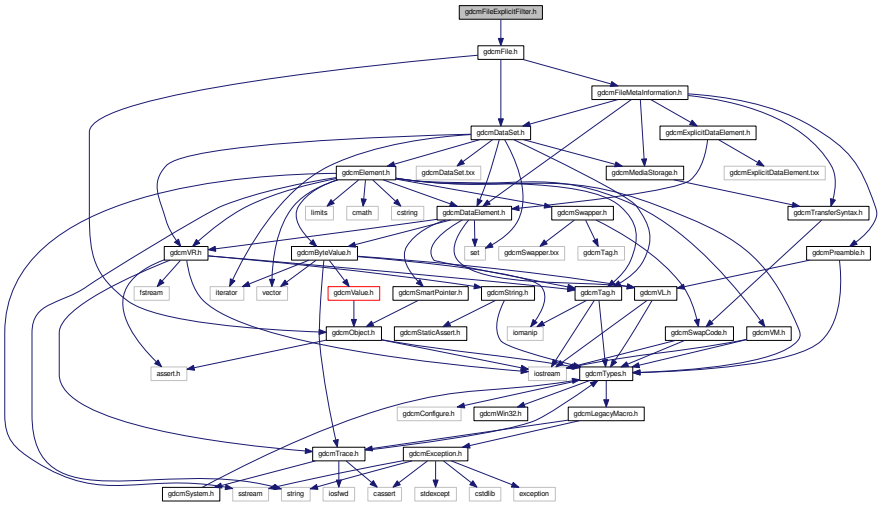
Namespaces

- gdcm

28.92 gdcmFileExplicitFilter.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmFileExplicitFilter.h:



Classes

- class gdcm::FileExplicitFilter

  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 reprensentation, the VR is not stored directly in the file.

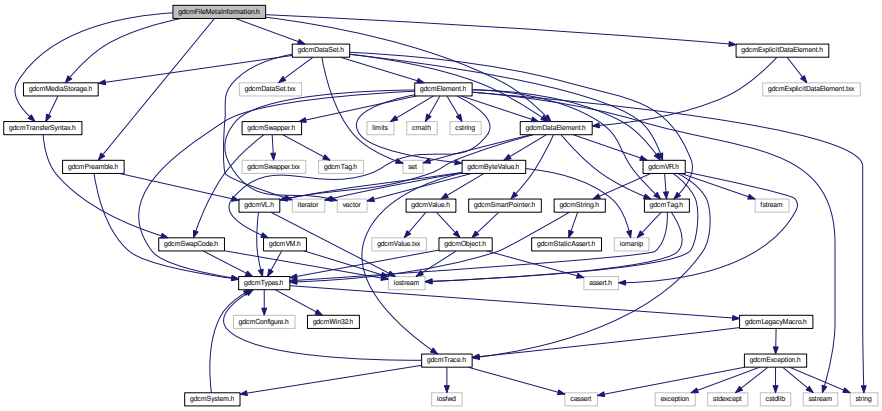
Namespaces

- gdcm

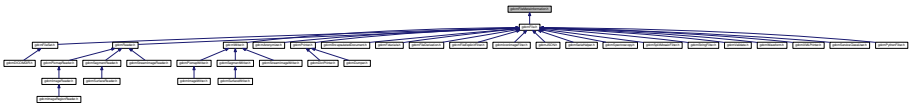
28.93 gdcmFileMetaInformation.h File Reference

```
#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

- gdcm

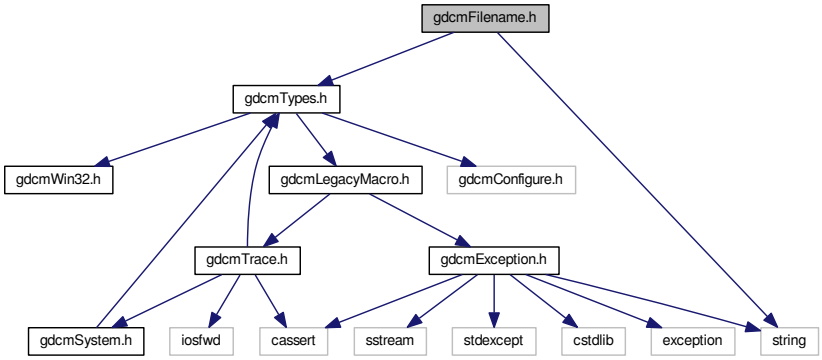
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileMetaInformation &val)

28.94 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.

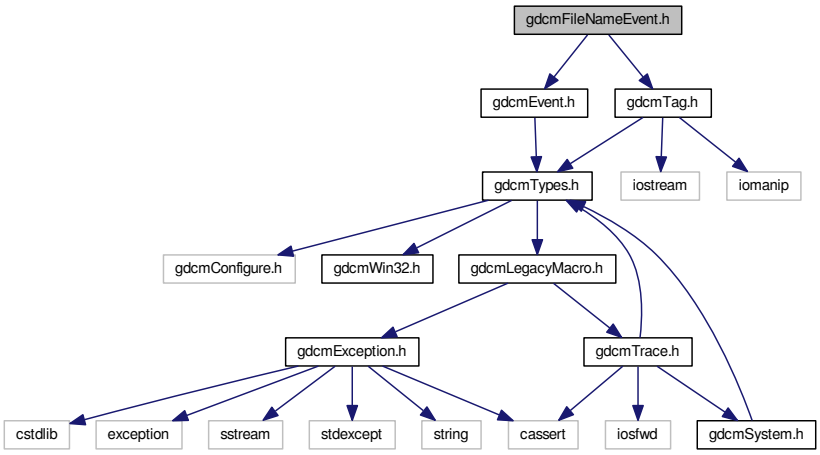
Namespaces

- gdcm

28.95 gdcmFileNameEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmFileNameEvent.h:



Classes

- class gdcm::FileNameEvent

  FileNameEvent Special type of event triggered during processing of FileSet.

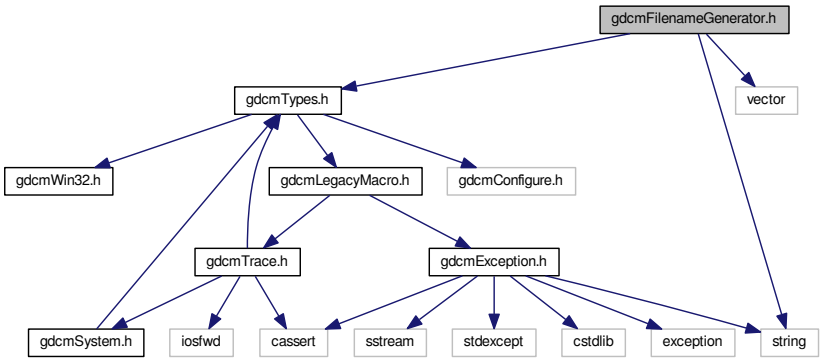
Namespaces

- gdcm

28.96 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
```

Include dependency graph for gdcmFilenameGenerator.h:



Classes

- class gdcm::FilenameGenerator

  FilenameGenerator.

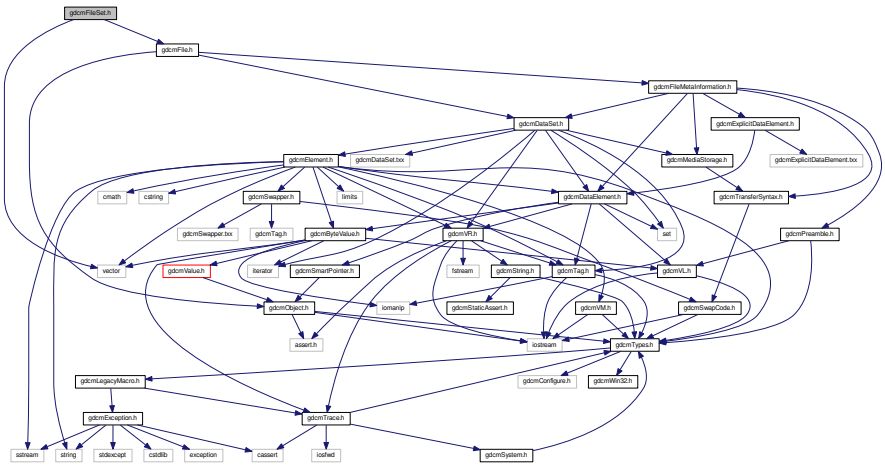
Namespaces

- gdcm

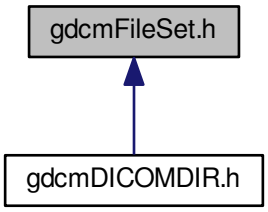
28.97 gdcmFileSet.h File Reference

```
#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

  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.

Namespaces

- gdcm

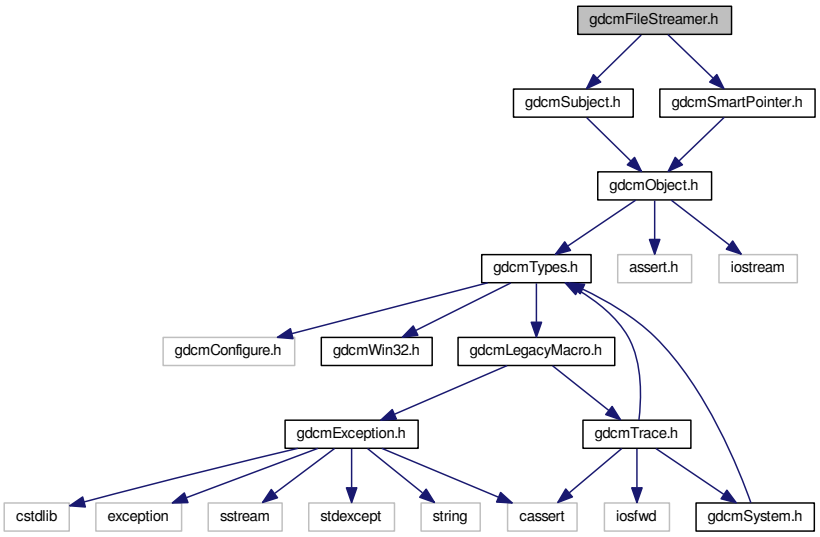
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)

28.98 gdcmFileStreamer.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileStreamer.h:



Classes

- class gdcm::FileStreamer

  FileStreamer This class let a user create a massive DICOM DataSet from a template DICOM file, by appending chunks of data.

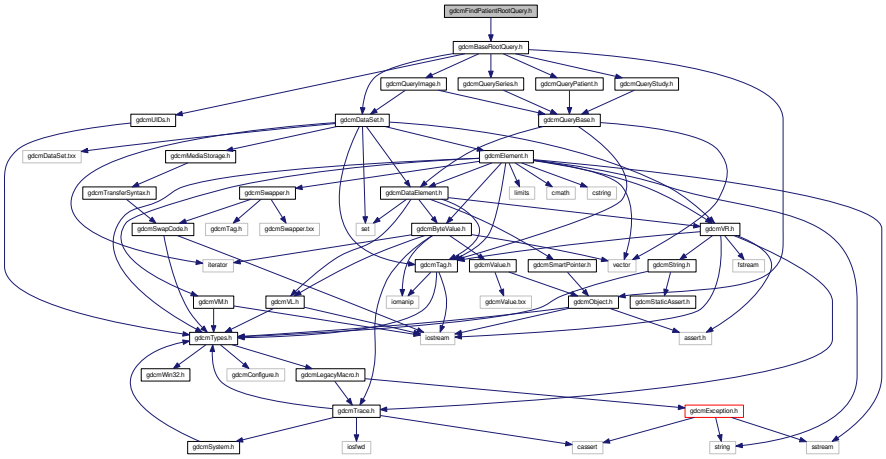
Namespaces

- gdcm

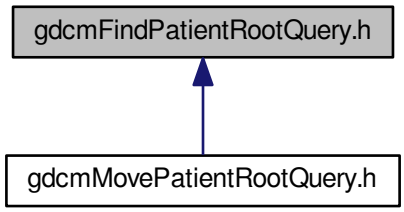
28.99 gdcmFindPatientRootQuery.h File Reference

```
#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 contains: the class which will produce a dataset for c-find with patient root.

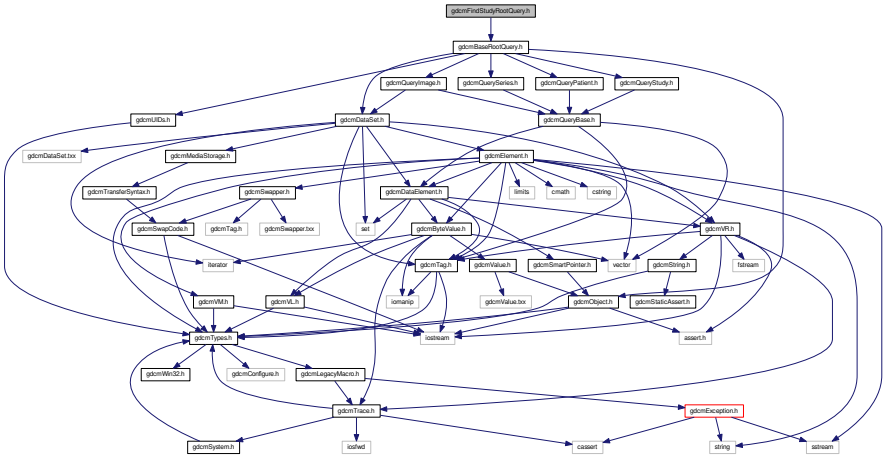
Namespaces

- gdcm

28.100 gdcmFindStudyRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmFindStudyRootQuery.h:



Classes

- class gdcm::FindStudyRootQuery

  FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.

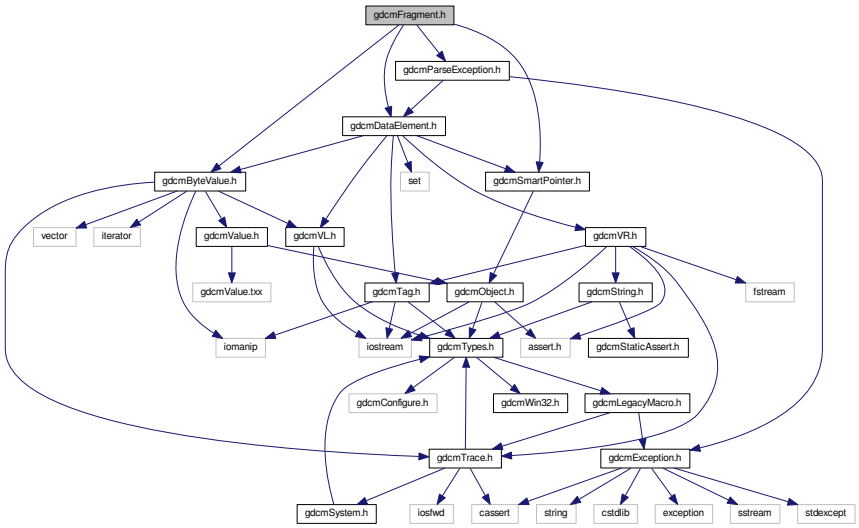
Namespaces

- gdcm

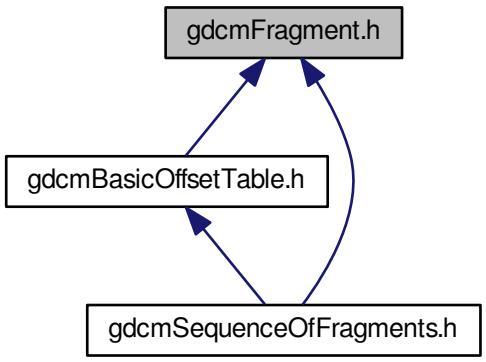
28.101 gdcmFragment.h File Reference

```
#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

- gdcm

Functions

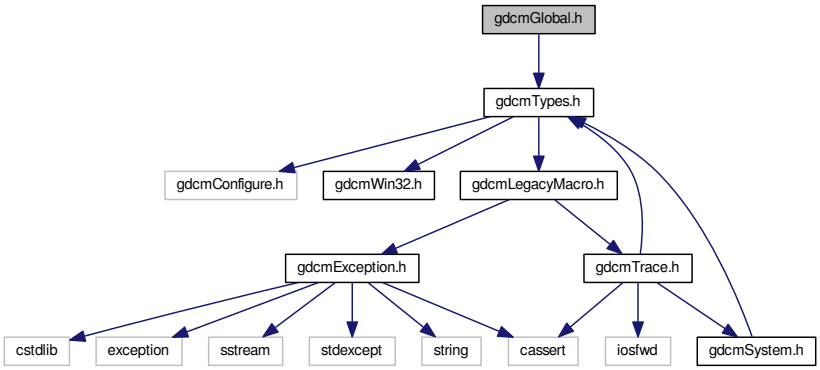
- std::ostream & gdcm::operator<< (std::ostream &os, const Fragment &val)

28.102 gdcmgendir.dox File Reference

28.103 gdcmGlobal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmGlobal.h:



Classes

- class gdcm::Global

  Global.

Namespaces

- gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Global &g)

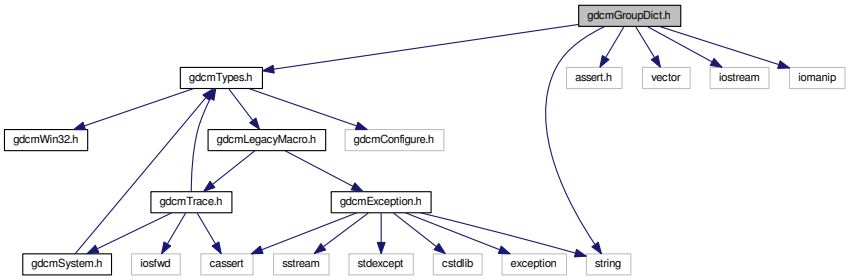
Variables

- static Global gdcm::GlobalInstance

28.104 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

- gdcm

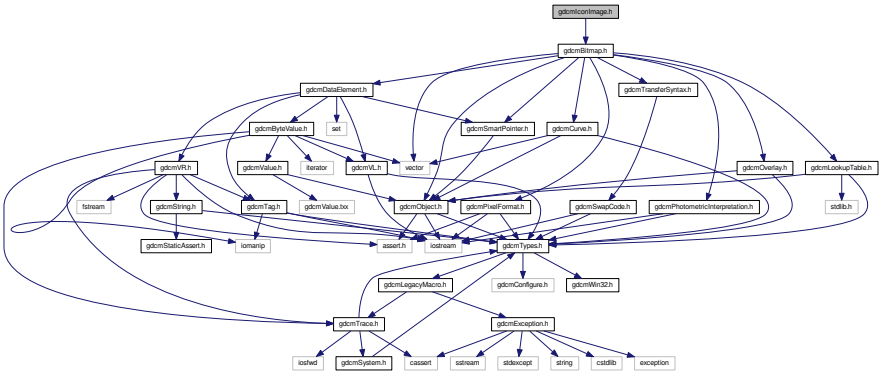
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)

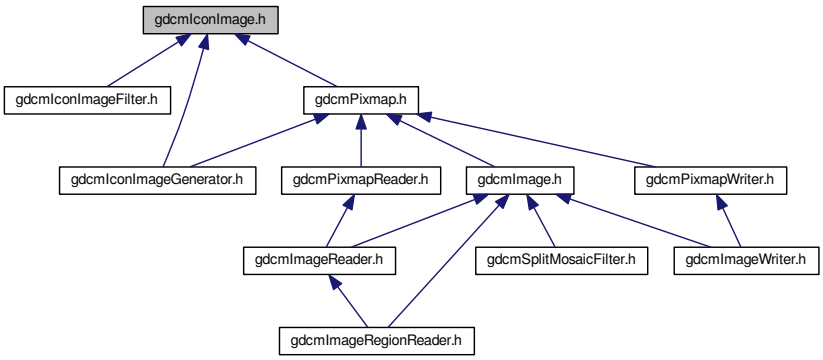
28.105 gdcmIconImage.h File Reference

```
#include "gdcmBitmap.h"
```

Include dependency graph for gdcmIconImage.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- gdcm

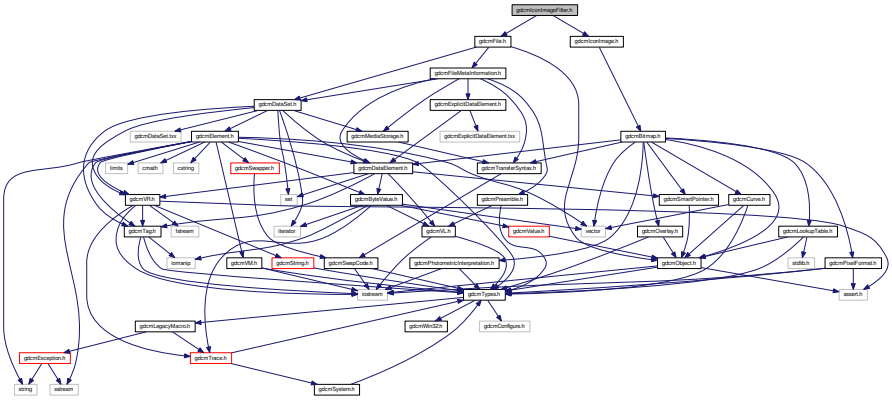
Typedefs

- typedef Bitmap gdcm::IconImage

28.106 gdcmIconImageFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageFilter.h:



Classes

- class gdcm::IconImageFilter

  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.

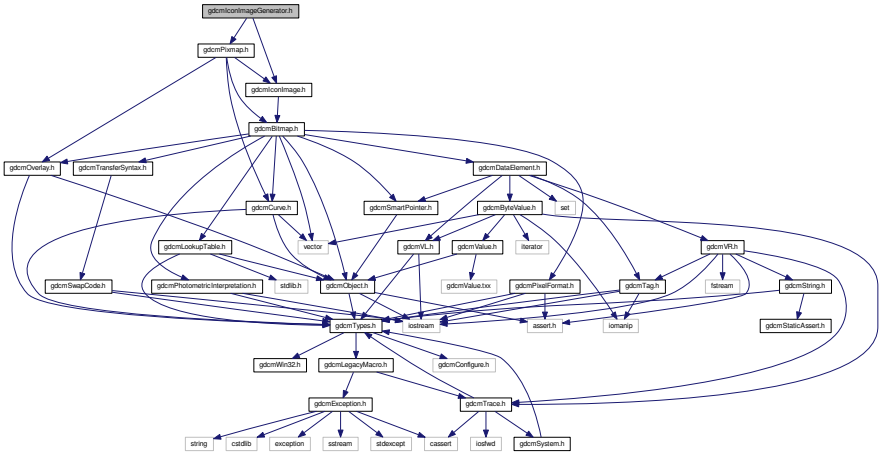
Namespaces

- gdcm

28.107 gdcmIconImageGenerator.h File Reference

```
#include "gdcmPixmap.h"
#include "gdcmIconImage.h"
```

Include dependency graph for gdcmIconImageGenerator.h:



Classes

- class gdcm::IconImageGenerator

  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:

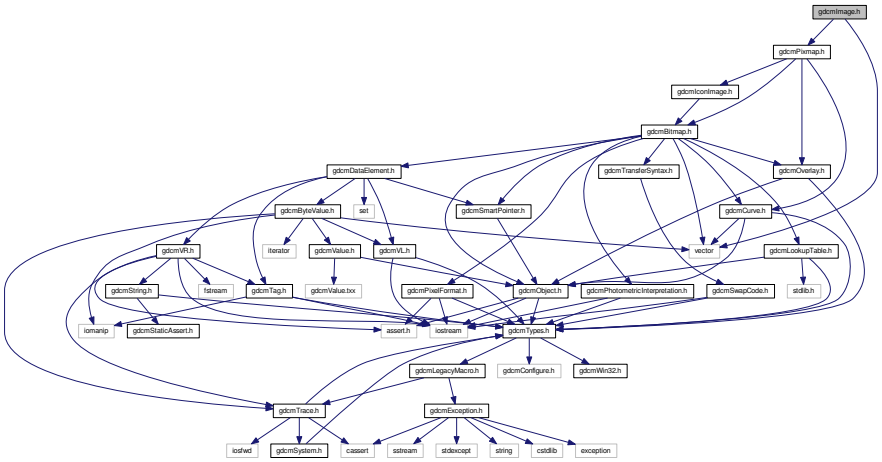
Namespaces

- gdcm

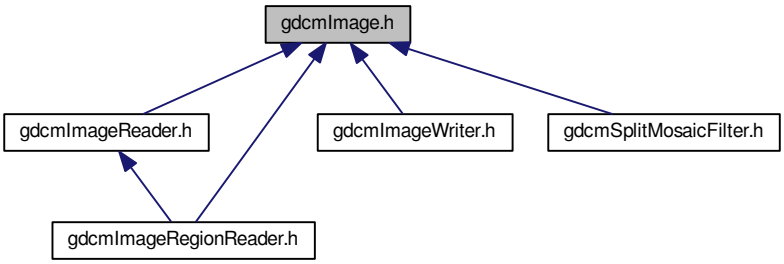
28.108 gdcmImage.h File Reference

```
#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 This is the container for an Image in the general sense. From this container you should be able to request information like:

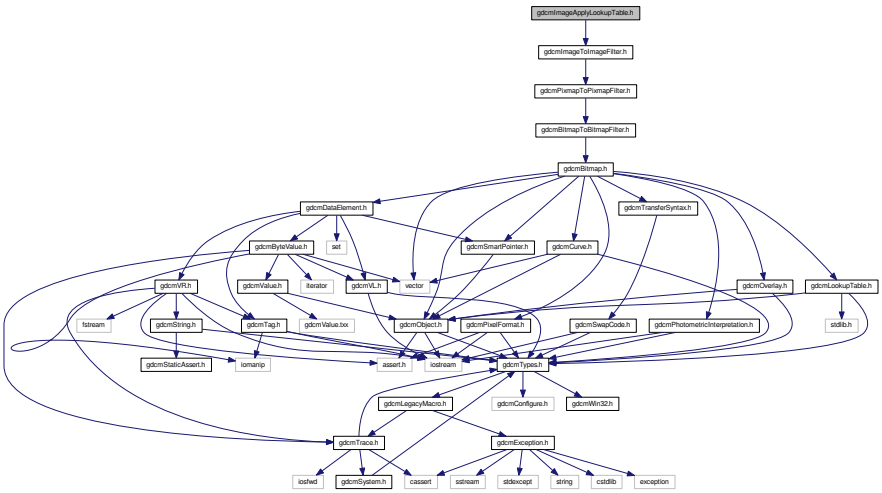
Namespaces

- gdcm

28.109 gdcmImageApplyLookupTable.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageApplyLookupTable.h:



Classes

- class gdcm::ImageApplyLookupTable

  ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation=RGB image.

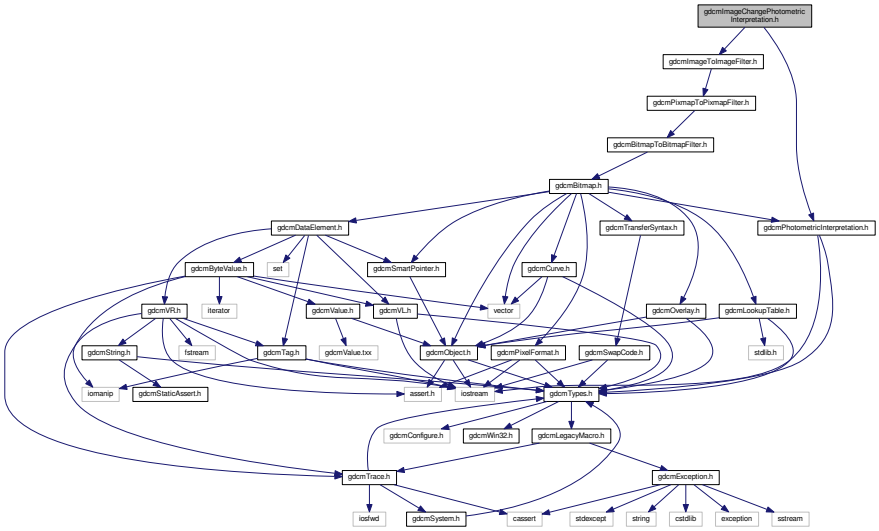
Namespaces

- gdcm

28.110 gdcmImageChangePhotometricInterpretation.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmPhotometricInterpretation.h"
```

Include dependency graph for gdcmImageChangePhotometricInterpretation.h:



Classes

- class gdcm::ImageChangePhotometricInterpretation

  ImageChangePhotometricInterpretation class Class to change the Photometric Interpetation of an input DICOM.

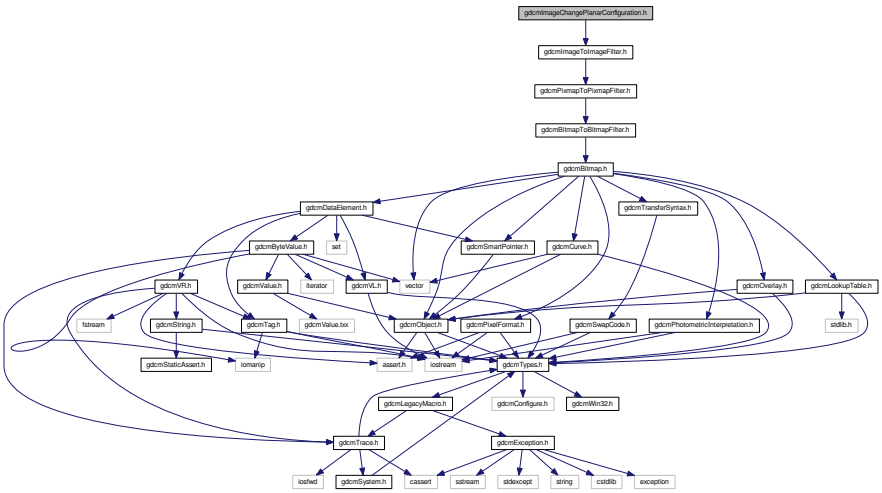
Namespaces

- gdcm

28.111 gdcmImageChangePlanarConfiguration.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageChangePlanarConfiguration.h:



Classes

- class gdcm::ImageChangePlanarConfiguration

  ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual reprensentation: PlanarConfiguration = 0.

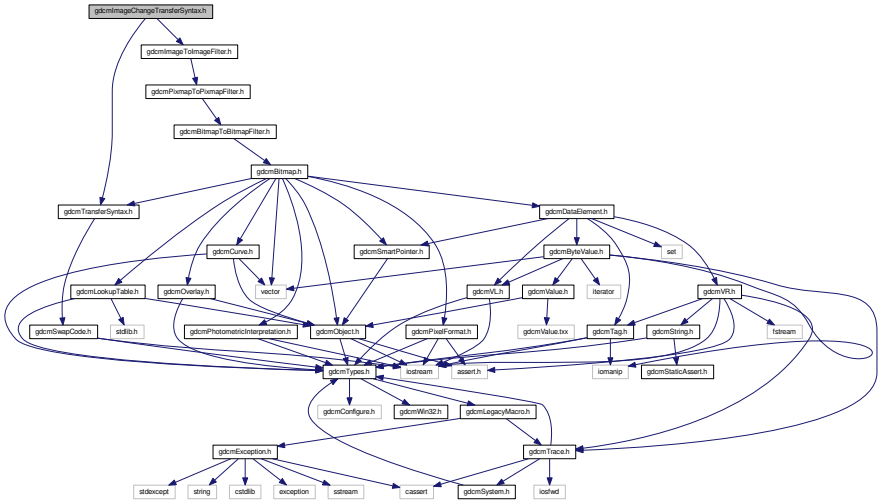
Namespaces

- gdcm

28.112 gdcmImageChangeTransferSyntax.h File Reference

```
#include "gdcmImageToImageFilter.h"
#include "gdcmTransferSyntax.h"
```

Include dependency graph for gdcmImageChangeTransferSyntax.h:



Classes

- class gdcm::ImageChangeTransferSyntax

  ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM.

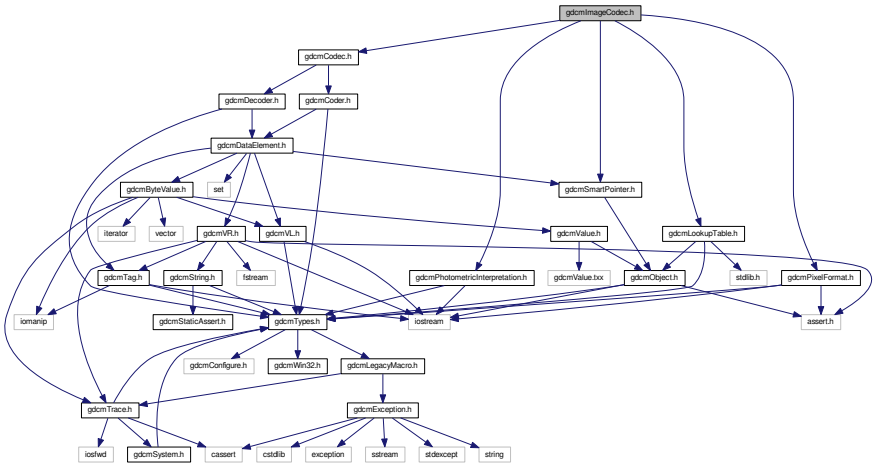
Namespaces

- gdcm

28.113 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.

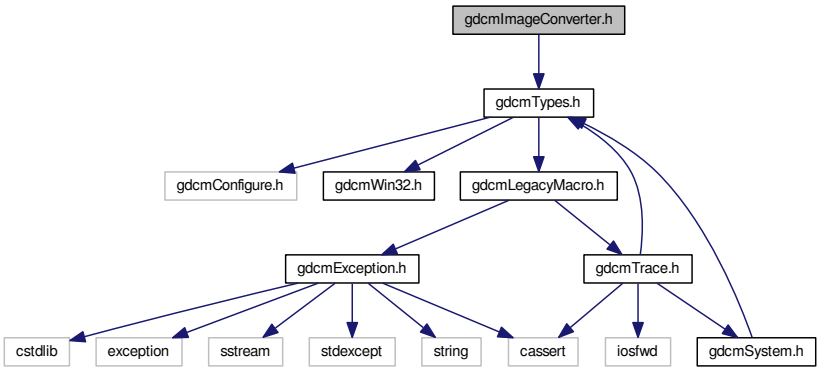
Namespaces

- gdcm

28.114 gdcmImageConverter.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImageConverter.h:



Classes

- class gdcm::ImageConverter

  Image Converter.

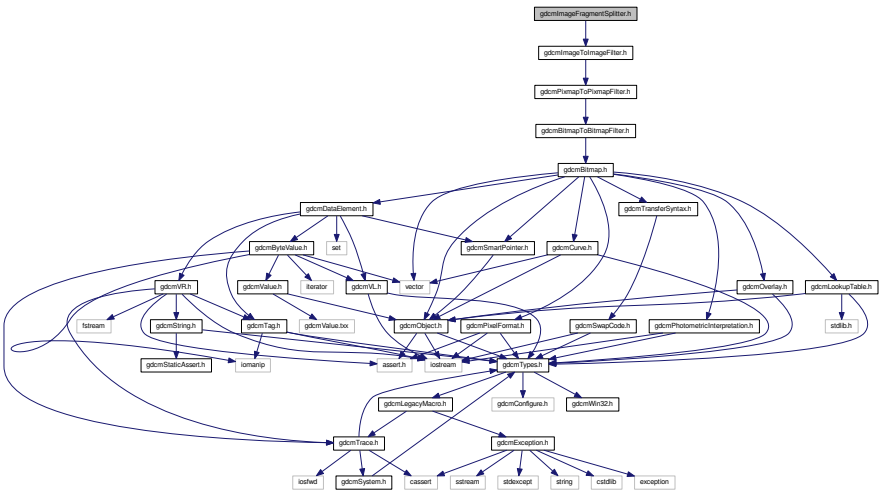
Namespaces

- gdcm

28.115 gdcmImageFragmentSplitter.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

Include dependency graph for gdcmImageFragmentSplitter.h:



Classes

- class gdcm::ImageFragmentSplitter

  ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

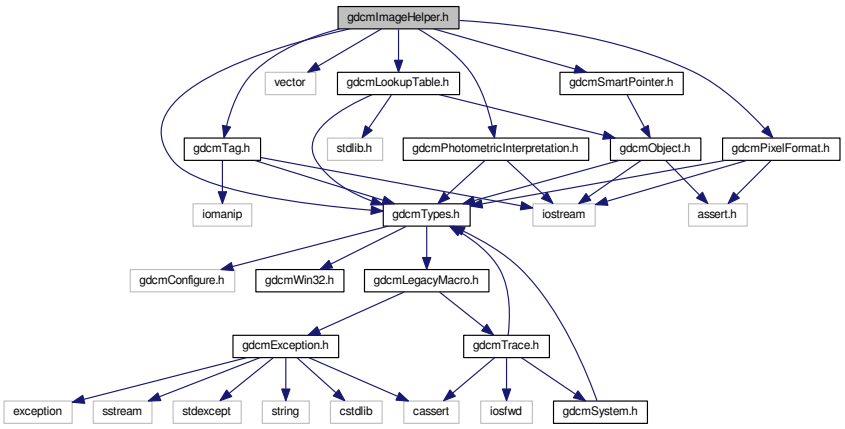
Namespaces

- gdcm

28.116 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)

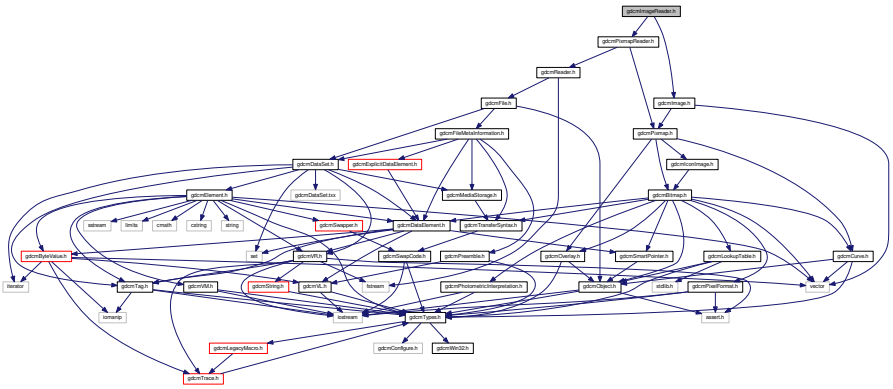
Namespaces

- gdcm

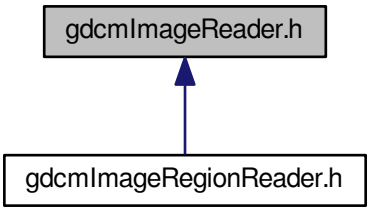
28.117 gdcmImageReader.h File Reference

```
#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

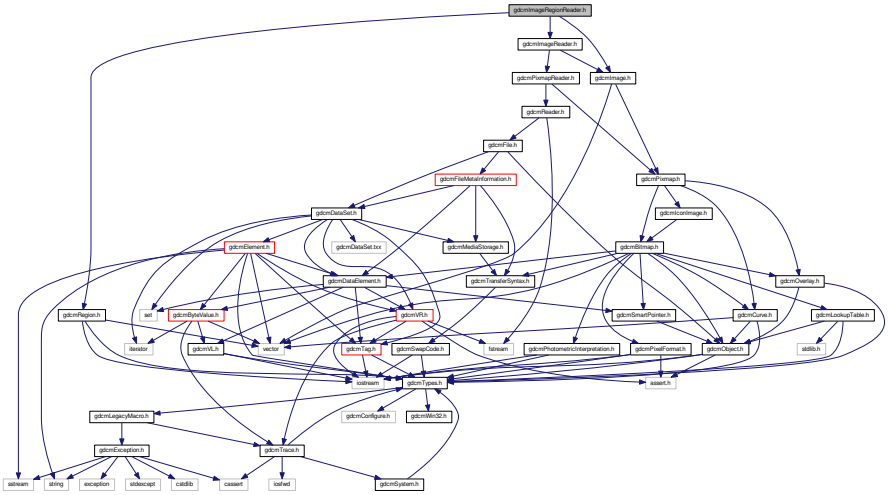
- gdcm

28.118 gdcmImageRegionReader.h File Reference

```
#include "gdcmImageReader.h"
```

```
#include "gdcmImage.h"
#include "gdcmRegion.h"
```

Include dependency graph for gdcmImageRegionReader.h:



Classes

- class gdcm::ImageRegionReader

  ImageRegionReader.

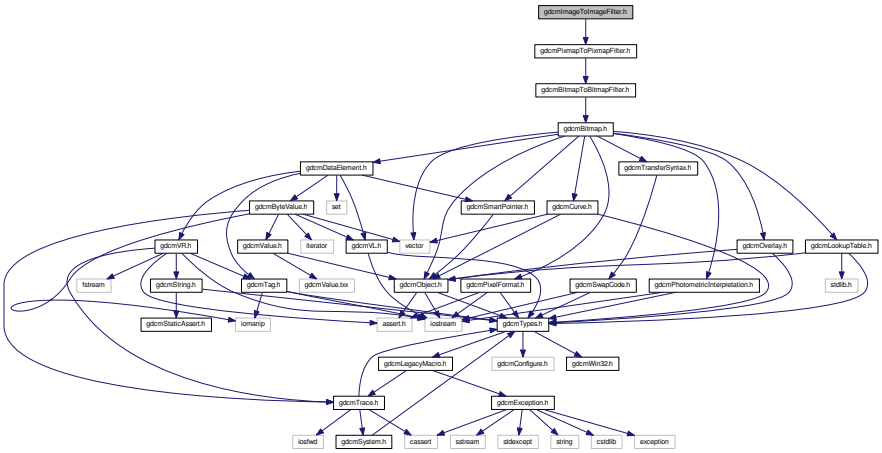
Namespaces

- gdcm

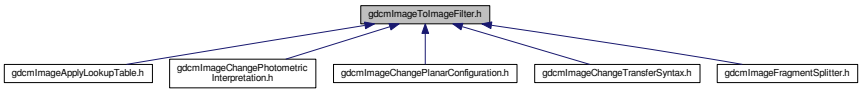
28.119 gdcmImageToImageFilter.h File Reference

```
#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 Super class for all filter taking an image and producing an output image.

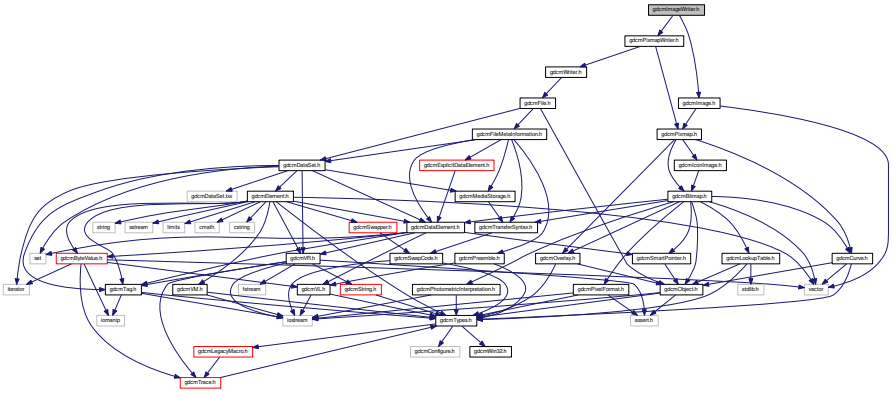
Namespaces

- gdcm

28.120 gdcmImageWriter.h File Reference

```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmImageWriter.h:



Classes

- class gdcm::ImageWriter

  ImageWriter.

Namespaces

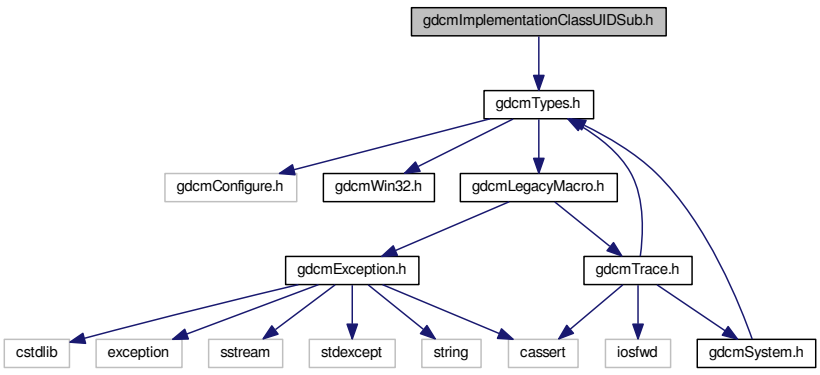
- gdcm

28.121 gdcmimg.dox File Reference

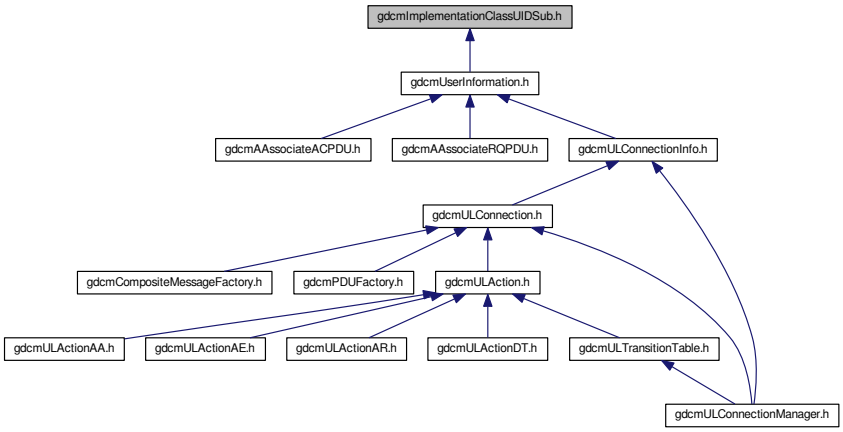
28.122 gdcmImplementationClassUIDSub.h File Reference

```
#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 PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-↩ RQ)

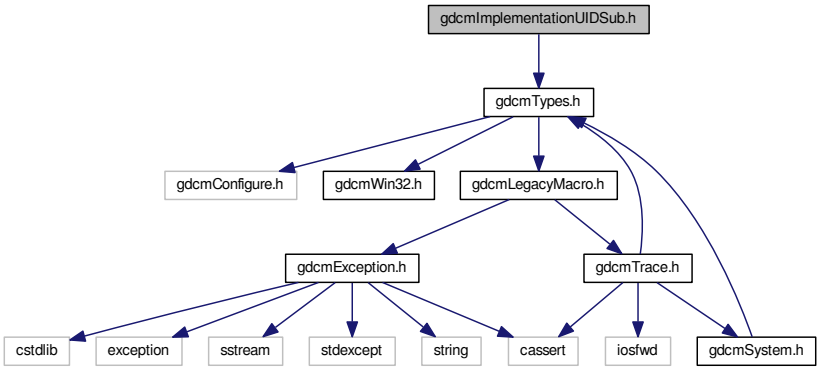
Namespaces

- gdcm
- gdcm::network

28.123 gdcmImplementationUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationUIDSub.h:



Classes

- class gdcm::network::ImplementationUIDSub

  ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

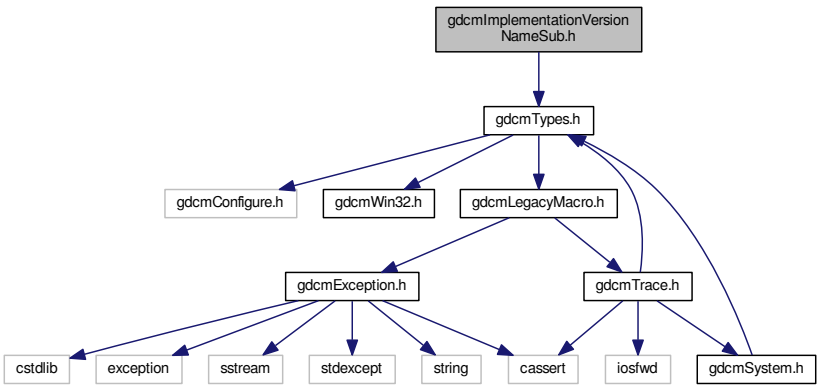
Namespaces

- gdcm
- gdcm::network

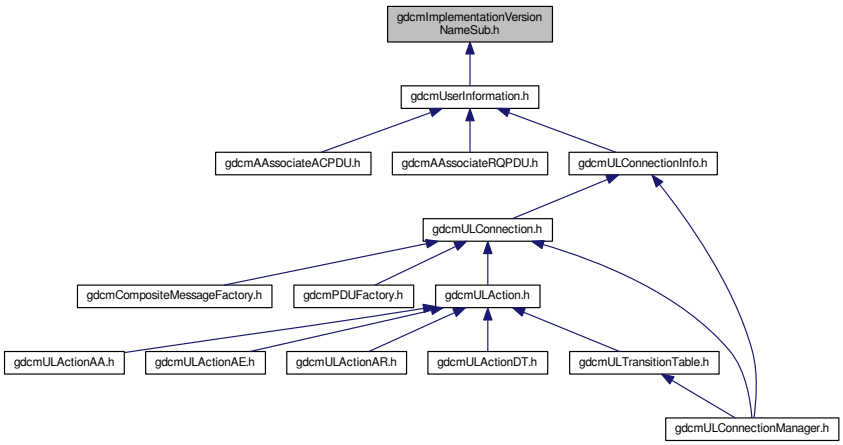
28.124 gdcmImplementationVersionNameSub.h File Reference

```
#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 Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

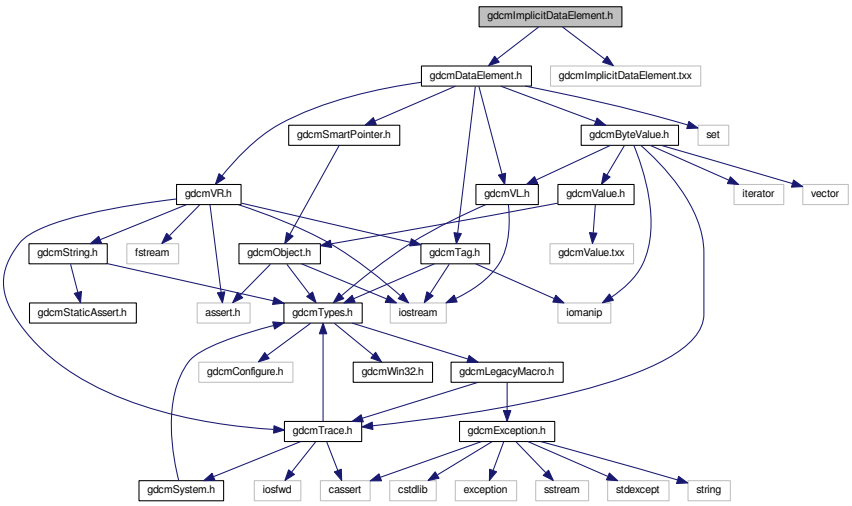
Namespaces

- gdcm
- gdcm::network

28.125 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

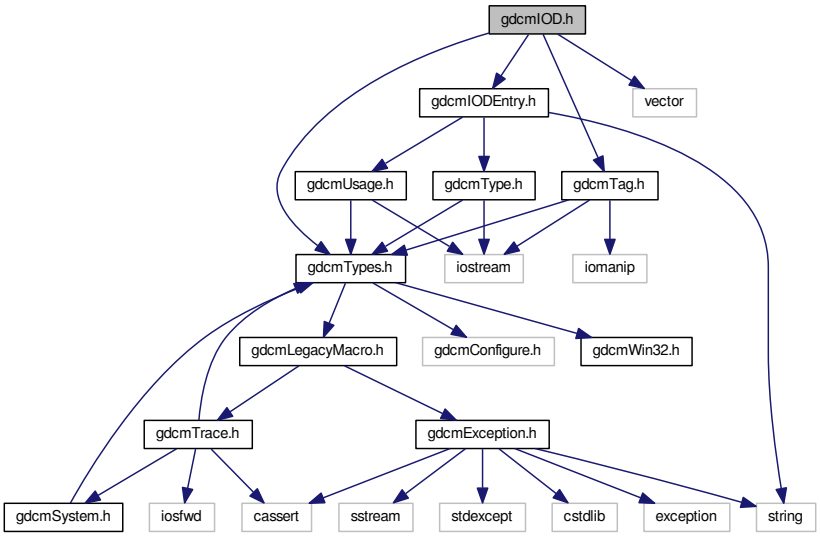
- gdcm

28.126 gdcminfo.dox File Reference

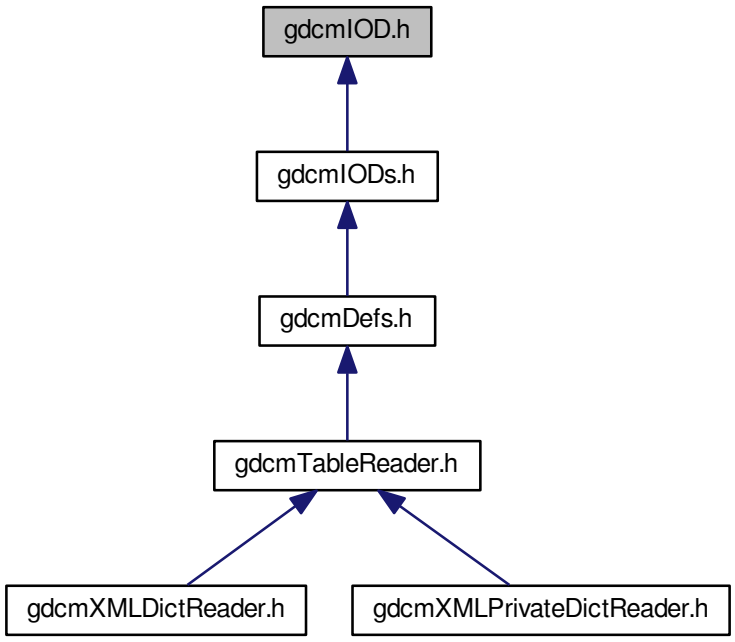
28.127 gdcmIOD.h File Reference

```
#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

- gdcm

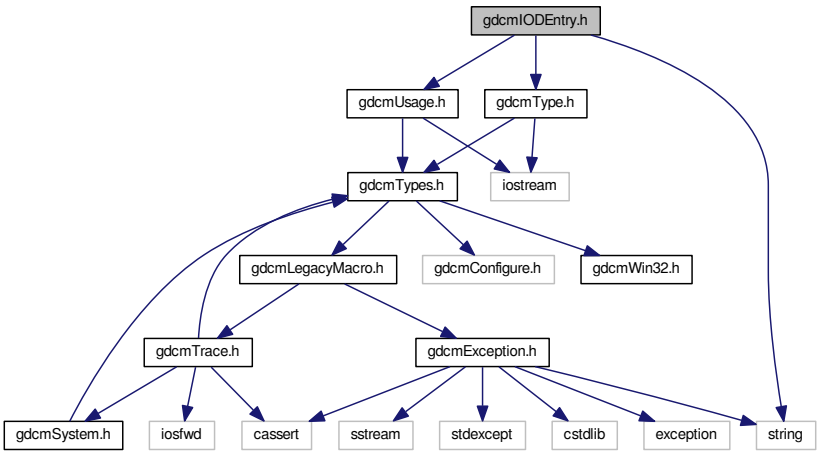
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)

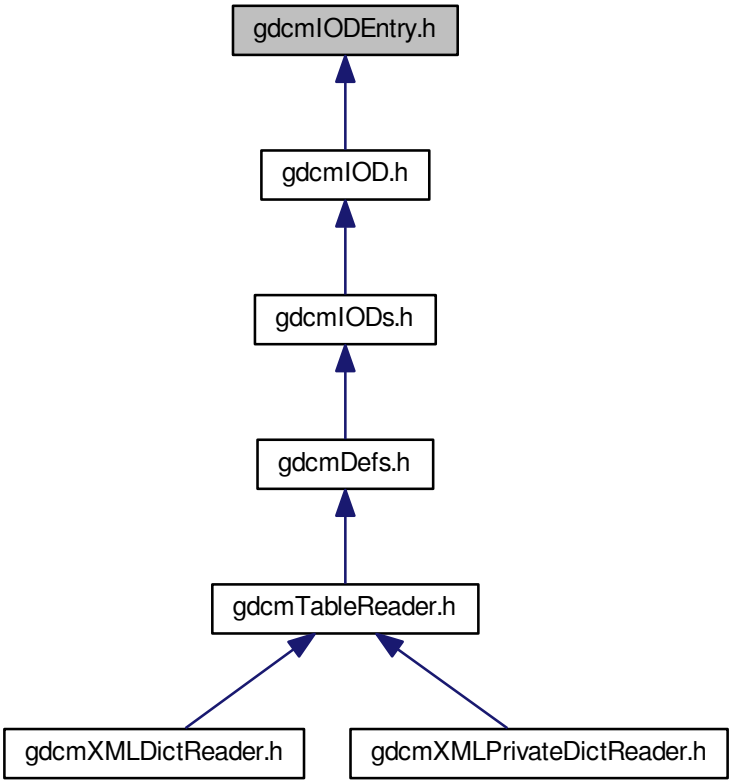
28.128 gdcmIODEntry.h File Reference

```
#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

- gdcm

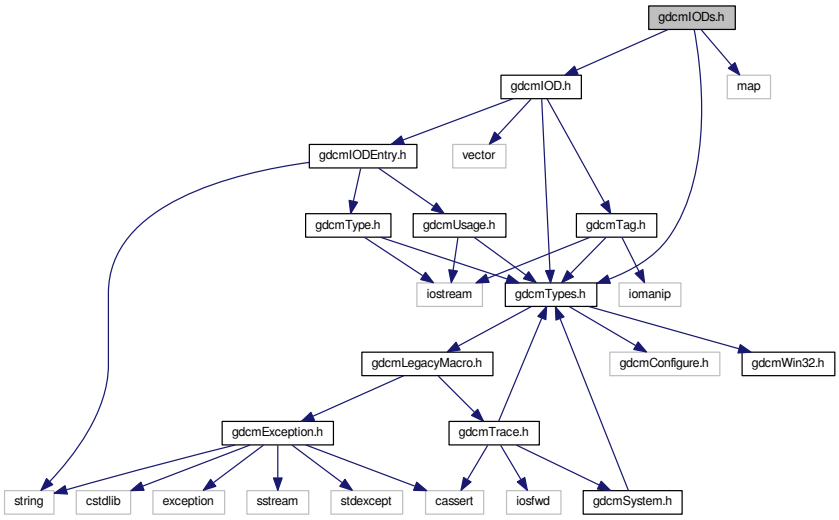
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)

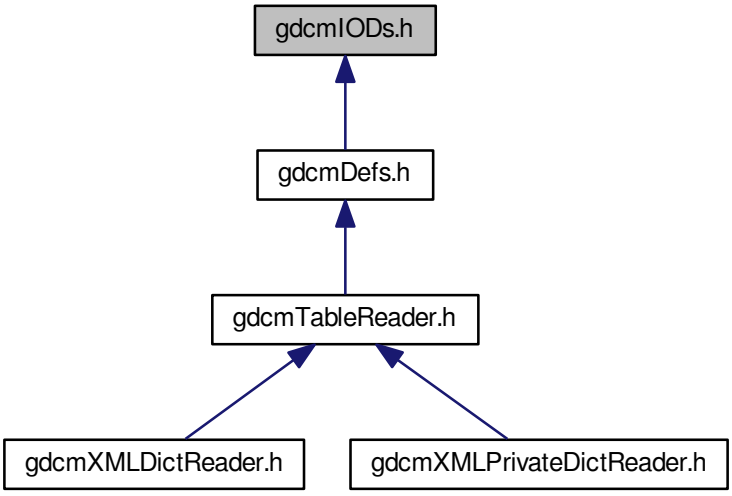
28.129 gdcmIODs.h File Reference

```
#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

- gdcm

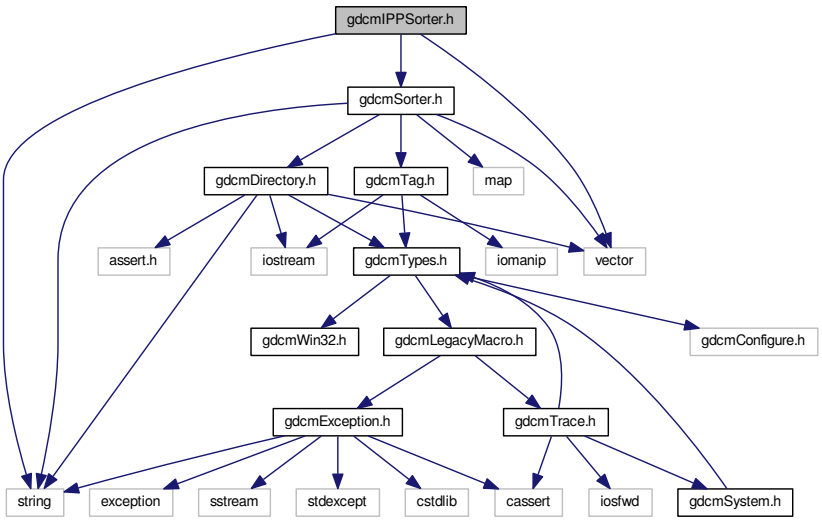
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)

28.130 gdcmIPPSorter.h File Reference

```
#include "gdcmSorter.h"
#include <vector>
#include <string>
```

Include dependency graph for gdcmIPPSorter.h:



Classes

- class gdcm::IPPSorter

  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.

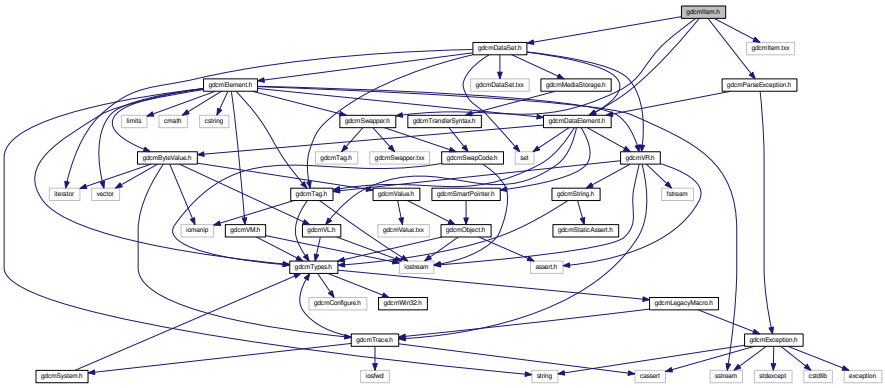
Namespaces

- gdcm

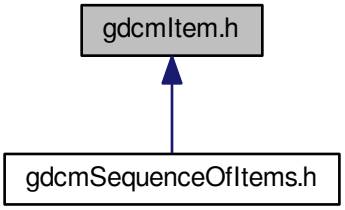
28.131 gdcmItem.h File Reference

```
#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 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 Standart 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.

Namespaces

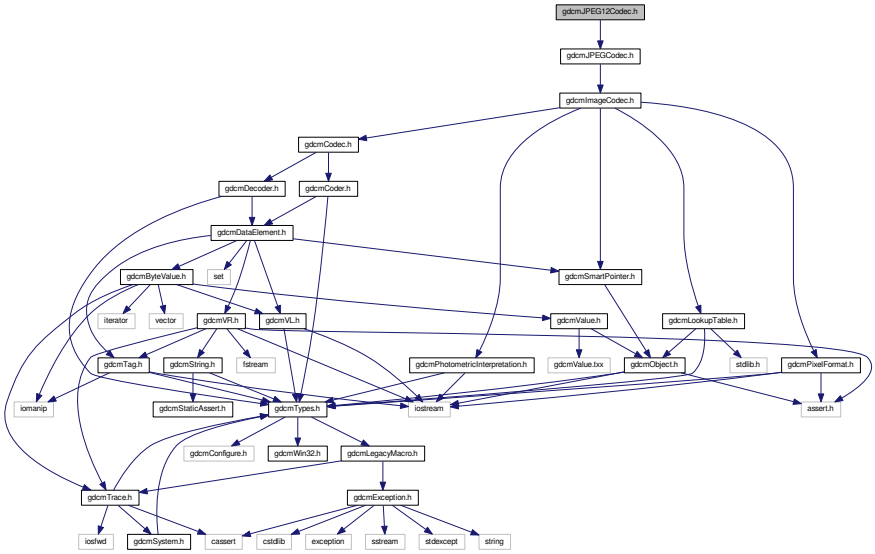
- gdcm

Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)

28.132 gdcmJPEG12Codec.h File Reference

`#include "gdcmJPEGCodec.h"`
Include dependency graph for gdcmJPEG12Codec.h:



Classes

- class gdcm::JPEG12Codec

  Class to do JPEG 12bits (lossy & lossless)

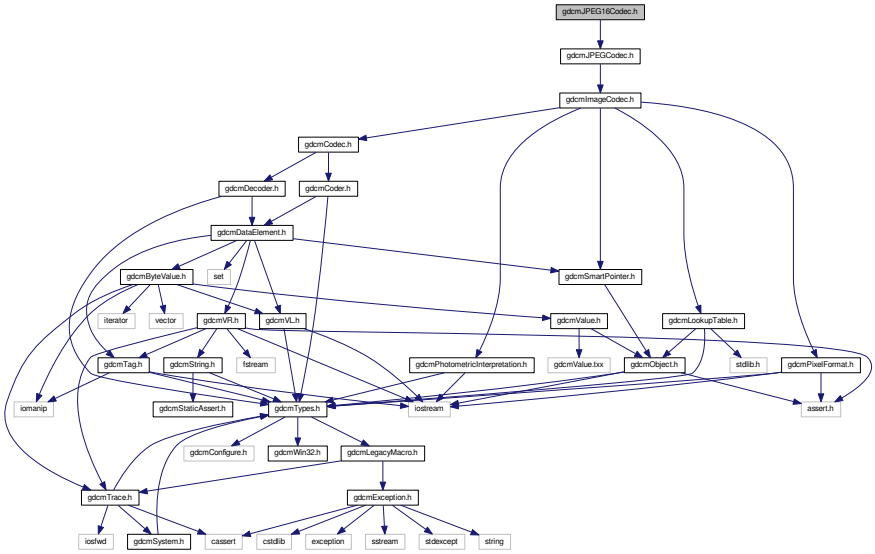
Namespaces

- gdcm

28.133 gdcmJPEG16Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG16Codec.h:



Classes

- class gdcm::JPEG16Codec

  Class to do JPEG 16bits (lossless)

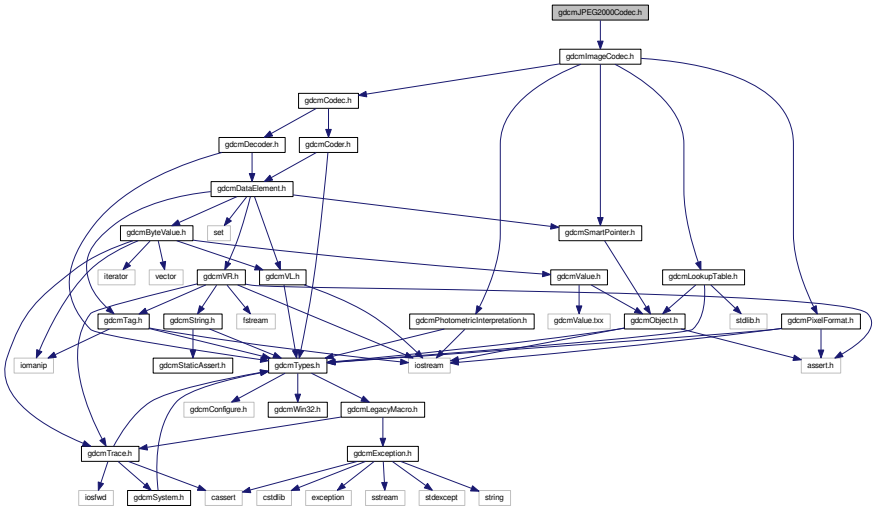
Namespaces

- gdcm

28.134 gdcmJPEG2000Codec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmJPEG2000Codec.h:



Classes

- class gdcm::JPEG2000Codec

  Class to do JPEG 2000.

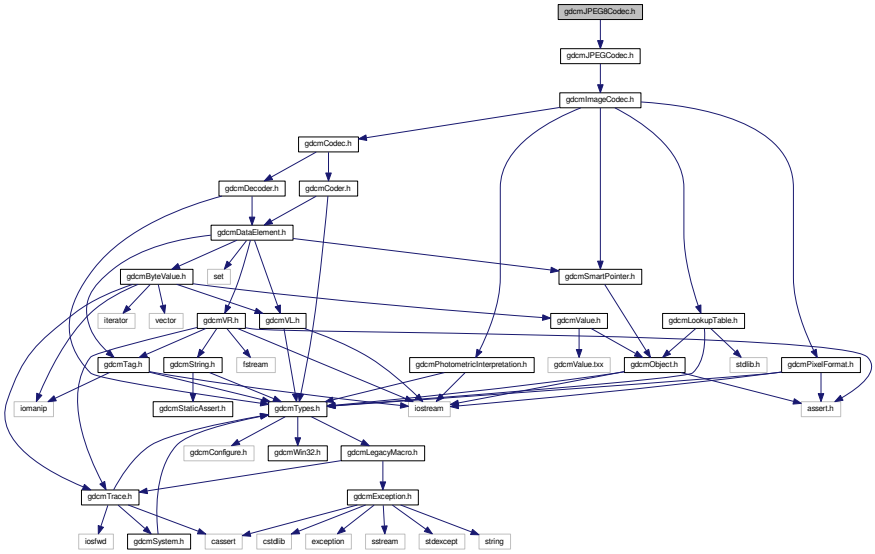
Namespaces

- gdcm

28.135 gdcmJPEG8Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG8Codec.h:



Classes

- class gdcm::JPEG8Codec

  Class to do JPEG 8bits (lossy & lossless)

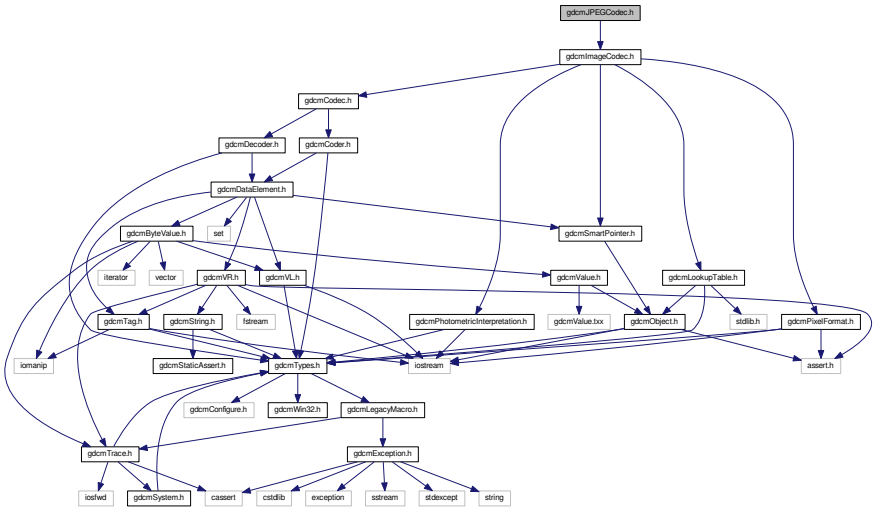
Namespaces

- gdcm

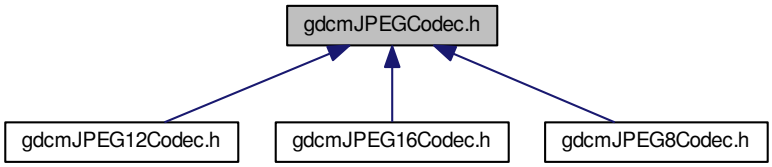
28.136 gdcmJPEGCodec.h File Reference

```
#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 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.

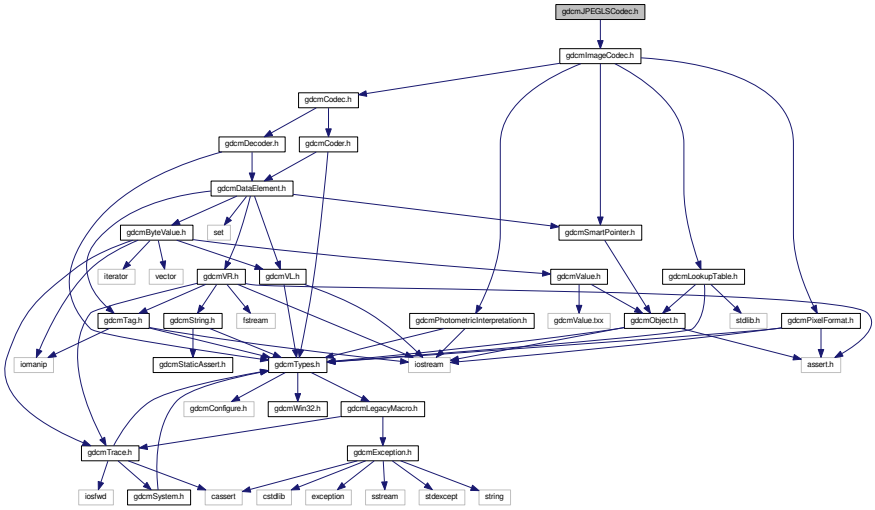
Namespaces

- gdcm

28.137 gdcmJPEGLSCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmJPEGLSCodec.h:



Classes

- class gdcm::JPEGLSCodec

  JPEG-LS.

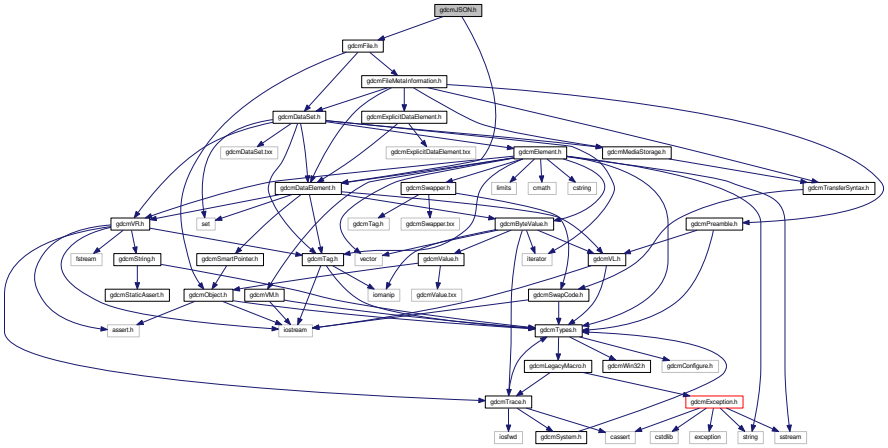
Namespaces

- gdcm

28.138 gdcmJSON.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmJSON.h:



Classes

- class gdcm::JSON

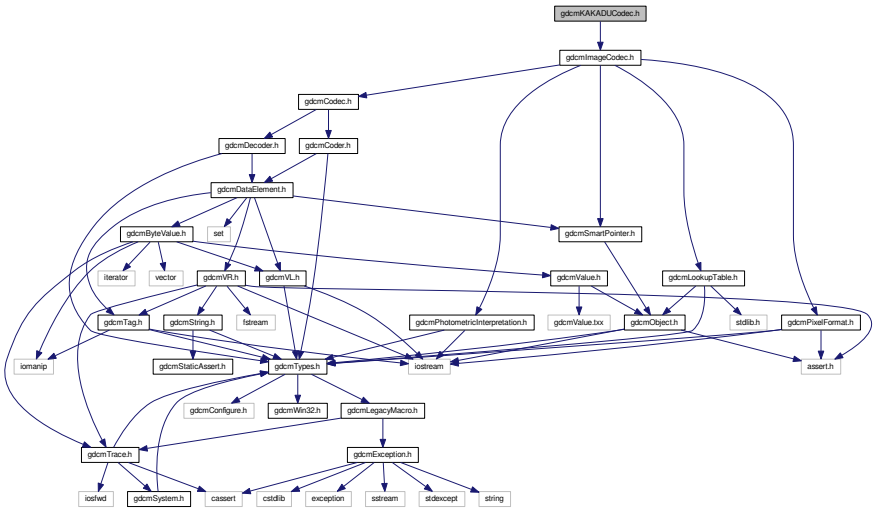
Namespaces

- gdcm

28.139 gdcmKAKADUCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmKAKADUCodec.h:



Classes

- class gdcm::KAKADUCodec
  KAKADUCodec.

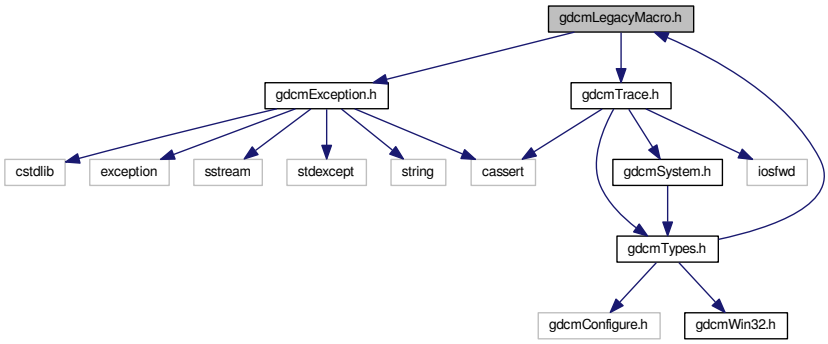
Namespaces

- gdcm

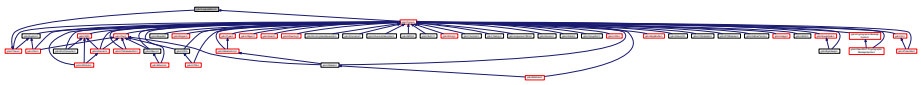
28.140 gdcmLegacyMacro.h File Reference

```
#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) method;
- #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.")

28.140.1 Macro Definition Documentation

28.140.1.1 #define GDCM_LEGACY(*method*) method;

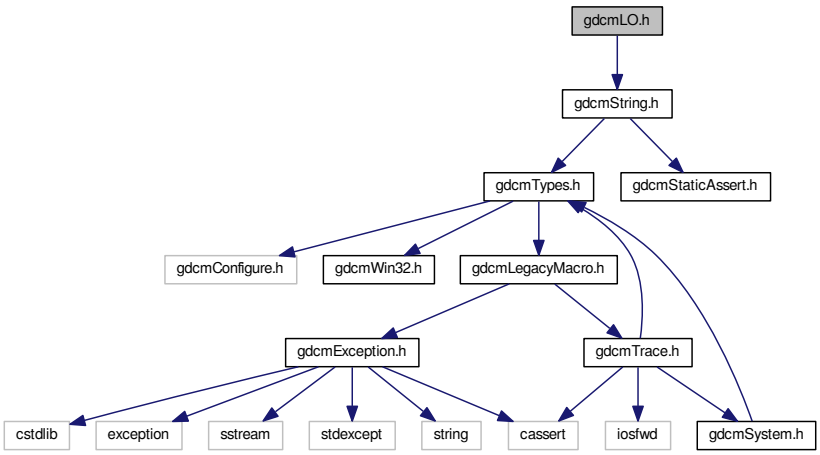
28.140.1.2 #define GDCM_LEGACY_BODY(*method, version*) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")

28.140.1.3 #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.")

28.141 gdcmLO.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmLO.h:



Classes

- class gdcm::LO

  LO.

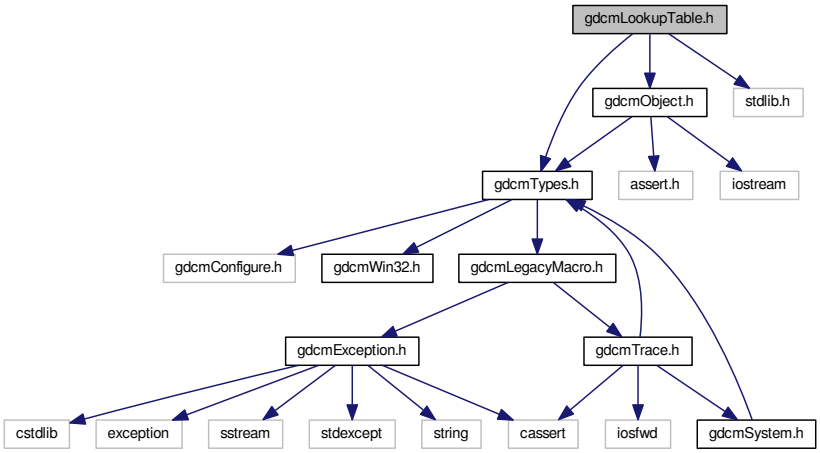
Namespaces

- gdcm

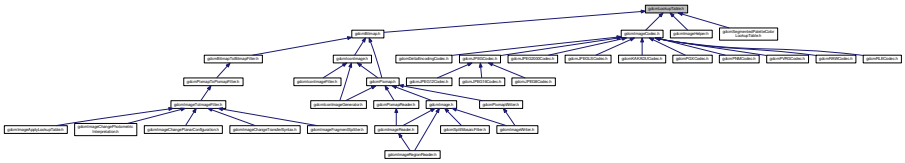
28.142 gdcmLookupTable.h File Reference

```
#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.

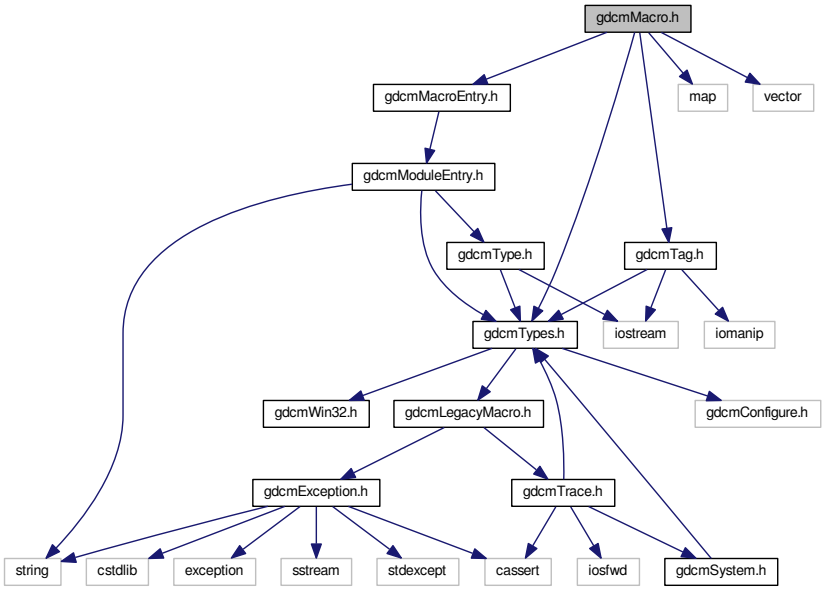
Namespaces

- gdcm

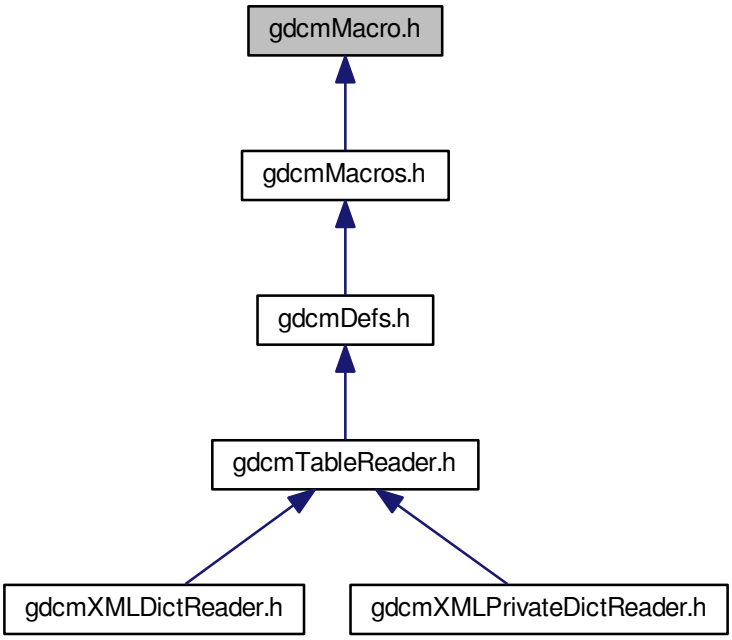
28.143 gdcmMacro.h File Reference

```
#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

- gdcm

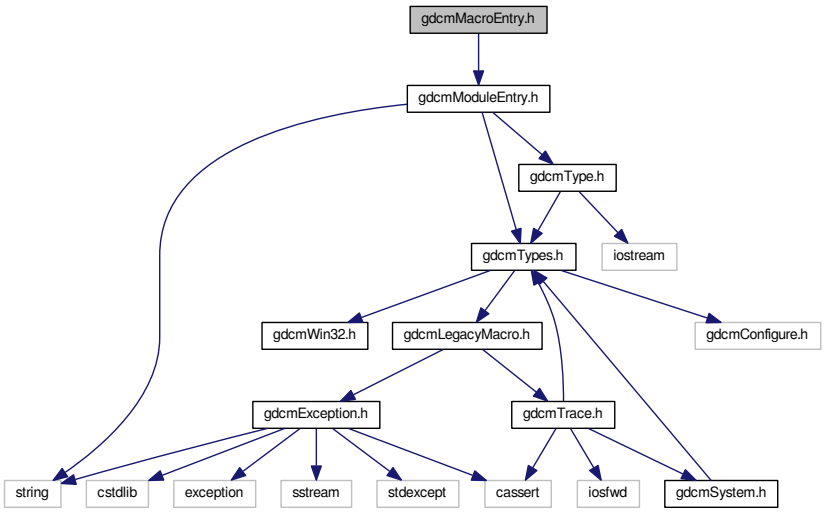
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)

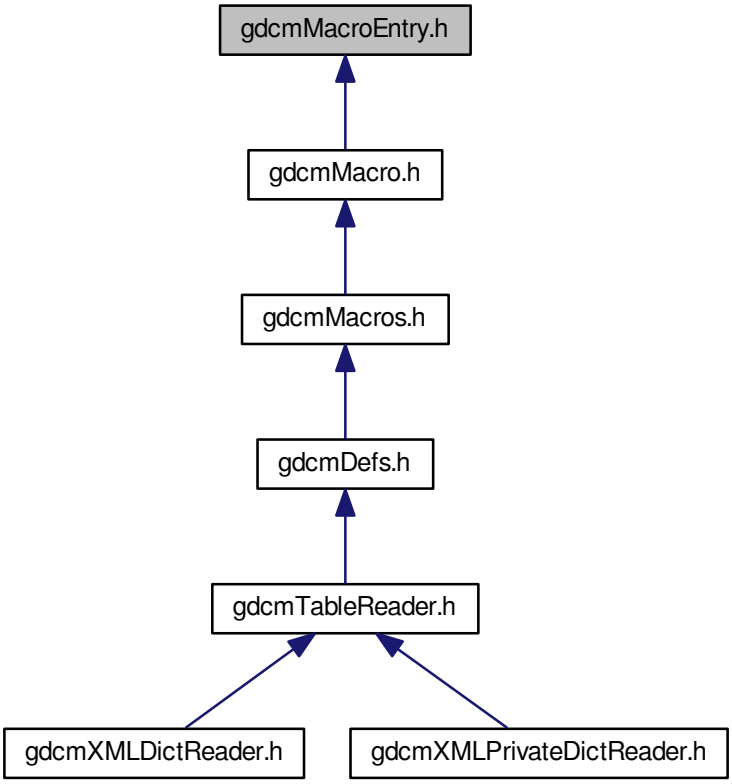
28.144 gdcmMacroEntry.h File Reference

```
#include "gdcmModuleEntry.h"
```

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define GDCMMACROENTRY_H

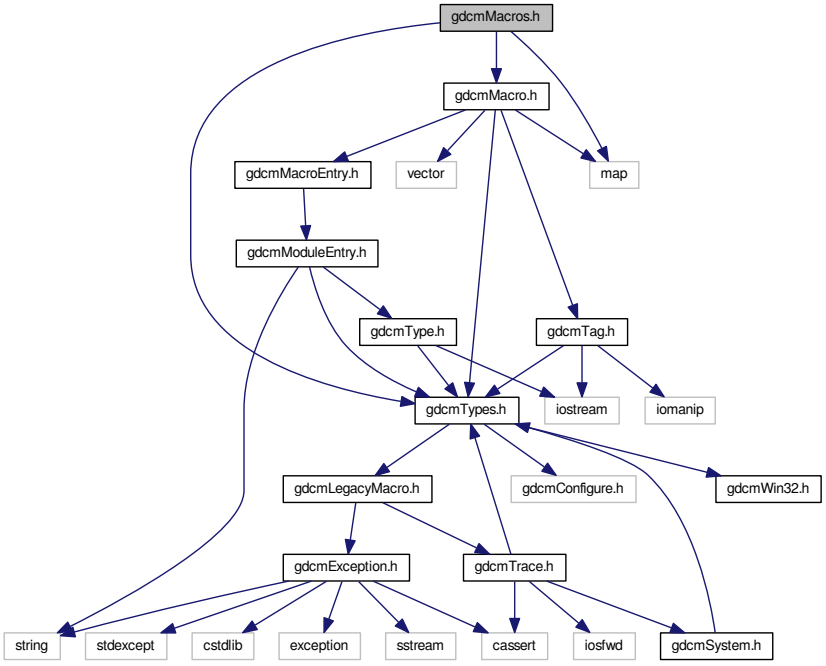
28.144.1 Macro Definition Documentation

28.144.1.1 #define GDCMMACROENTRY_H

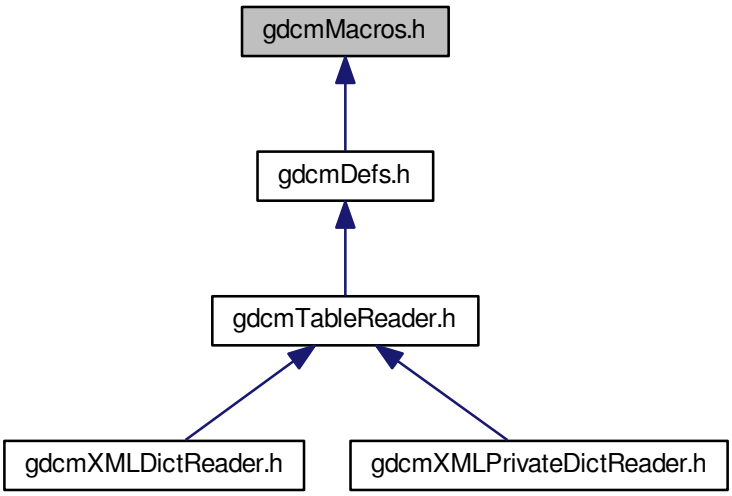
28.145 gdcmMacros.h File Reference

```
#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

- gdcm

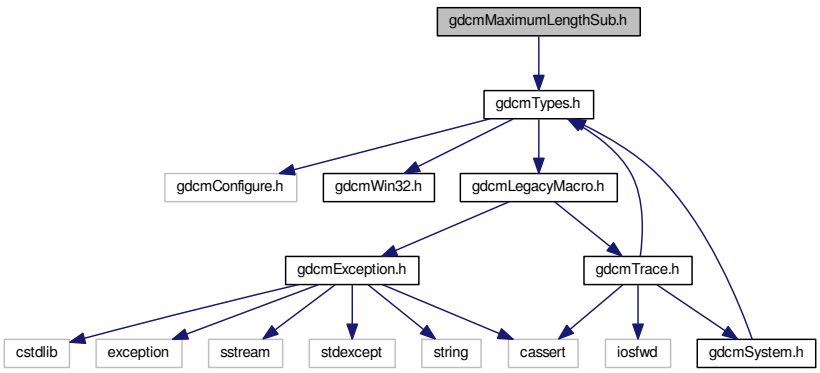
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)

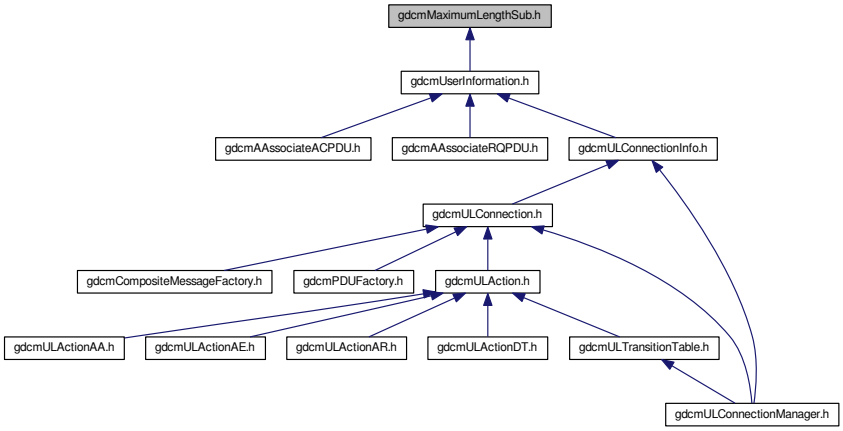
28.146 gdcmMaximumLengthSub.h File Reference

```
#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 Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

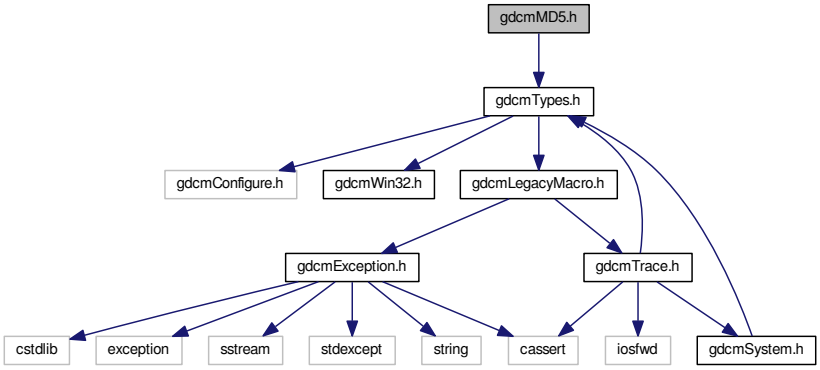
Namespaces

- gdcm
- gdcm::network

28.147 gdcmMD5.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmMD5.h:



Classes

- class gdcm::MD5

  Class for MD5.

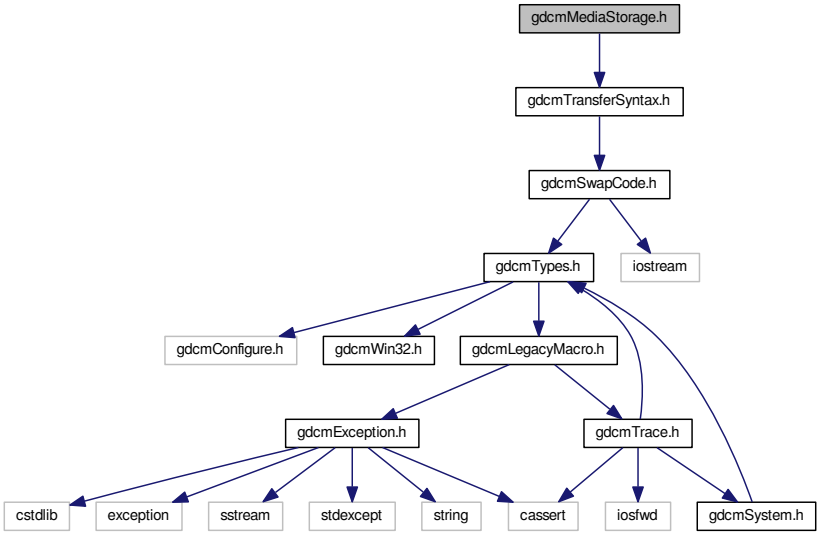
Namespaces

- gdcm

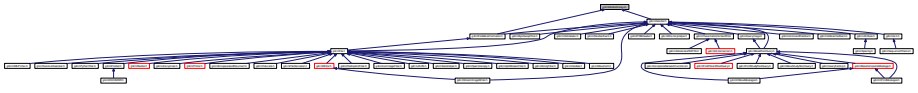
28.148 gdcmMediaStorage.h File Reference

```
#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

- gdcm

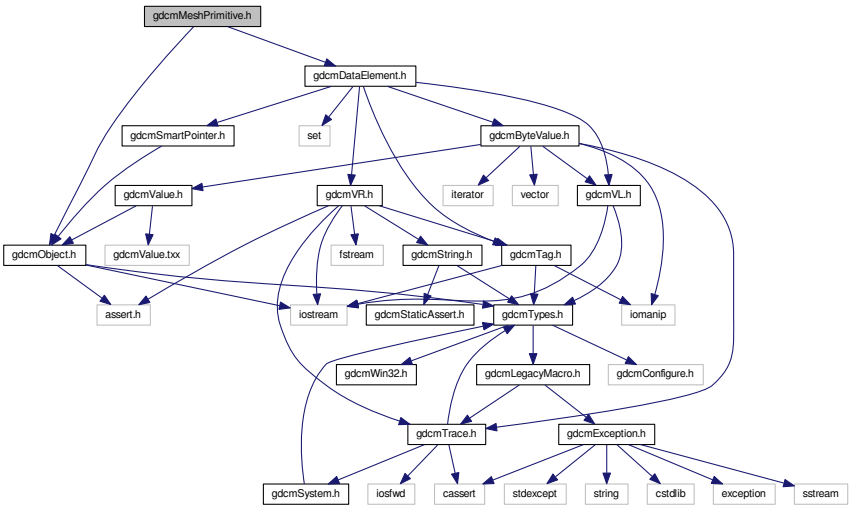
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)

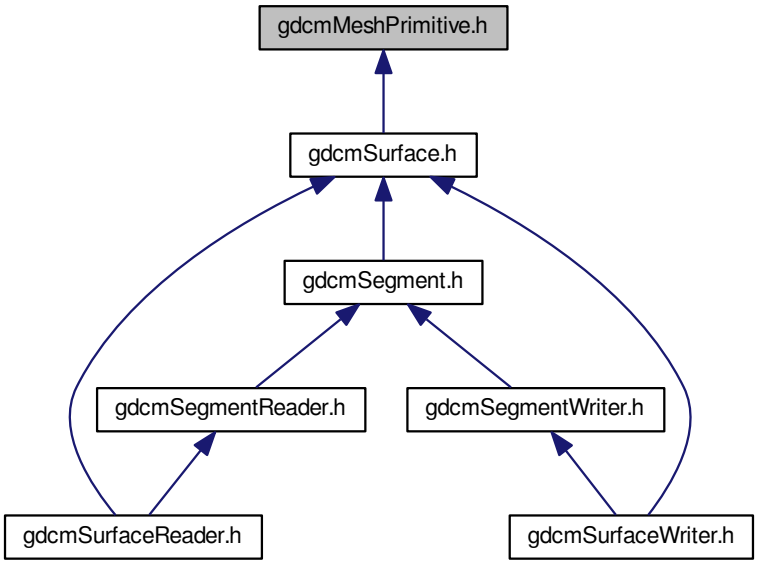
28.149 gdcmMeshPrimitive.h File Reference

```
#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. It is designed from surface mesh primitives macro.

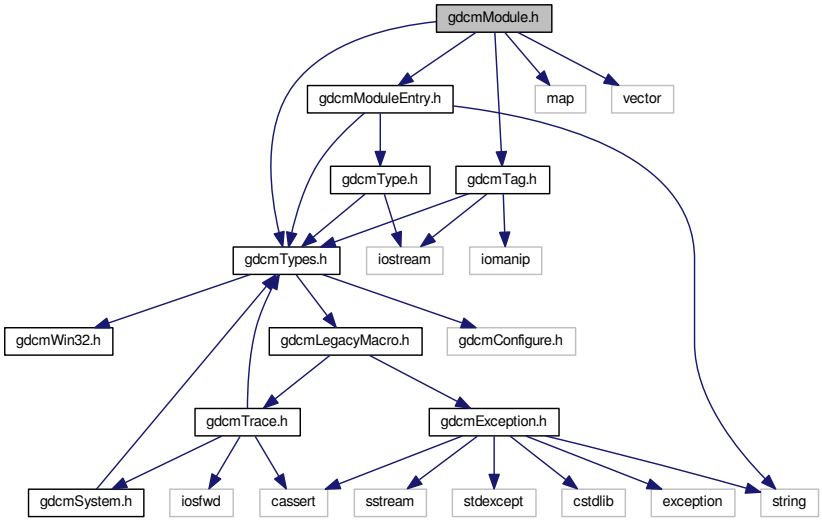
Namespaces

- gdcm

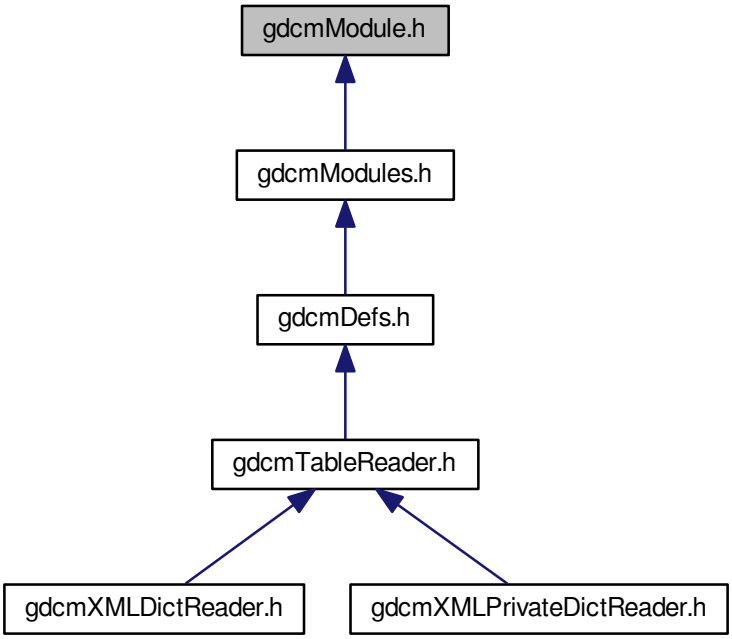
28.150 gdcmModule.h File Reference

```
#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

- gdcm

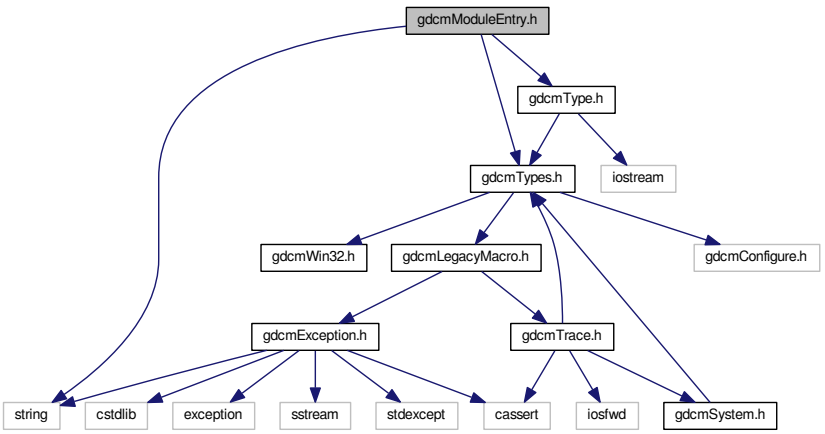
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)

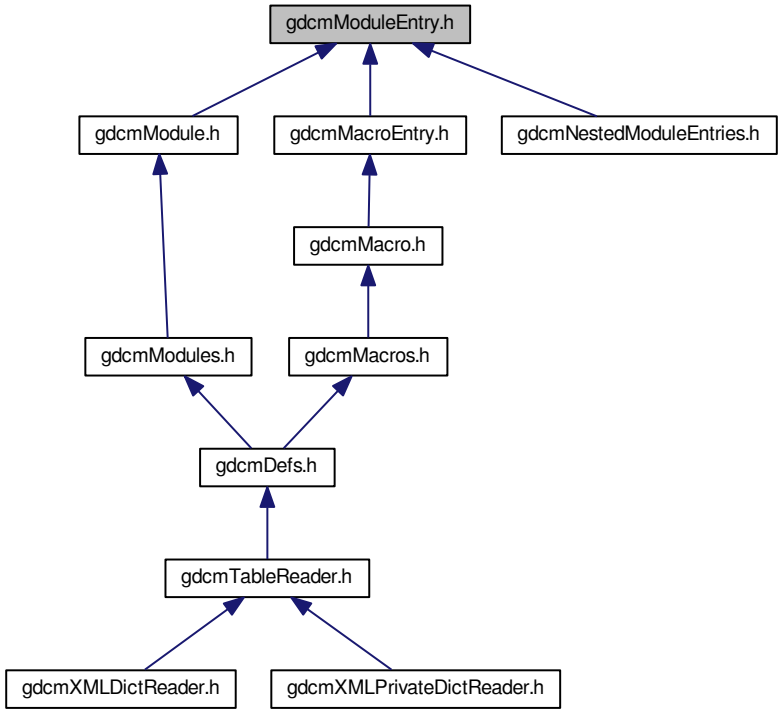
28.151 gdcmModuleEntry.h File Reference

```
#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

- gdcm

Typedefs

- typedef ModuleEntry gdcm::MacroEntry

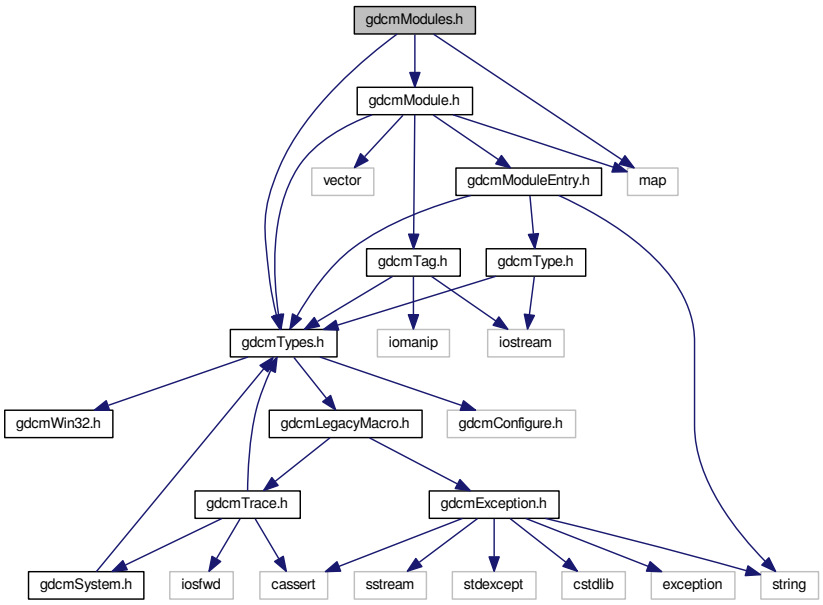
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const ModuleEntry &_val)

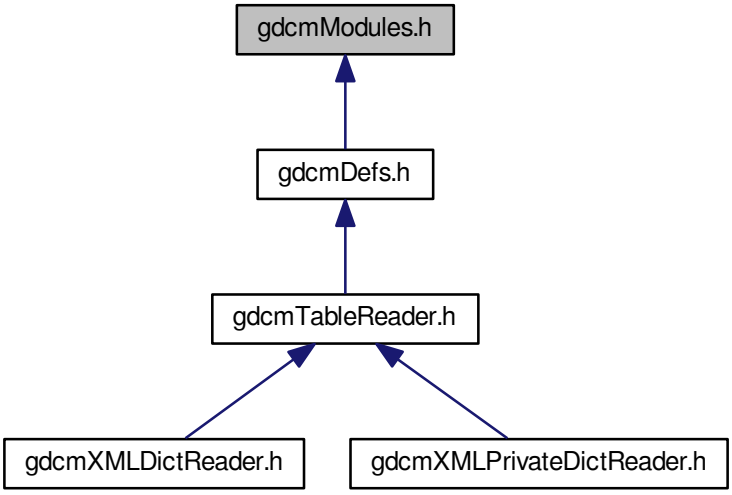
28.152 gdcmModules.h File Reference

```
#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

- gdcm

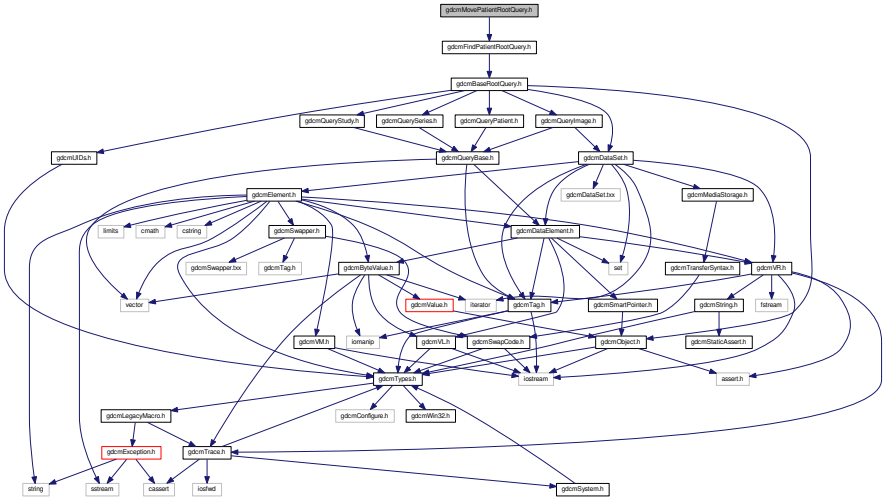
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)

28.153 gdcmMovePatientRootQuery.h File Reference

```
#include "gdcmFindPatientRootQuery.h"
```

Include dependency graph for gdcmMovePatientRootQuery.h:



Classes

- class gdcm::MovePatientRootQuery

  MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root.

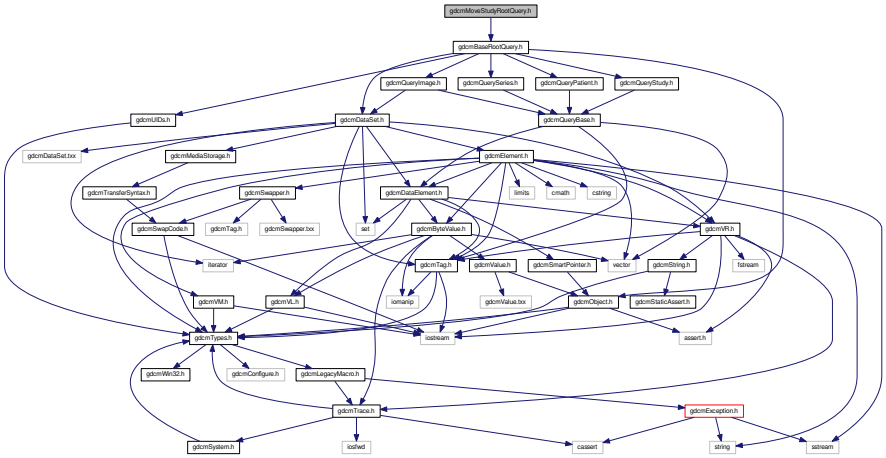
Namespaces

- gdcm

28.154 gdcmMoveStudyRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmMoveStudyRootQuery.h:



Classes

- class gdcm::MoveStudyRootQuery

  MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root.

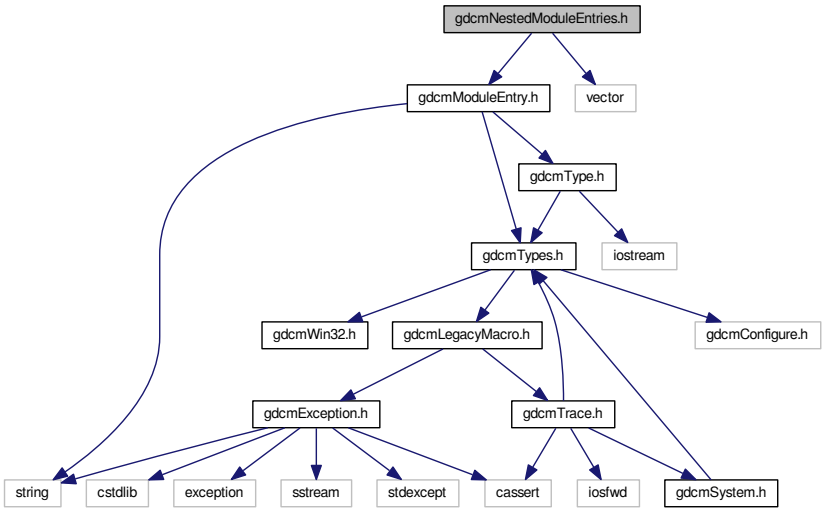
Namespaces

- gdcm

28.155 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

- gdcm

Typedefs

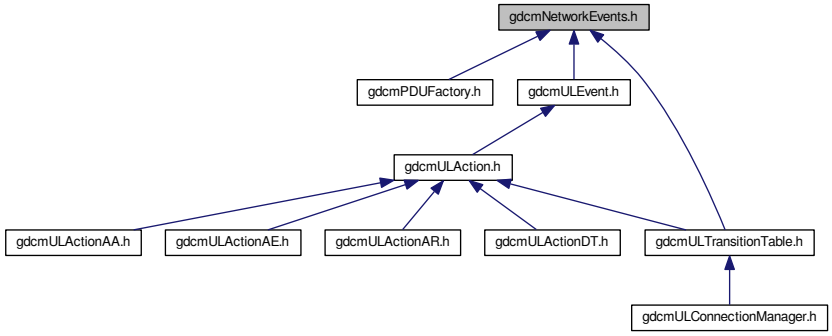
- typedef NestedModuleEntries gdcm::NestedMacroEntries

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const NestedModuleEntries &_val)

28.156 gdcmNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- gdcm
- gdcm::network

Enumerations

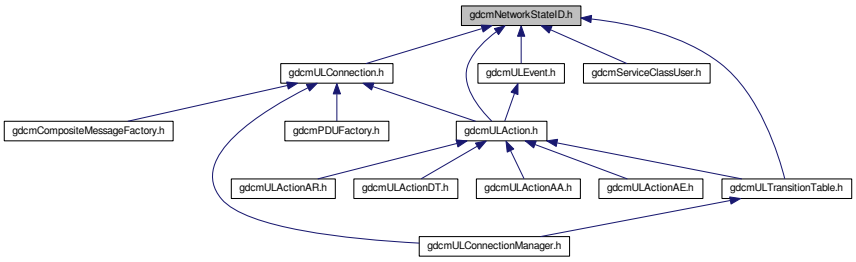
- enum gdcm::network::EEventID {
  gdcm::network::eAASSOCIATERequestLocalUser = 0,
  gdcm::network::eTransportConnConfirmLocal,
  gdcm::network::eASSOCIATE_ACPDUreceived,
  gdcm::network::eASSOCIATE_RJPDUreceived,
  gdcm::network::eTransportConnIndicLocal,
  gdcm::network::eAASSOCIATE_RQPDUreceived,
  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

28.157 gdcmNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- gdcm
- 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)

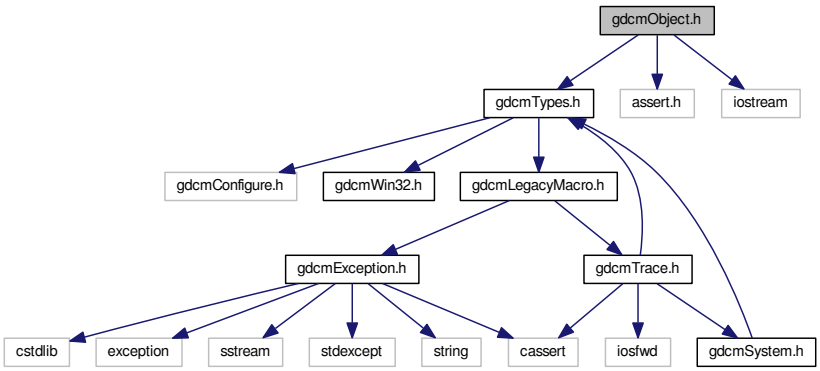
Variables

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

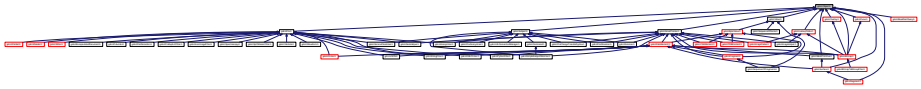
28.158 gdcmObject.h File Reference

```
#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.
- class gdcm::SmartPointer< ObjectType >

  Class for Smart Pointer.

Namespaces

- gdcm

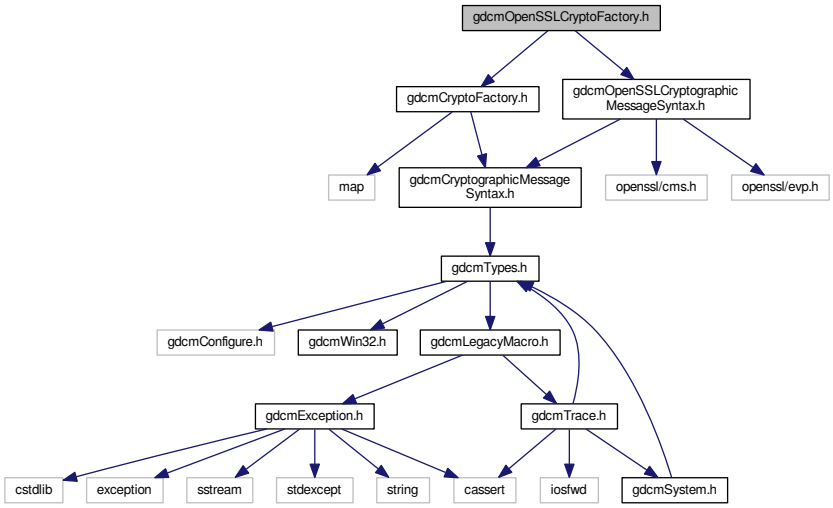
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)

28.159 gdcmOpenSSLCryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLCryptoFactory.h:



Classes

- class gdcm::OpenSSLCryptoFactory

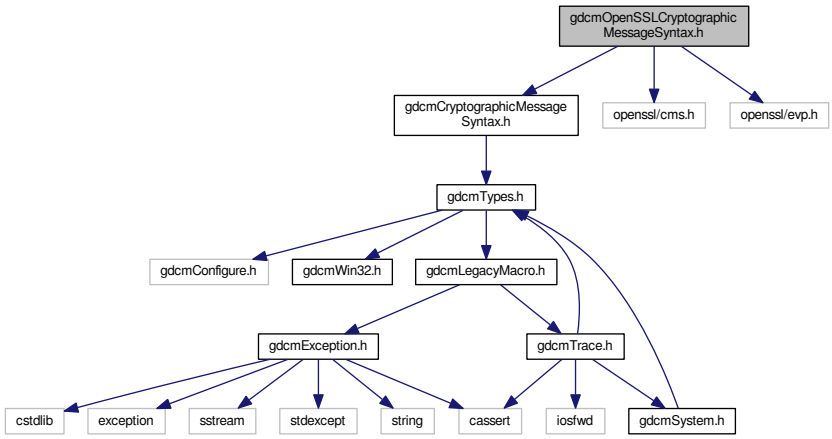
Namespaces

- gdcm

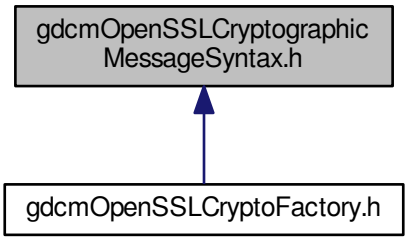
28.160 gdcmOpenSSLCryptographicMessageSyntax.h File Reference

```
#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

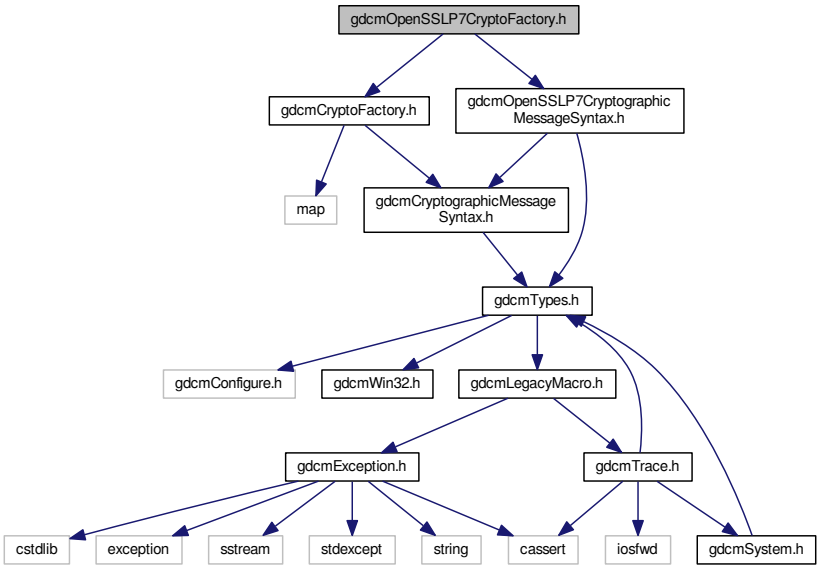
Namespaces

- gdcm

28.161 gdcmOpenSSLP7CryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLP7CryptographicMessageSyntax.h"
```

Include dependency graph for gdcmOpenSSLP7CryptoFactory.h:



Classes

- class gdcm::OpenSSLP7CryptoFactory

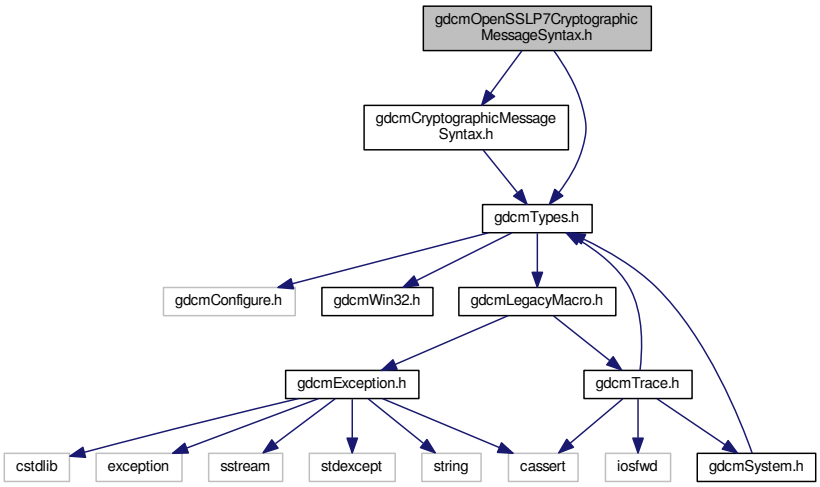
Namespaces

- gdcm

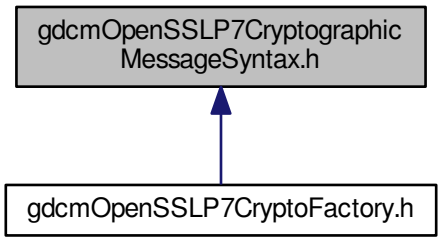
28.162 gdcmOpenSSLP7CryptographicMessageSyntax.h File Reference

```
#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

  Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

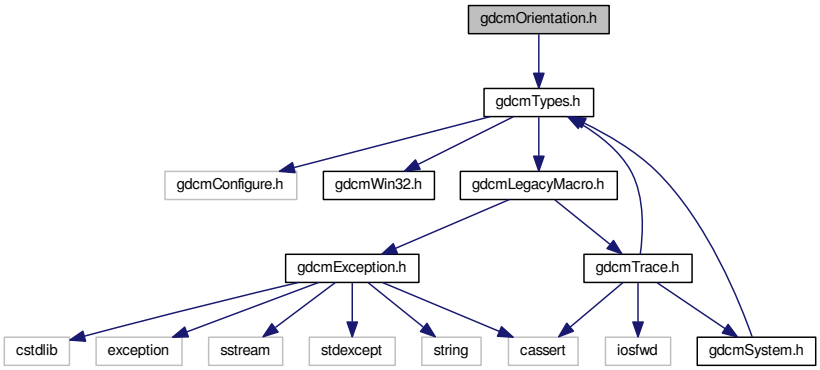
Namespaces

- gdcm

28.163 gdcmOrientation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOrientation.h:



Classes

- class gdcm::Orientation

  class to handle Orientation

Namespaces

- gdcm

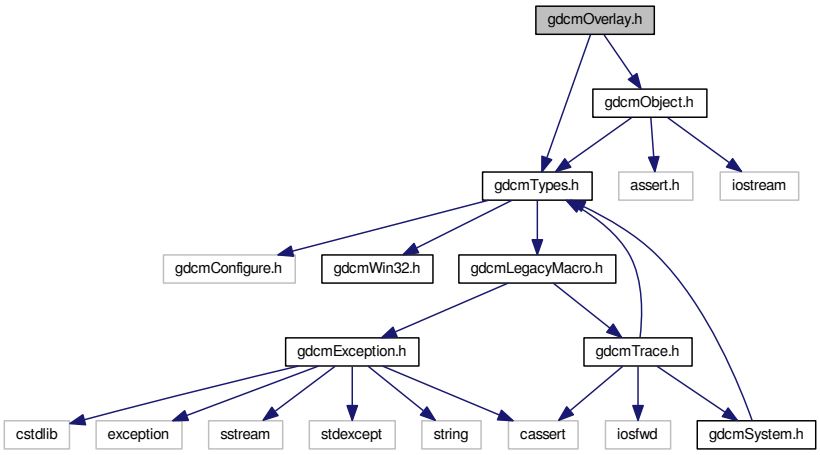
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)

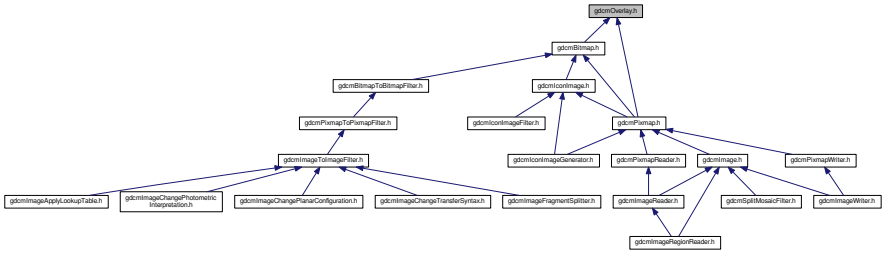
28.164 gdcmOverlay.h File Reference

```
#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

- gdcm

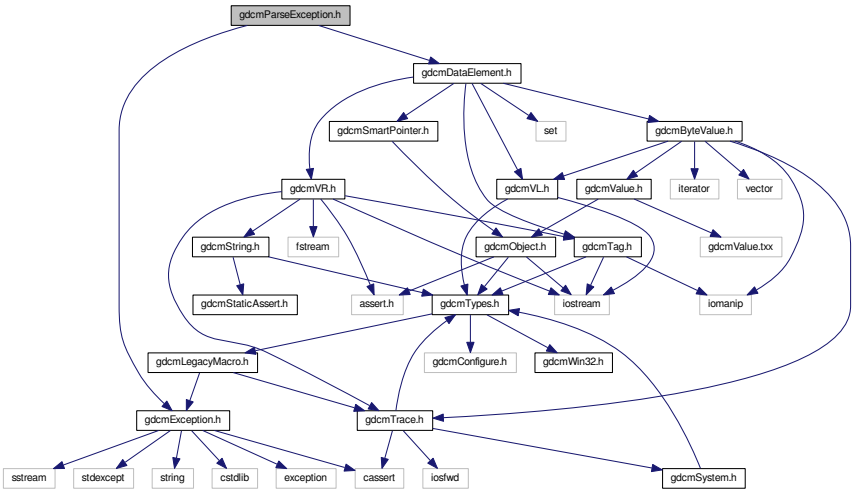
28.165 gdcmpap3.dox File Reference

28.166 gdcmParseException.h File Reference

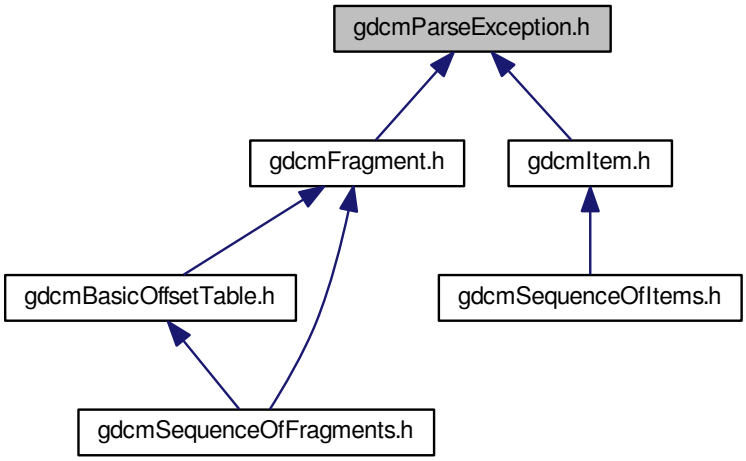
```
#include "gdcmException.h"
```

```
#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.

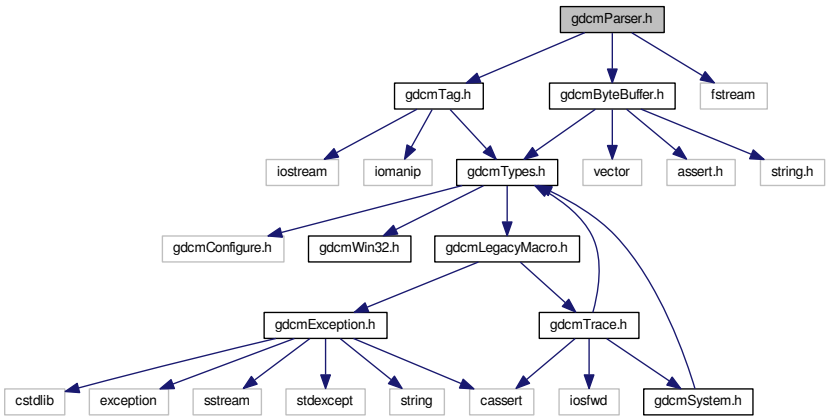
Namespaces

- gdcm

28.167 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)

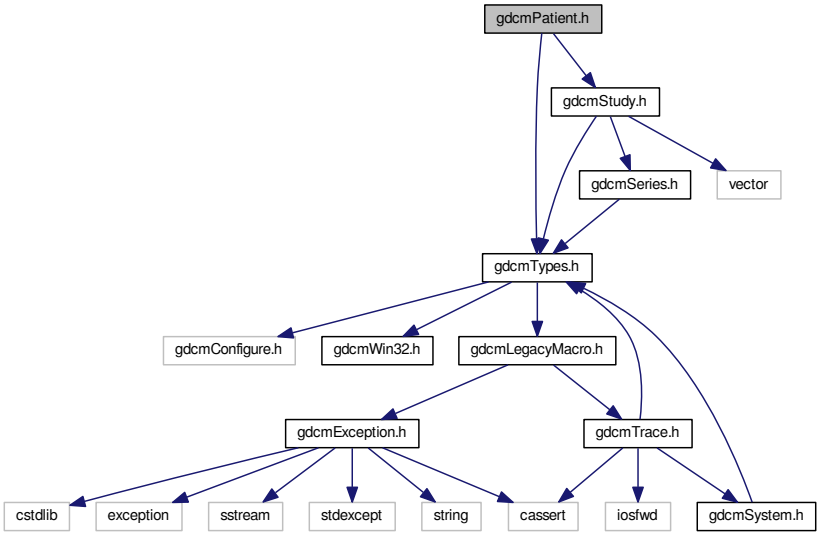
Namespaces

- gdcm

28.168 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.

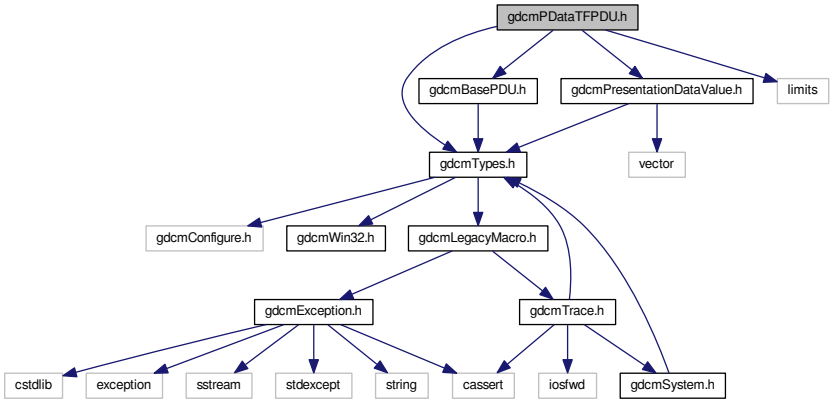
Namespaces

- gdcm

28.169 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 Table 9-22 P-DATA-TF PDU FIELDS.

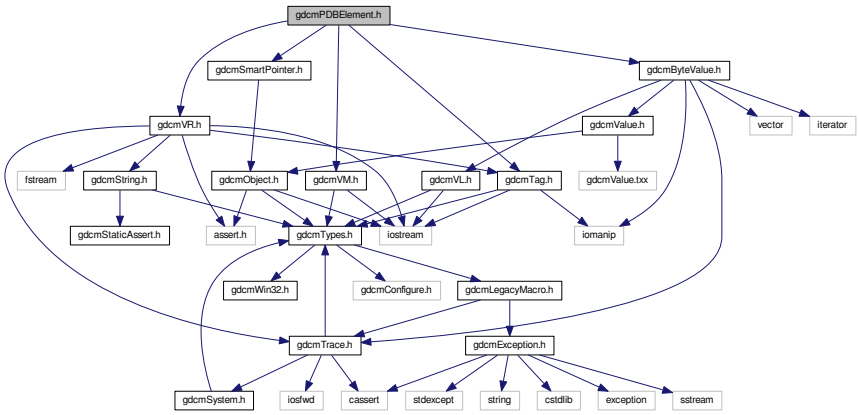
Namespaces

- gdcm
- gdcm::network

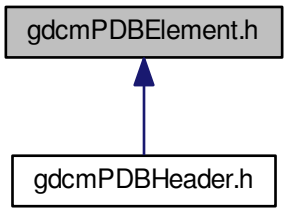
28.170 gdcmPDBElement.h File Reference

```
#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

- gdcm

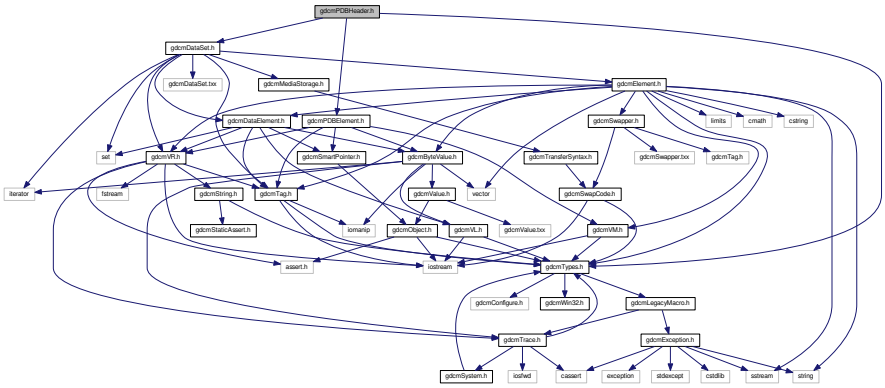
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PDBElement &val)

28.171 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

- gdcm

Functions

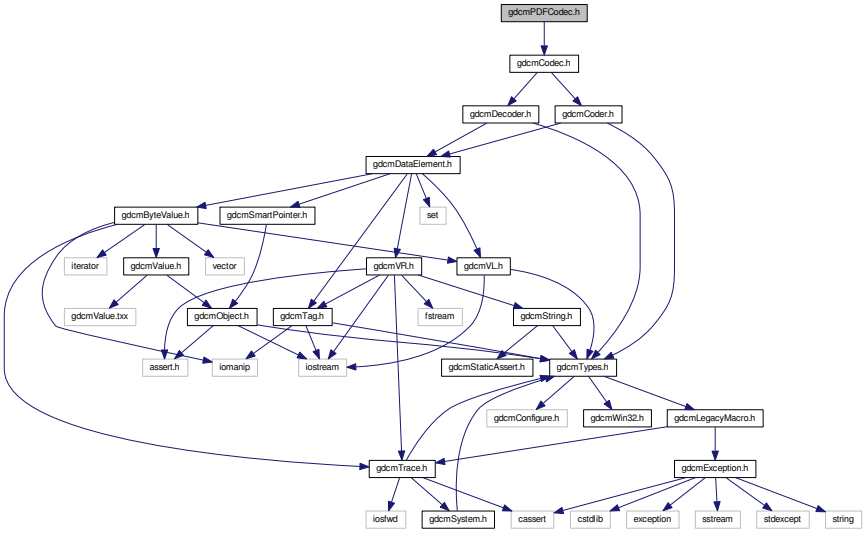
- std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)

28.172 gdcmpdf.dox File Reference

28.173 gdcmPDFCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmPDFCodec.h:



Classes

- class gdcm::PDFCodec

  PDFCodec class.

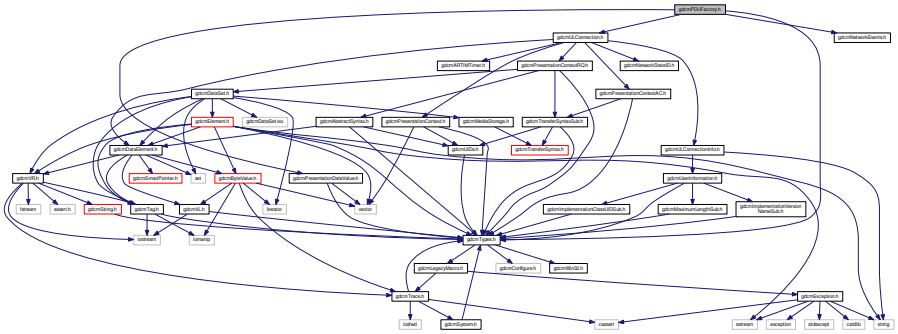
Namespaces

- gdcm

28.174 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 appropriate PDU. This way, the event loop doesn't have to know about all the different PDU types.

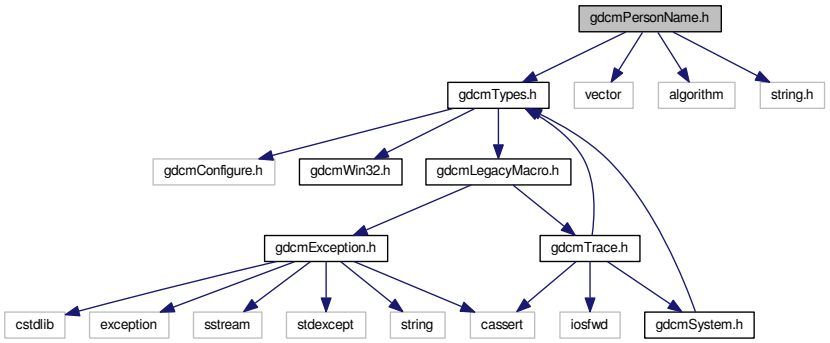
Namespaces

- gdcm
- gdcm::network

28.175 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.

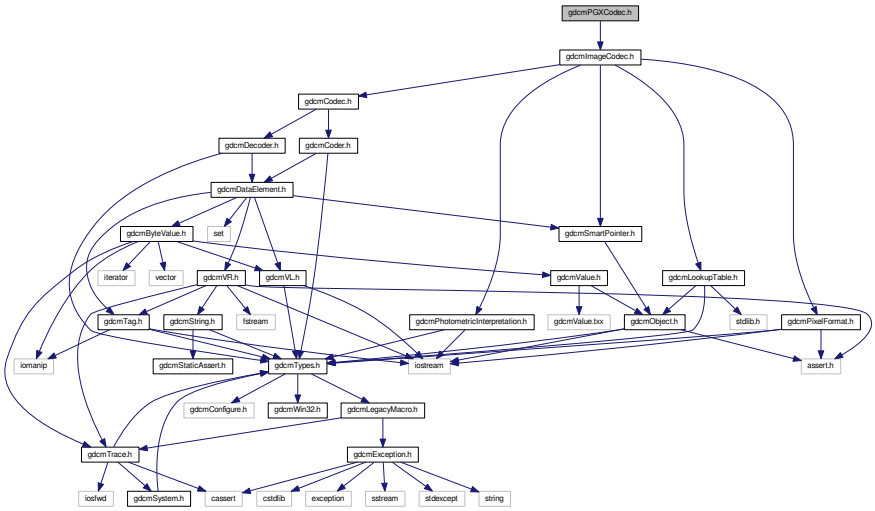
Namespaces

- gdcm

28.176 gdcmPGXCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPGXCodec.h:



Classes

- class gdcm::PGXCodec

  Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

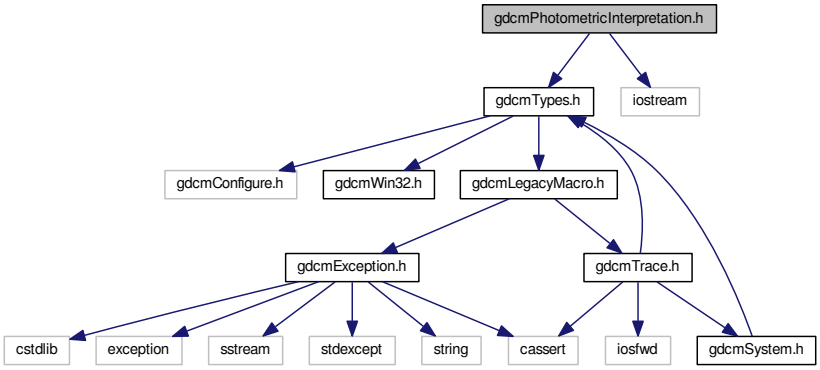
Namespaces

- gdcm

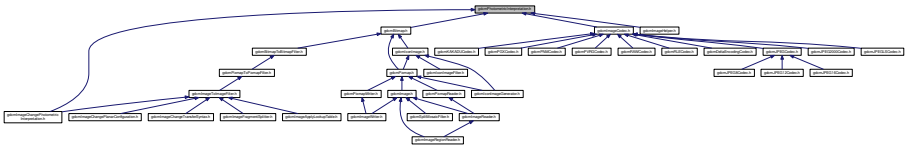
28.177 gdcmPhotometricInterpretation.h File Reference

```
#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

- gdcm

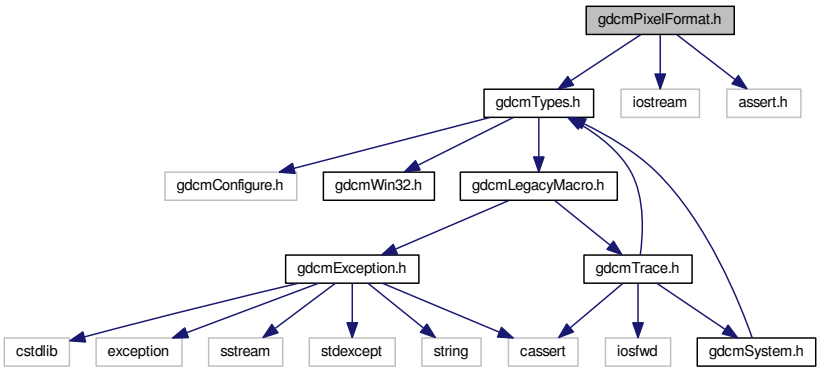
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PhotometricInterpretation &val)

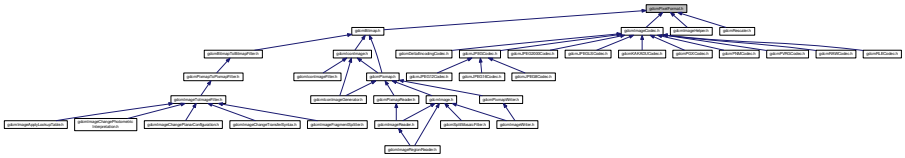
28.178 gdcmPixelFormat.h File Reference

```
#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

- gdcm

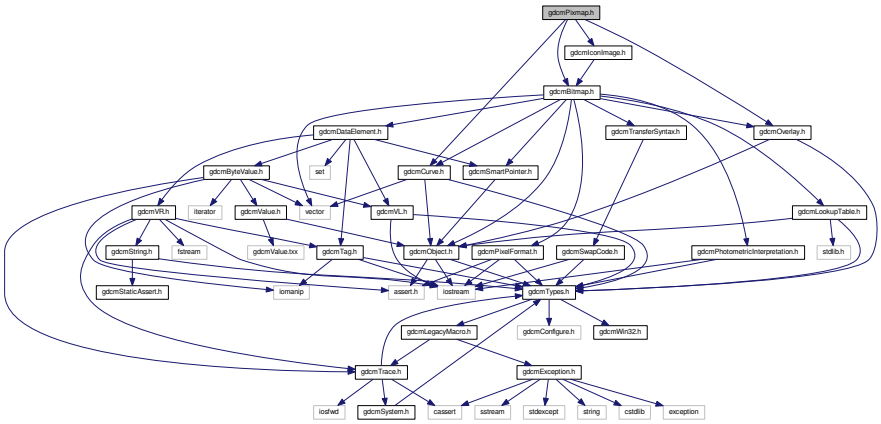
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PixelFormat &pf)

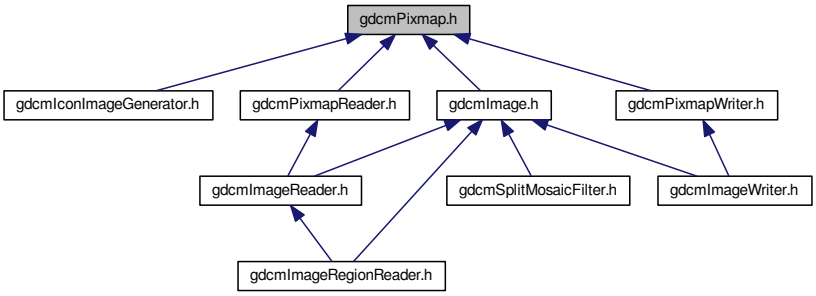
28.179 gdcmPixmap.h File Reference

```
#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 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)

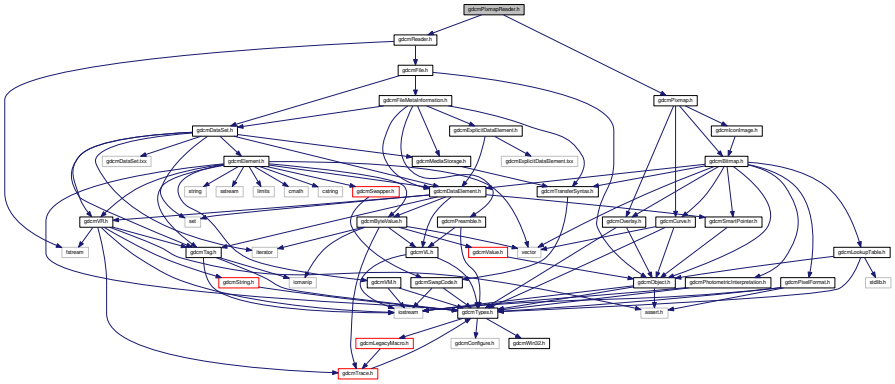
Namespaces

- gdcm

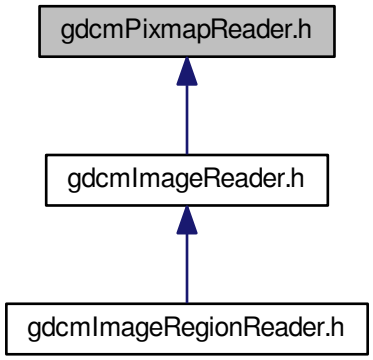
28.180 gdcmPixmapReader.h File Reference

```
#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.

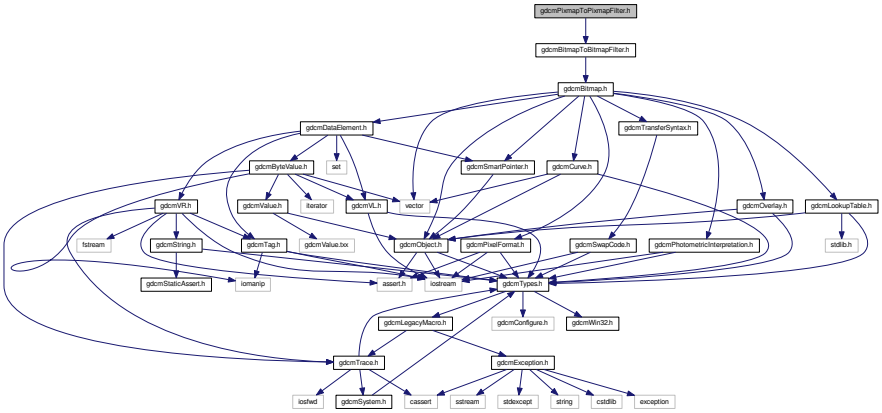
Namespaces

- gdcm

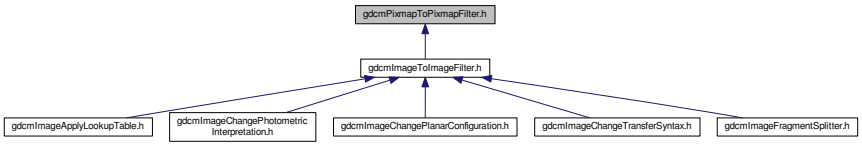
28.181 gdcmPixmapToPixmapFilter.h File Reference

```
#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 Super class for all filter taking an image and producing an output image.

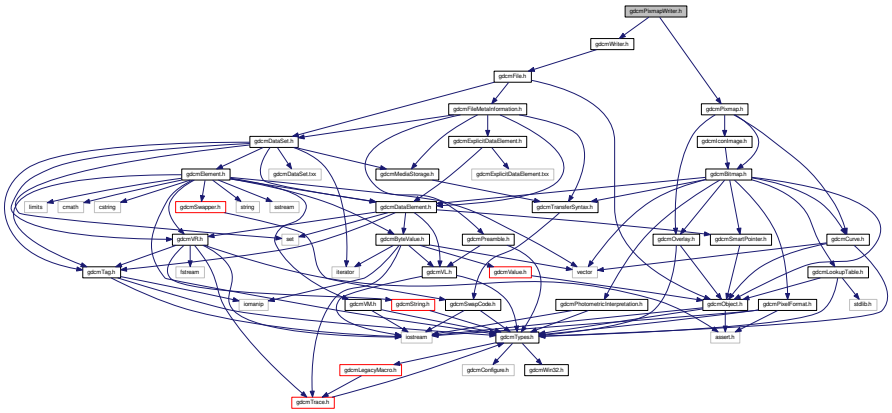
Namespaces

- gdcm

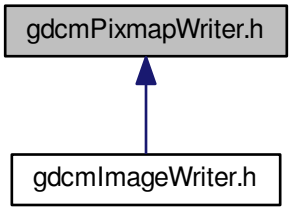
28.182 gdcmPixmapWriter.h File Reference

```
#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 This class will takes two inputs:

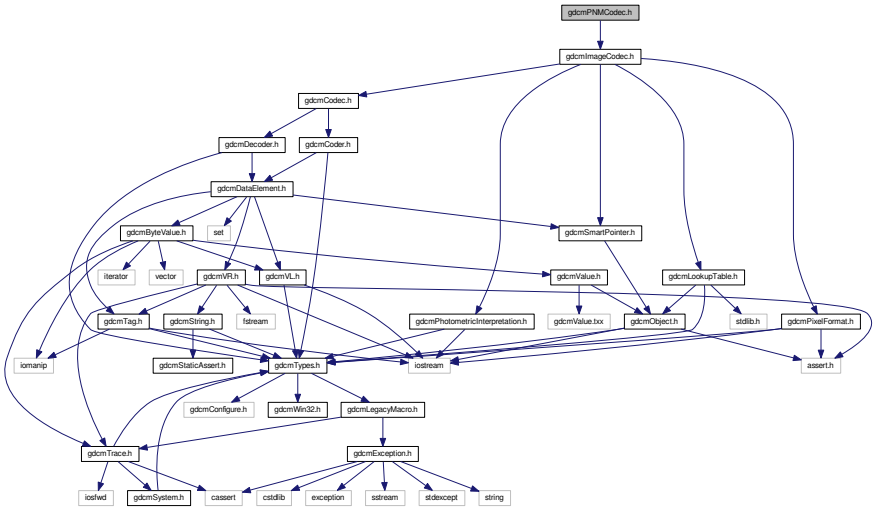
Namespaces

- gdcm

28.183 gdcmPNMCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPNMCodec.h:



Classes

- class gdcm::PNMCodec

  Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: `http://netpbm.↩ sourceforge.net/`.

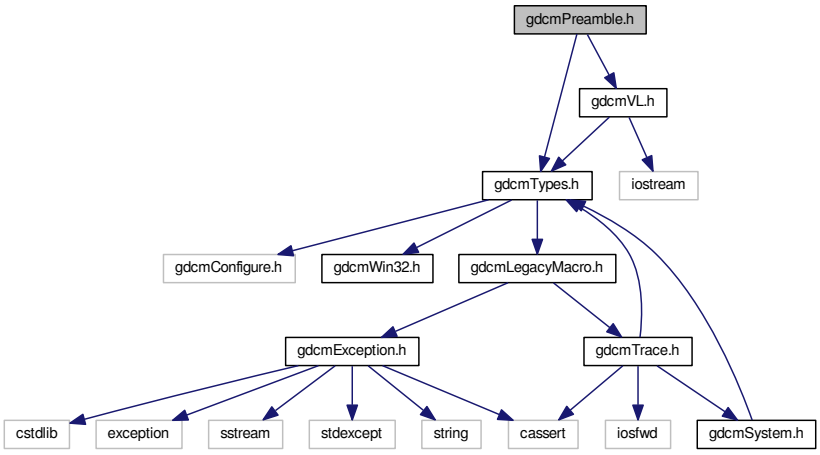
Namespaces

- gdcm

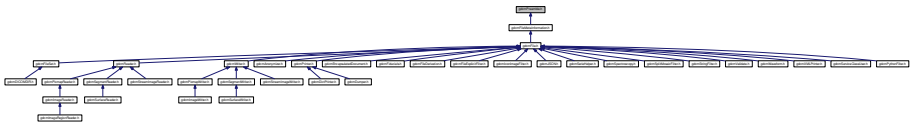
28.184 gdcmPreamble.h File Reference

```
#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

- gdcm

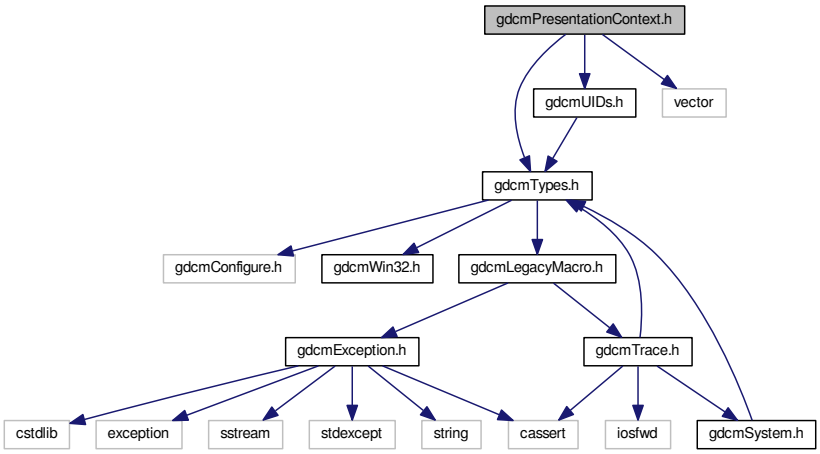
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Preamble &val)

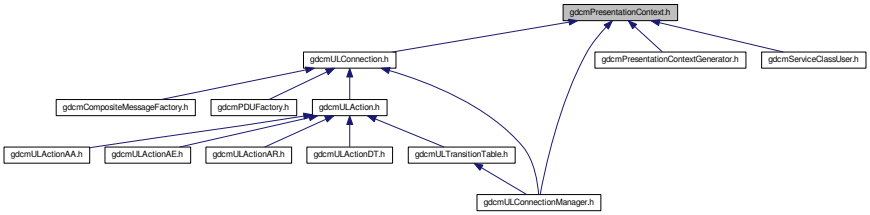
28.185 gdcmPresentationContext.h File Reference

```
#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.

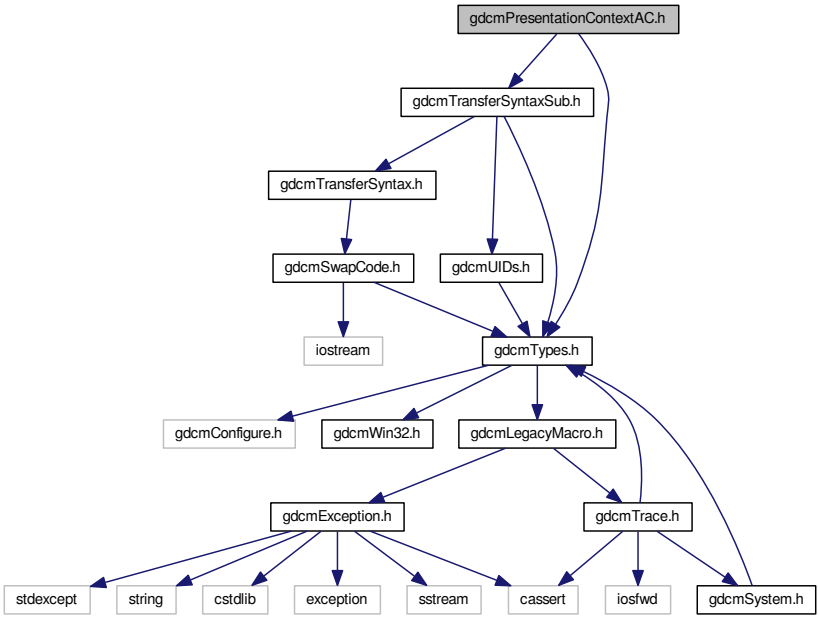
Namespaces

- gdcm

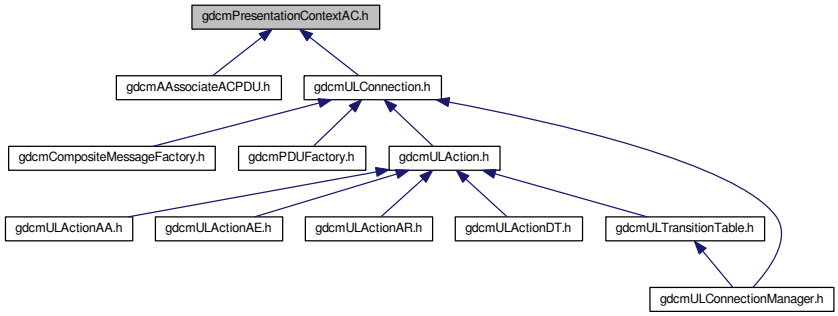
28.186 gdcmPresentationContextAC.h File Reference

```
#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 Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

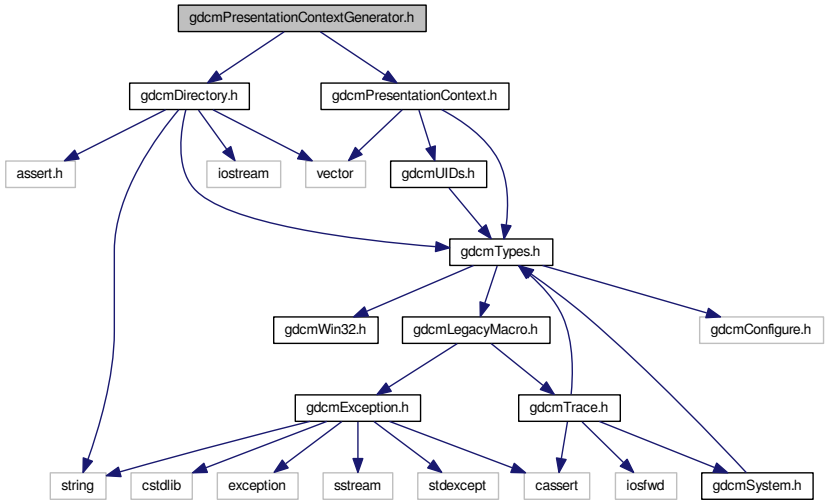
Namespaces

- gdcm
- gdcm::network

28.187 gdcmPresentationContextGenerator.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmPresentationContextGenerator.h:



Classes

- class gdcm::PresentationContextGenerator

  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.

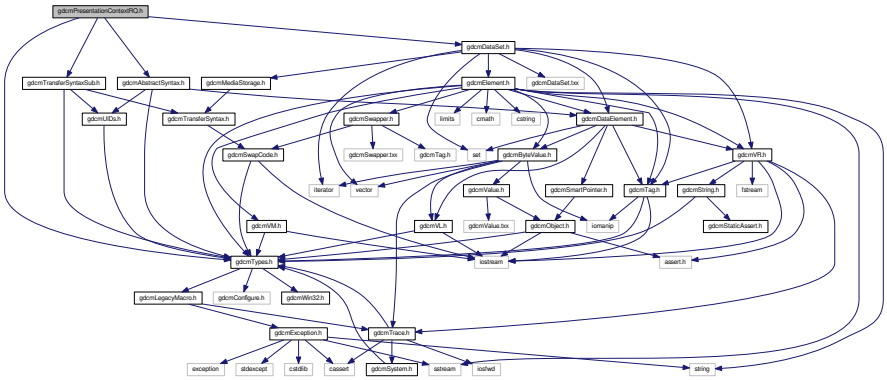
Namespaces

- gdcm

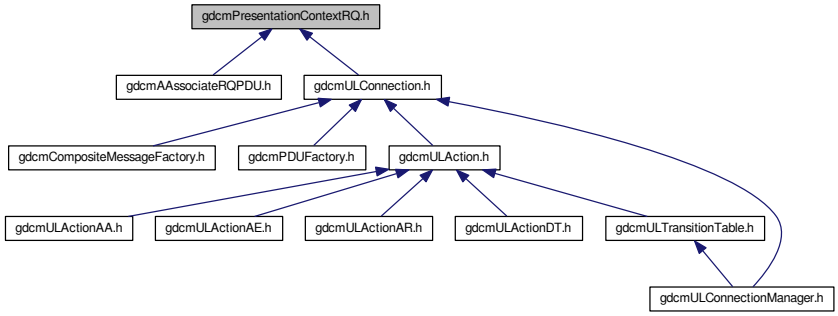
28.188 gdcmPresentationContextRQ.h File Reference

```
#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 Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

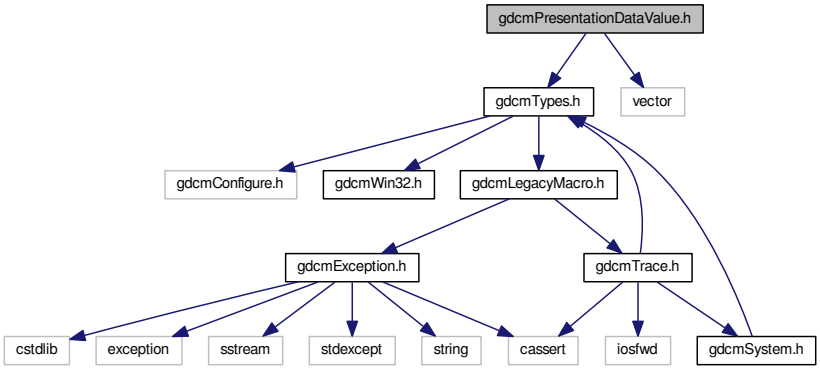
Namespaces

- gdcm
- gdcm::network

28.189 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 Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

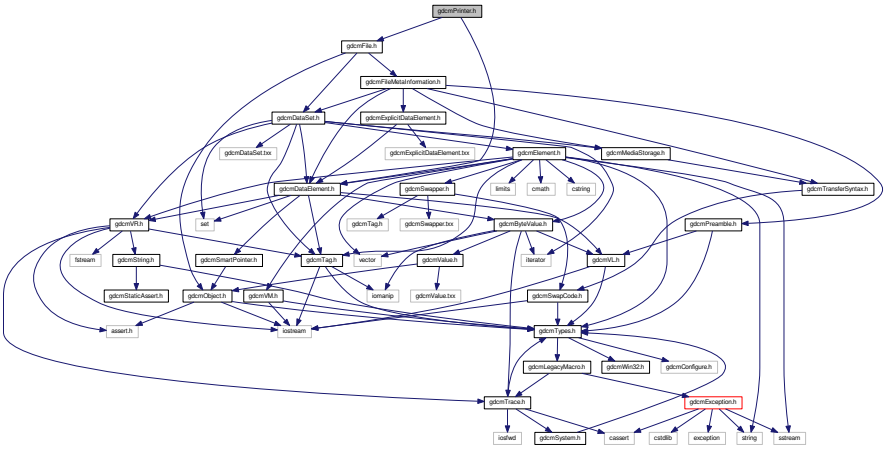
Namespaces

- gdcm
- gdcm::network

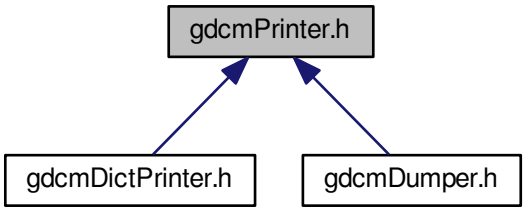
28.190 gdcmPrinter.h File Reference

```
#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

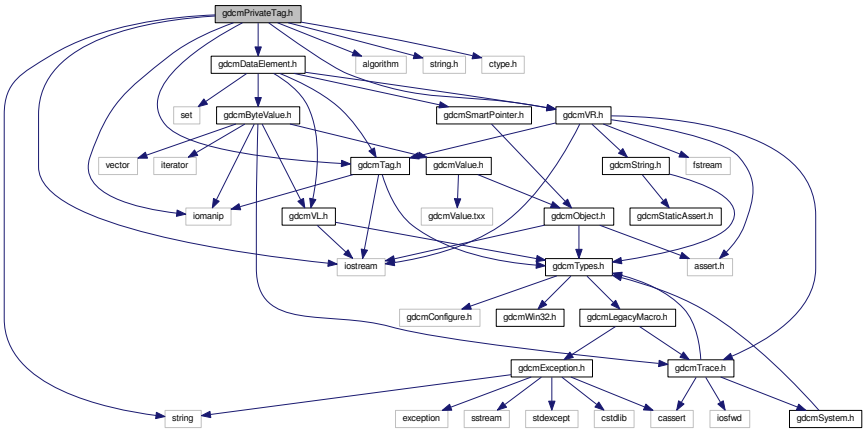
- gdcm

28.191 gdcmPrivateTag.h File Reference

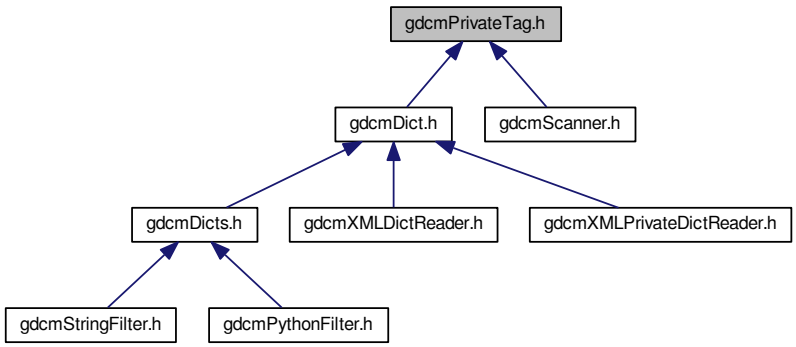
```
#include "gdcmTag.h"
```

```
#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

- gdcm

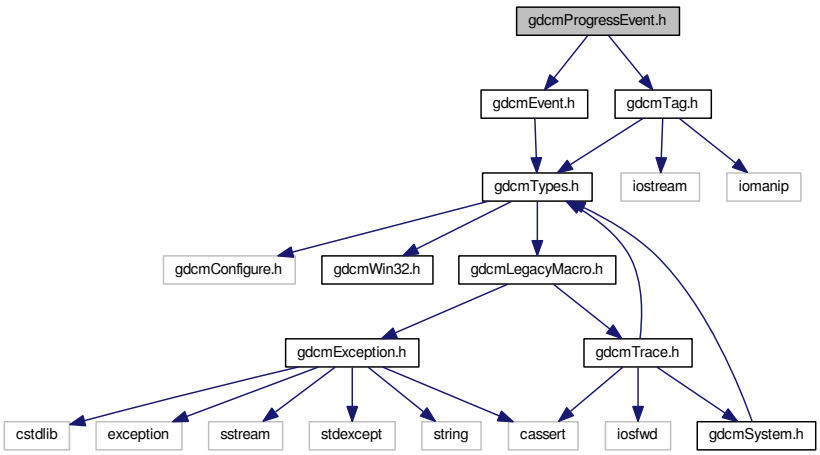
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)

28.192 gdcmProgressEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmProgressEvent.h:



Classes

- class gdcm::ProgressEvent

  ProgressEvent Special type of event triggered during.

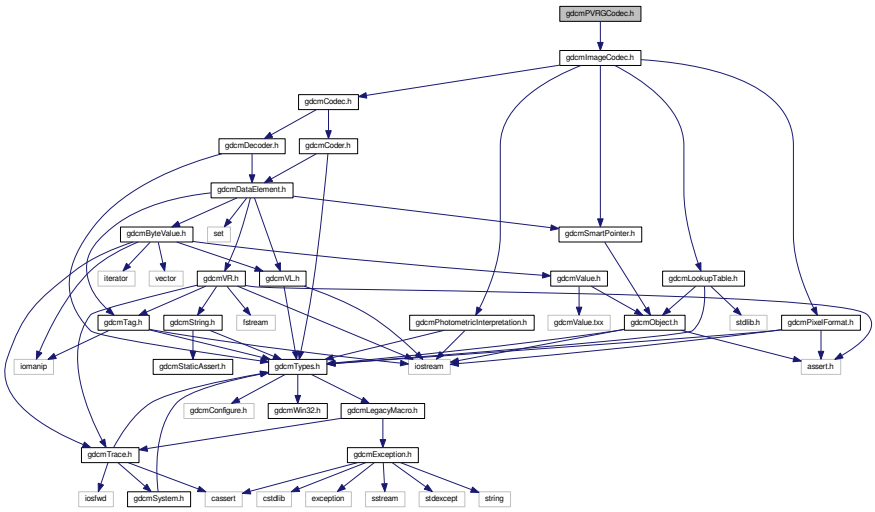
Namespaces

- gdcm

28.193 gdcmPVRGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPVRGCodec.h:



Classes

- class gdcm::PVRGCodec
  PVRGCodec.

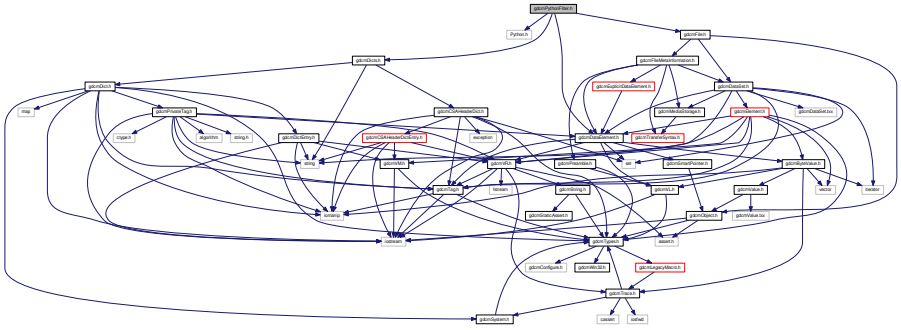
Namespaces

- gdcm

28.194 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.

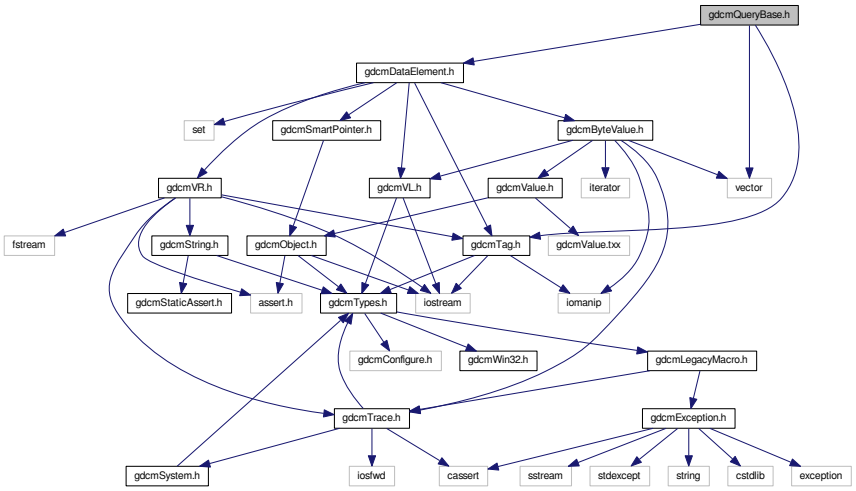
Namespaces

- gdcm

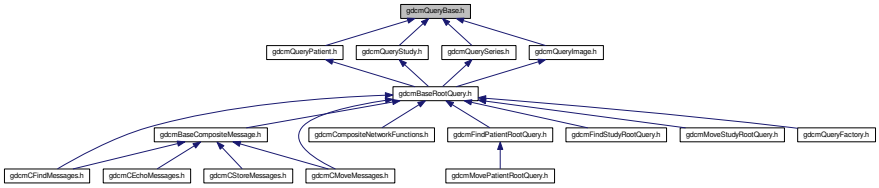
28.195 gdcmQueryBase.h File Reference

```
#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 contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

Namespaces

- gdcm

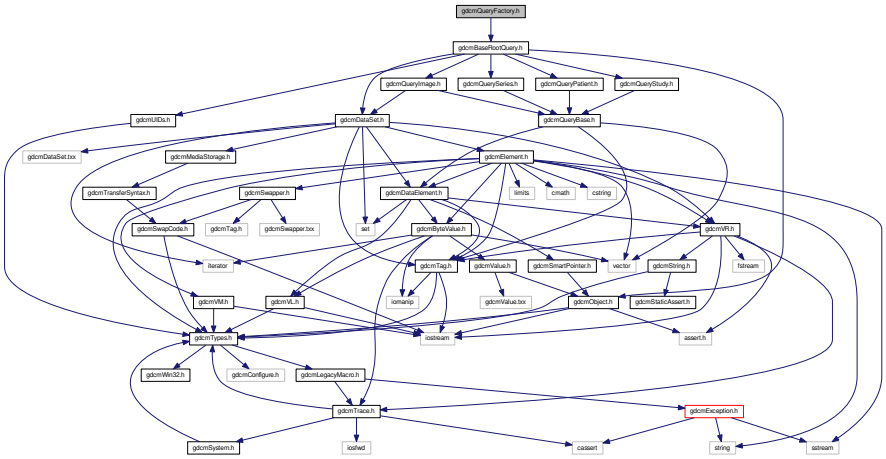
Enumerations

- enum gdcm::ERootType {
  gdcm::ePatientRootType,
  gdcm::eStudyRootType }

28.196 gdcmQueryFactory.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmQueryFactory.h:



Classes

- class gdcm::QueryFactory

  QueryFactory.h.

Namespaces

- gdcm

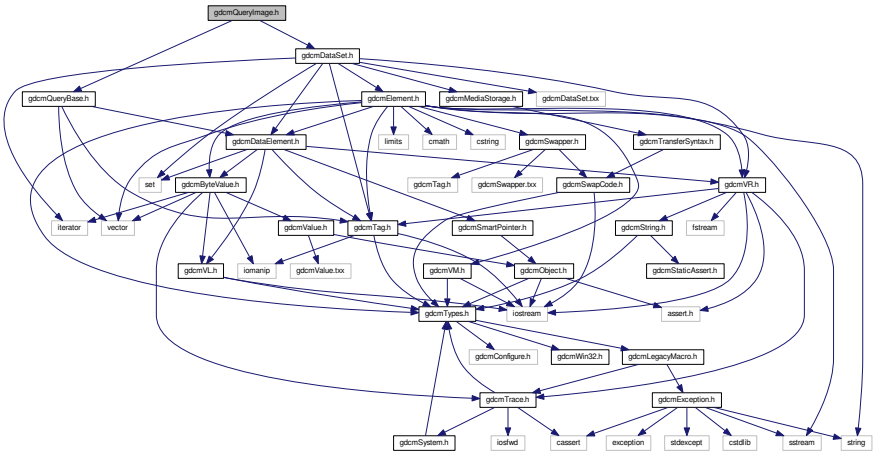
Enumerations

- 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 }

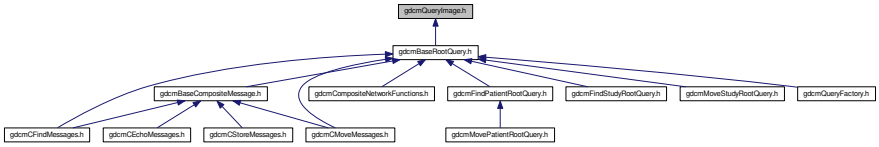
28.197 gdcmQueryImage.h File Reference

```
#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

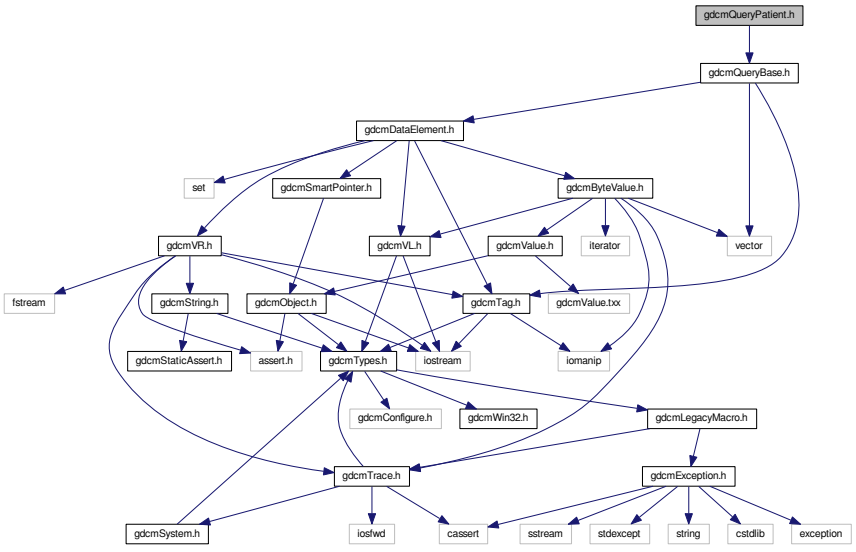
  QueryImage contains: class to construct an image-based query for C-FIND and C-MOVE.

Namespaces

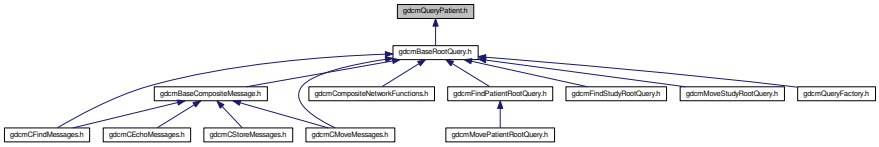
- gdcm

28.198 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 contains: class to construct a patient-based query for c-find and c-move.

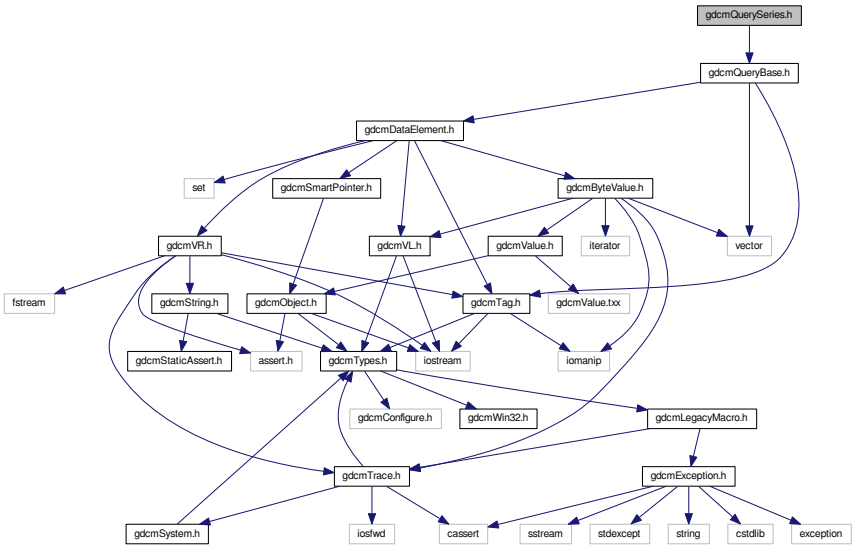
Namespaces

- gdcm

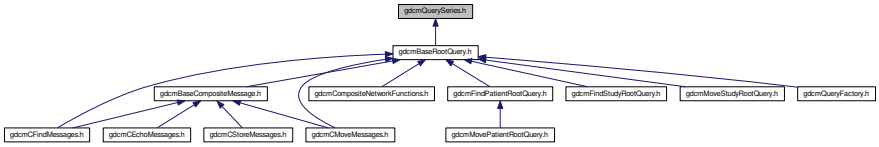
28.199 gdcmQuerySeries.h File Reference

```
#include "gdcmQueryBase.h"
```

Include dependency graph for gdcmQuerySeries.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::QuerySeries

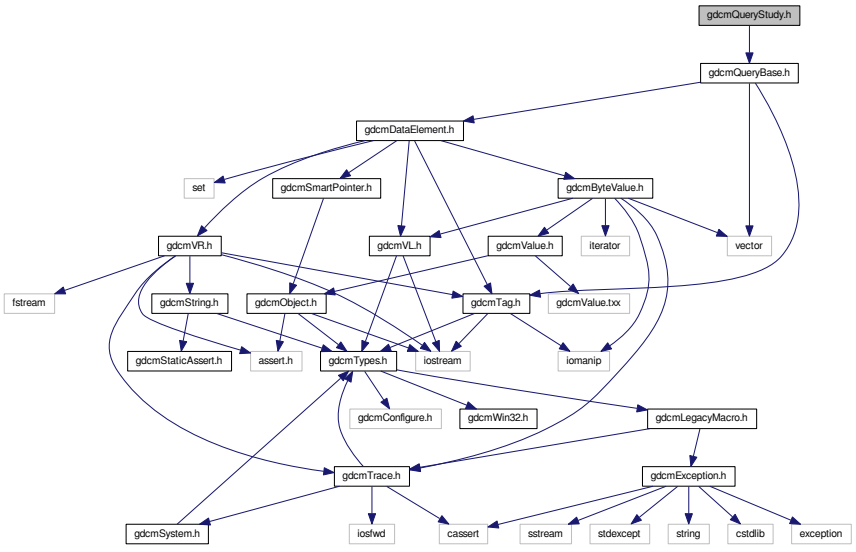
  QuerySeries contains: class to construct a series-based query for c-find and c-move.

Namespaces

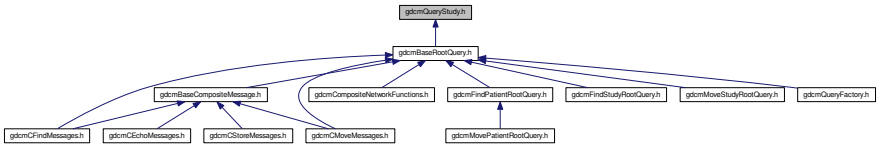
- gdcm

28.200 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 contains: class to construct a study-based query for C-FIND and C-MOVE.

Namespaces

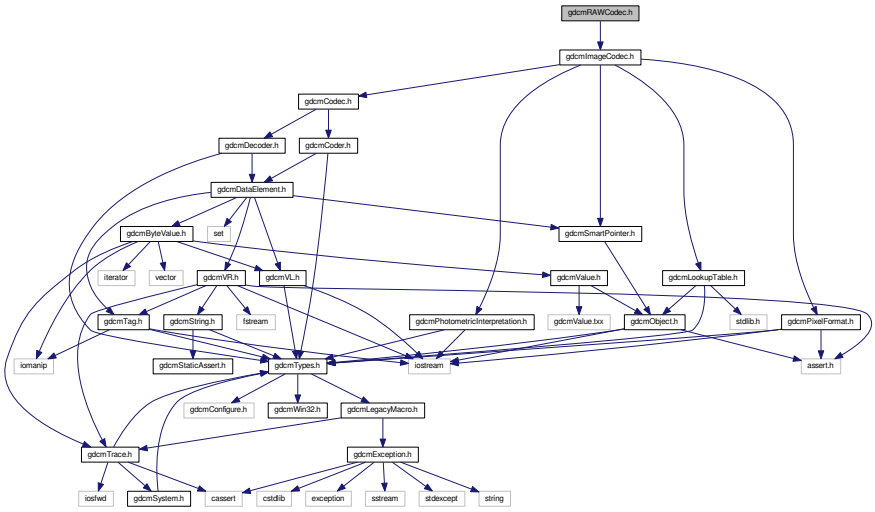
- gdcm

28.201 gdcmraw.dox File Reference

28.202 gdcmRAWCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmRAWCodec.h:



Classes

- class gdcm::RAWCodec

RAWCodec class.

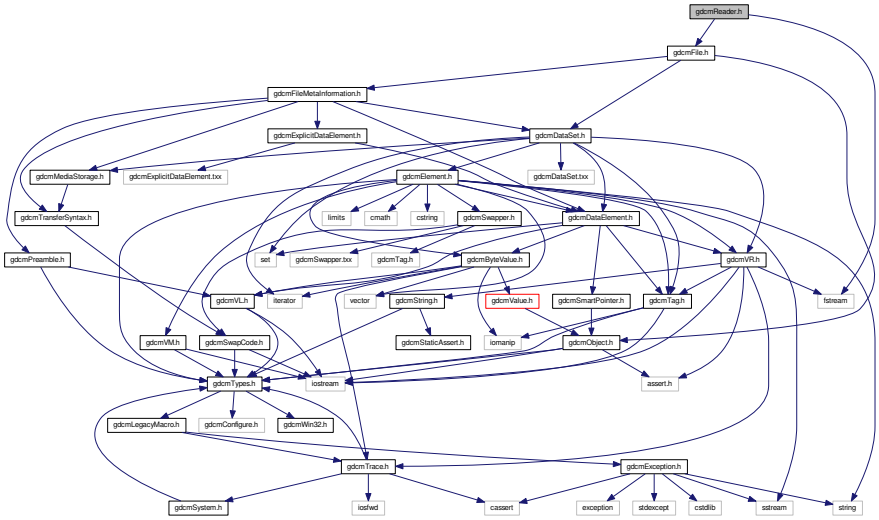
Namespaces

- gdcm

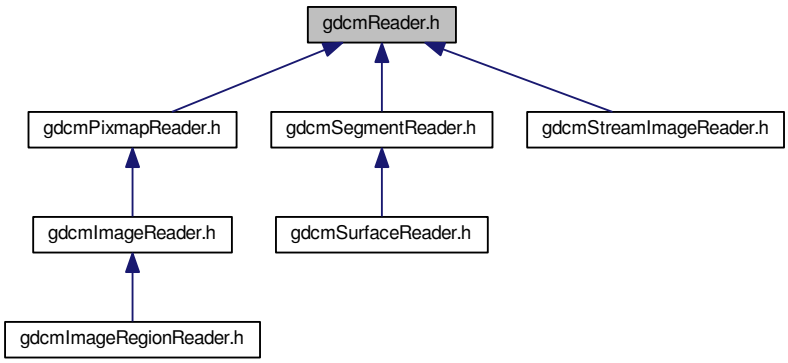
28.203 gdcmReader.h File Reference

```
#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)

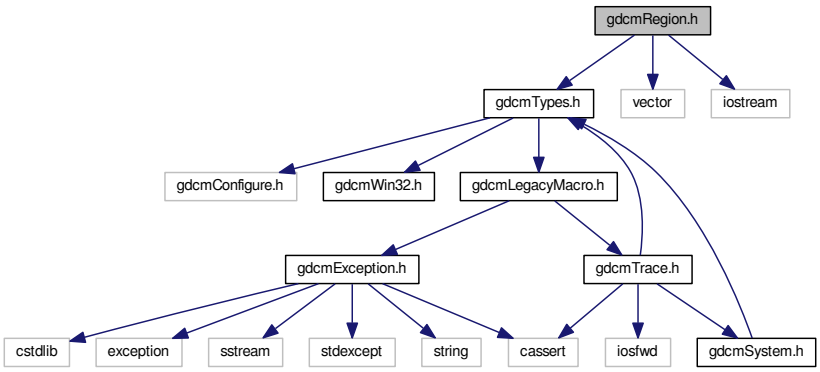
Namespaces

- gdcm

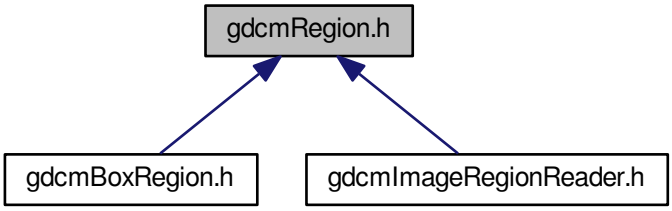
28.204 gdcmRegion.h File Reference

```
#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

- gdcm

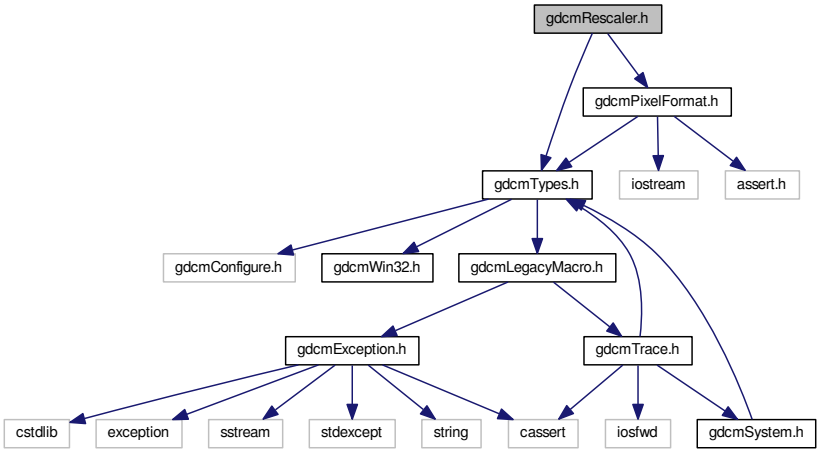
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)

28.205 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmRescaler.h:



Classes

- class gdcm::Rescaler

  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.

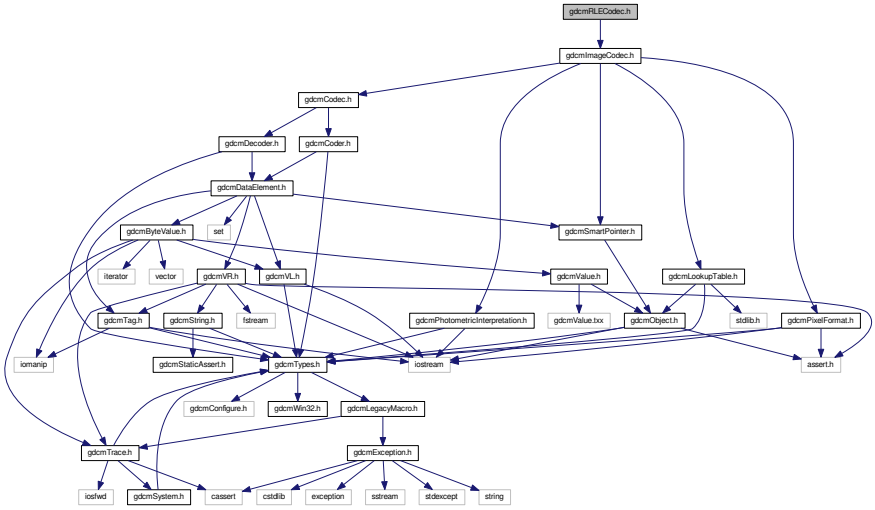
Namespaces

- gdcm

28.206 gdcmRLECodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmRLECodec.h:



Classes

- class gdcm::RLECodec

  Class to do RLE.

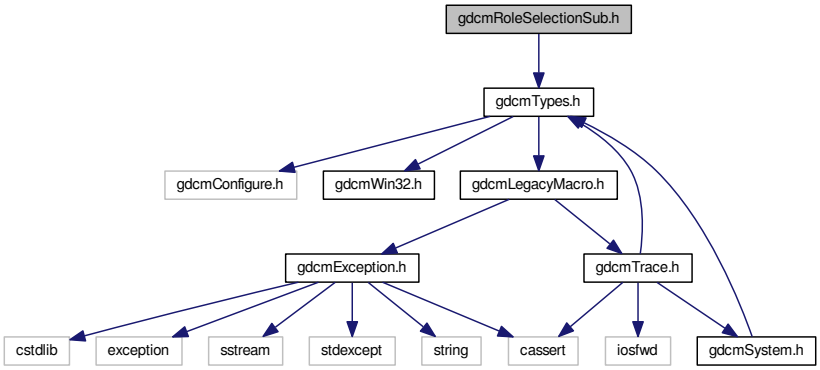
Namespaces

- gdcm

28.207 gdcmRoleSelectionSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmRoleSelectionSub.h:



Classes

- class gdcm::network::RoleSelectionSub

  RoleSelectionSub PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

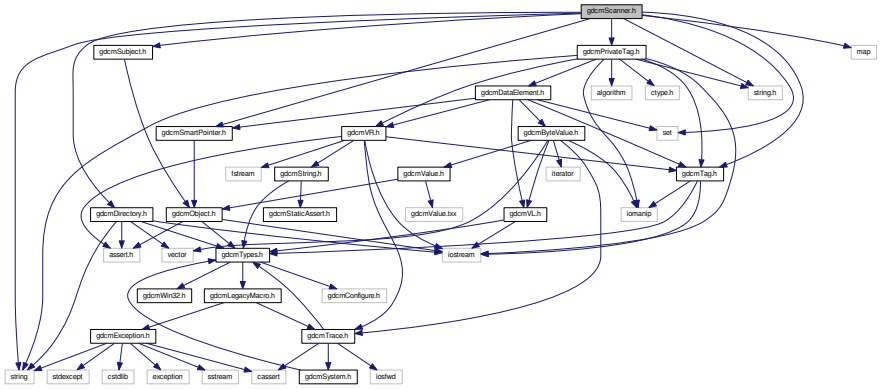
- gdcm
- gdcm::network

28.208 gdcmscanner.dox File Reference

28.209 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 This filter is meant for quickly browsing a FileSet (a set of files on disk). Special consideration are taken so as to read the mimimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute.

Namespaces

- gdcm

Functions

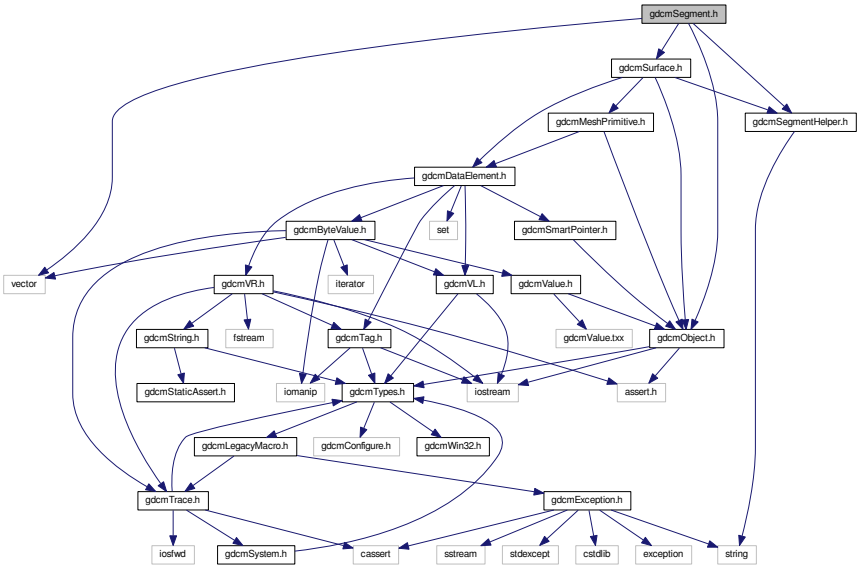
- std::ostream & gdcm::operator<< (std::ostream &os, const Scanner &s)

28.210 gdcmscu.dox File Reference

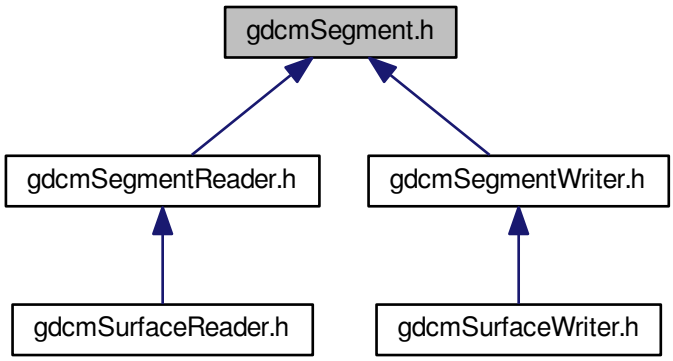
28.211 gdcmSegment.h File Reference

```
#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. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

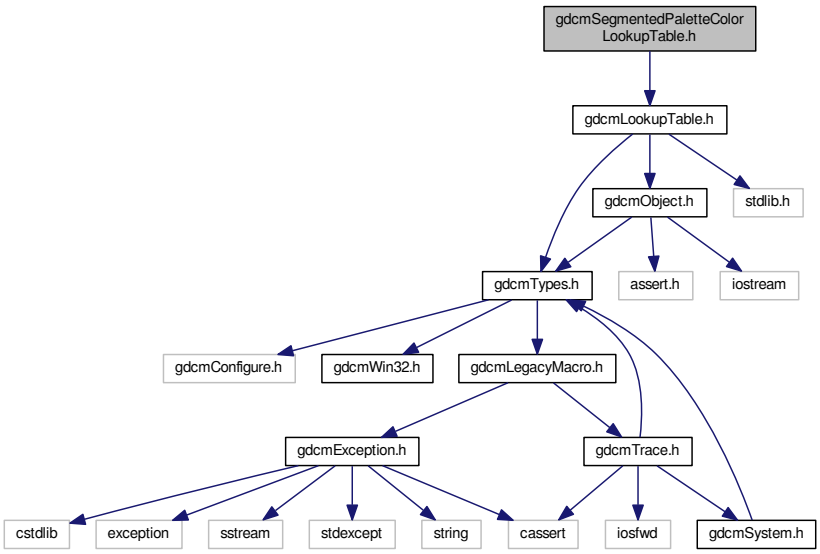
Namespaces

- gdcm

28.212 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



Classes

- class gdcm::SegmentedPaletteColorLookupTable

  SegmentedPaletteColorLookupTable class.

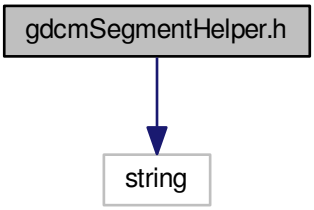
Namespaces

- gdcm

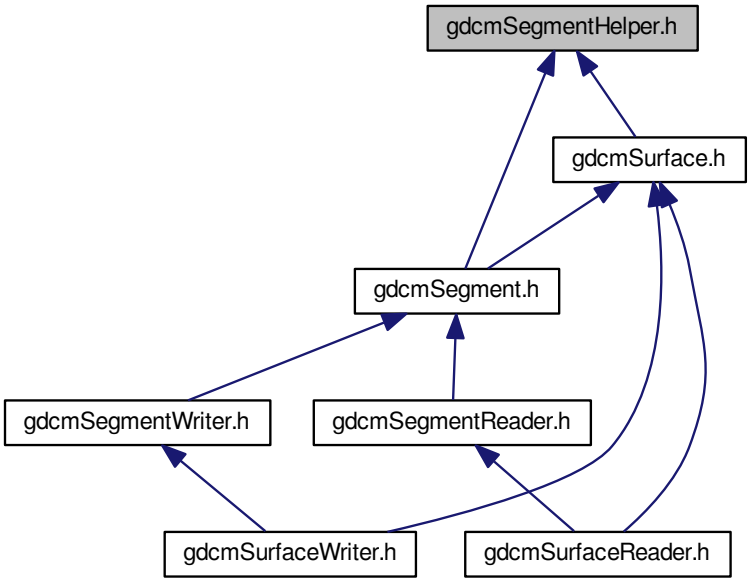
28.213 gdcmSegmentHelper.h File Reference

```
#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

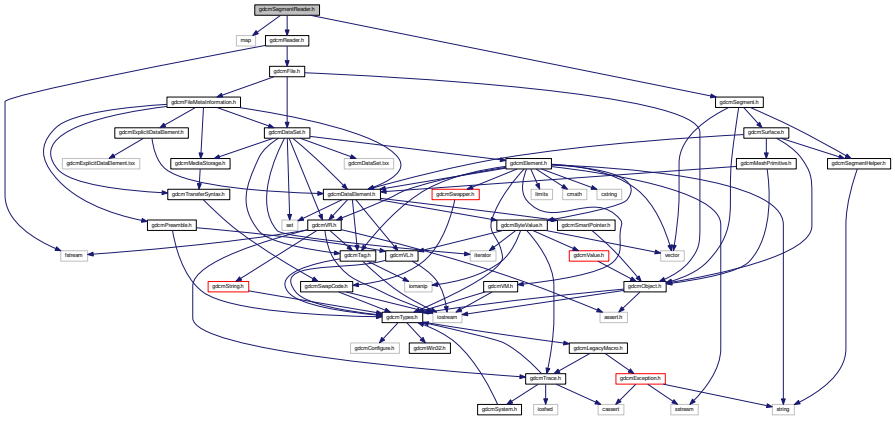
- gdcm

- gdcm::SegmentHelper

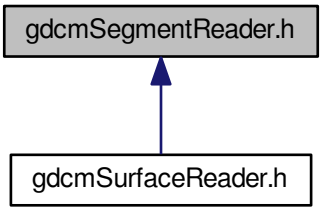
28.214 gdcmSegmentReader.h File Reference

```
#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. It reads attributes of group 0x0062.

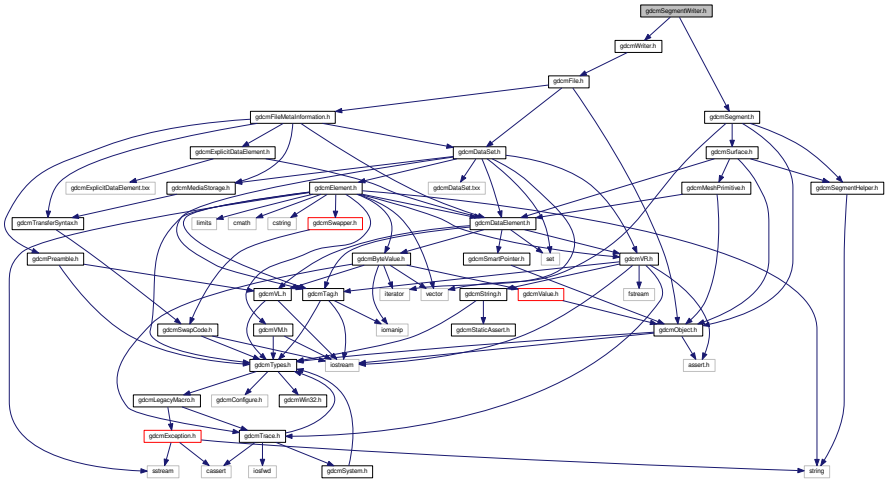
Namespaces

- gdcm

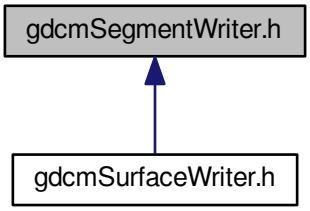
28.215 gdcmSegmentWriter.h File Reference

```
#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. It writes attributes of group 0x0062.

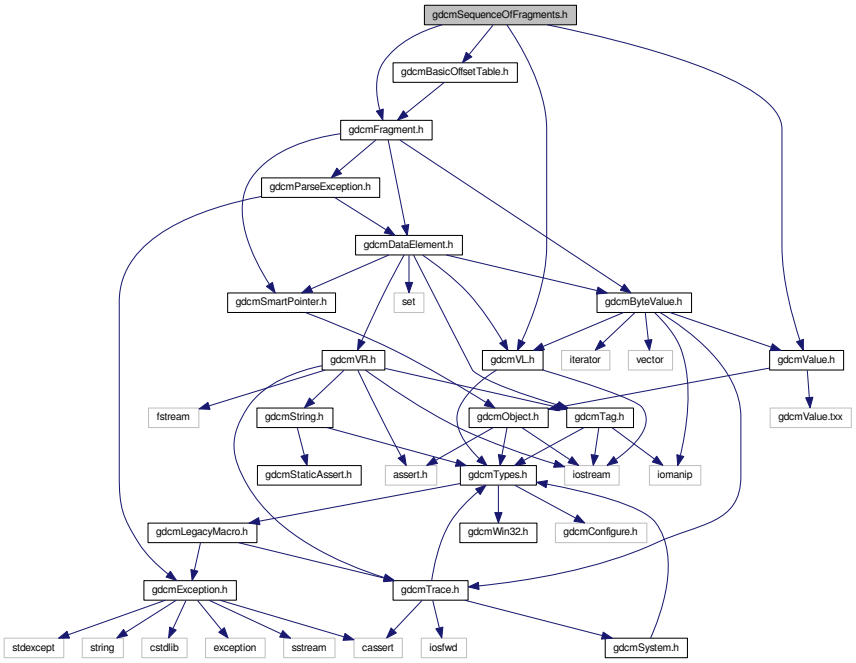
Namespaces

- gdcm

28.216 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.

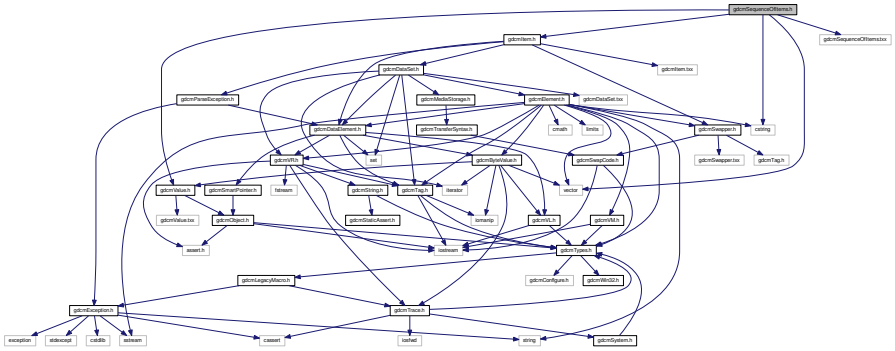
Namespaces

- gdcm

28.217 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 (value representation : SQ)

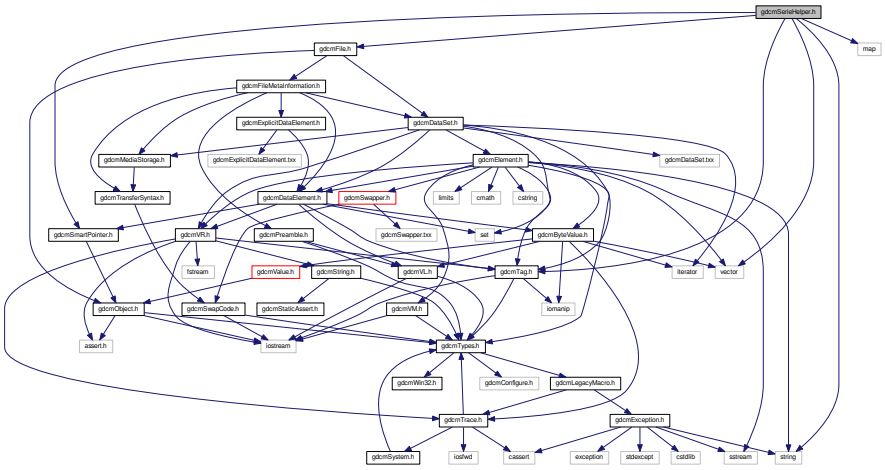
Namespaces

- gdcm

28.218 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.
- struct gdcm::SerieHelper::Rule
- 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 disapear soon, you've been warned.

Namespaces

- gdcm

Typedefs

- typedef bool(∗ gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER)(File ∗, File ∗)
- typedef std::vector
  < SmartPointer< FileWithName > > gdcm::FileList

Enumerations

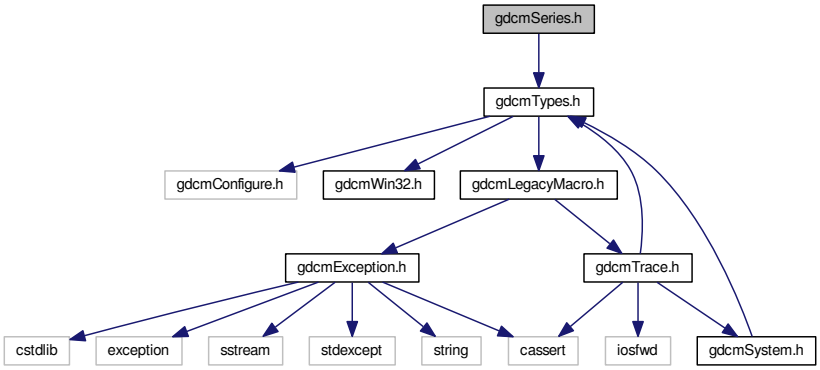
- enum gdcm::CompOperators {
  gdcm::GDCM_EQUAL = 0,
  gdcm::GDCM_DIFFERENT,
  gdcm::GDCM_GREATER,
  gdcm::GDCM_GREATEROREQUAL,
  gdcm::GDCM_LESS,
  gdcm::GDCM_LESSOREQUAL }

- enum gdcm::LodModeType {
  gdcm::LD_ALL = 0x00000000,
  gdcm::LD_NOSEQ = 0x00000001,
  gdcm::LD_NOSHADOW = 0x00000002,
  gdcm::LD_NOSHADOWSEQ = 0x00000004 }

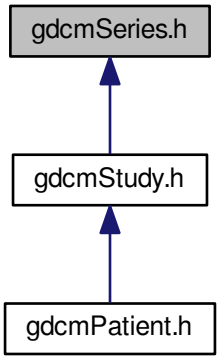
28.219 gdcmSeries.h File Reference

```
#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.

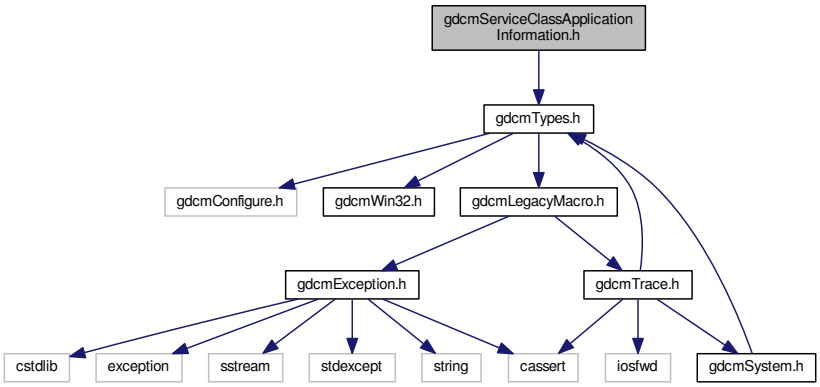
Namespaces

- gdcm

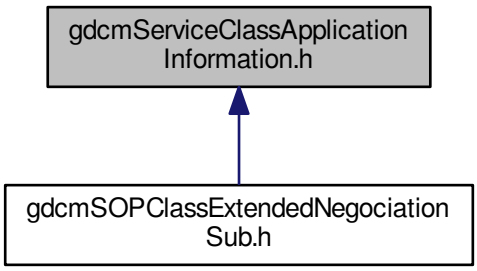
28.220 gdcmServiceClassApplicationInformation.h File Reference

```
#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

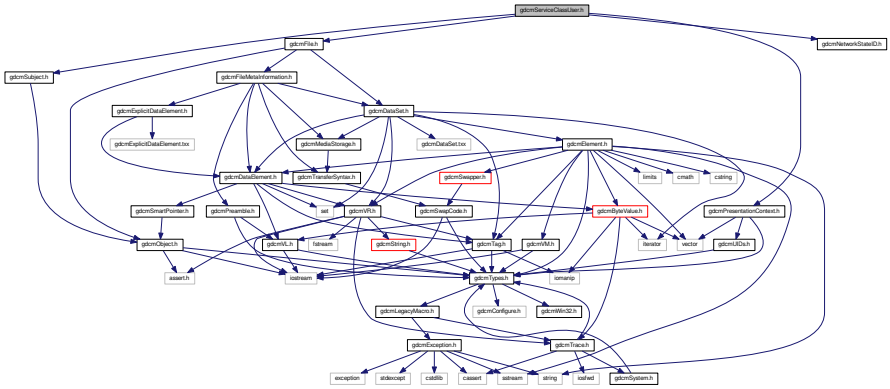
Namespaces

- gdcm
- gdcm::network

28.221 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.

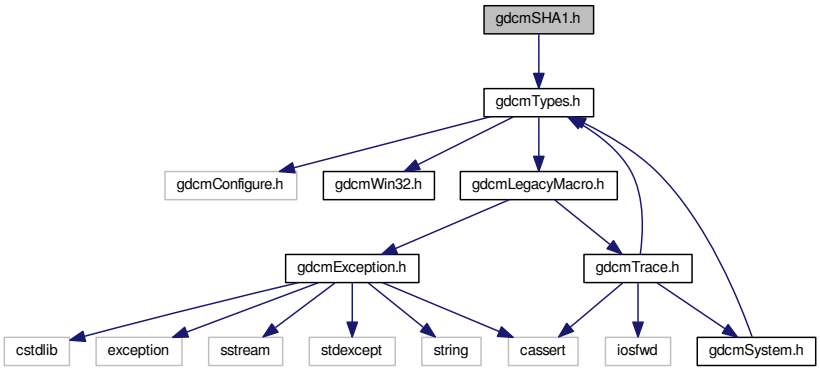
Namespaces

- gdcm
- gdcm::network

28.222 gdcmSHA1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmSHA1.h:



Classes

- class gdcm::SHA1

  Class for SHA1.

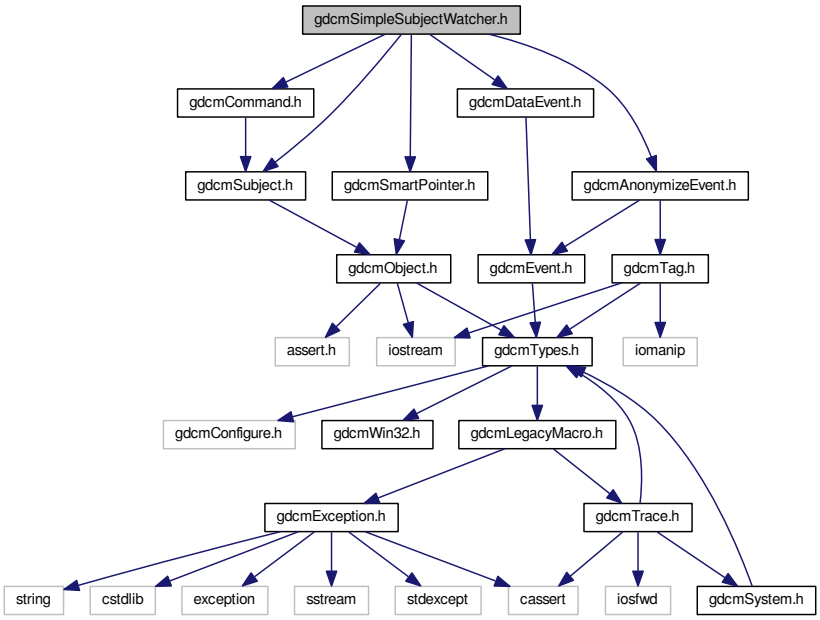
Namespaces

- gdcm

28.223 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 This is a typical Subject Watcher class. It will observe all events.

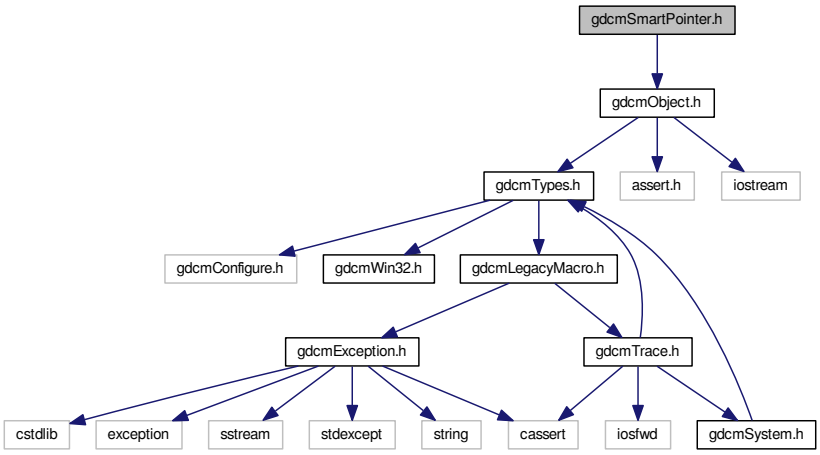
Namespaces

- gdcm

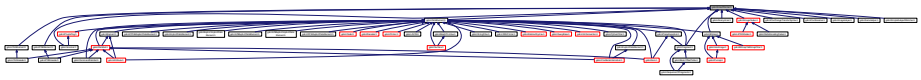
28.224 gdcmSmartPointer.h File Reference

```
#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.

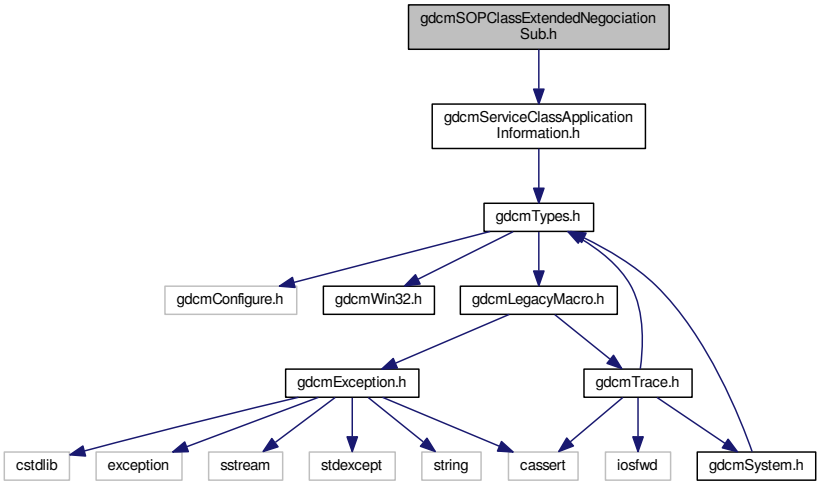
Namespaces

- gdcm

28.225 gdcmSOPClassExtendedNegociationSub.h File Reference

```
#include "gdcmServiceClassApplicationInformation.h"
```

Include dependency graph for gdcmSOPClassExtendedNegociationSub.h:



Classes

- class gdcm::network::SOPClassExtendedNegociationSub

  SOPClassExtendedNegociationSub PS 3.7 Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

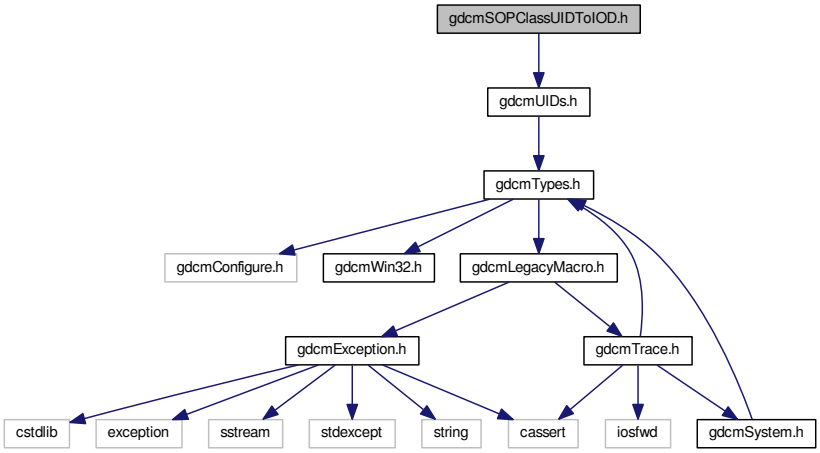
Namespaces

- gdcm
- gdcm::network

28.226 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.

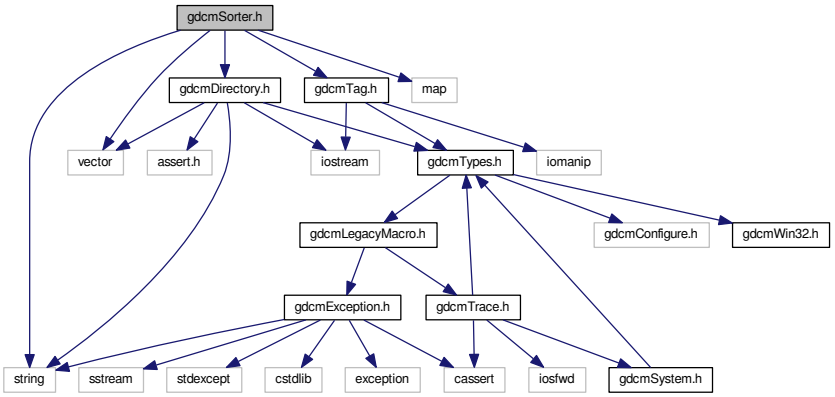
Namespaces

- gdcm

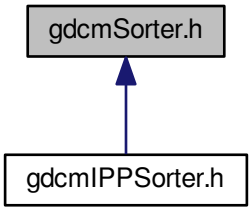
28.227 gdcmSorter.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
#include <map>
```

Include dependency graph for gdcmSorter.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gdcm::Sorter

  Sorter General class to do sorting using a custom function You simply need to provide a function of type: *Sorter::Sort↩ Function*.

Namespaces

- gdcm

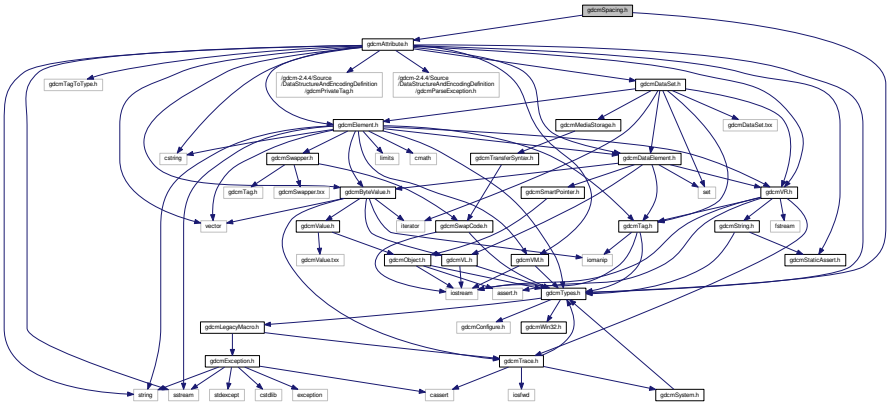
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)

28.228 gdcmSpacing.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmAttribute.h"
```

Include dependency graph for gdcmSpacing.h:



Classes

- class gdcm::Spacing

  Class for Spacing.

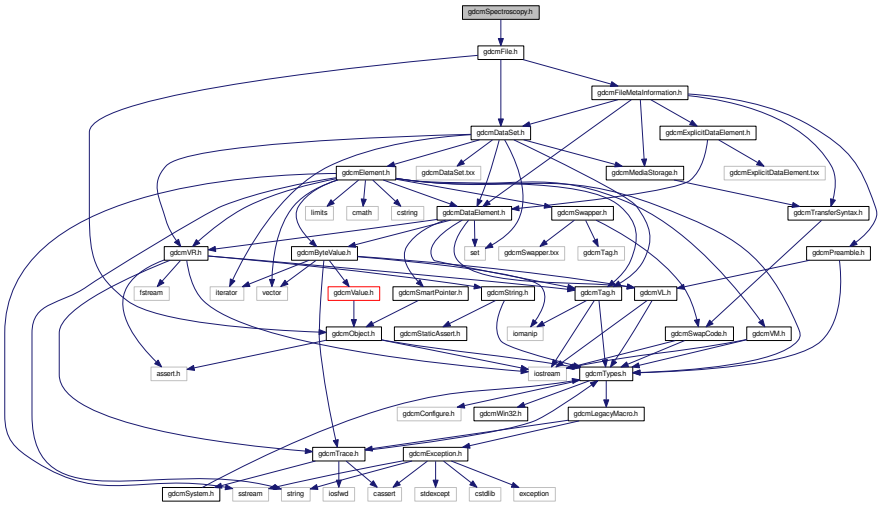
Namespaces

- gdcm

28.229 gdcmSpectroscopy.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmSpectroscopy.h:



Classes

- class gdcm::Spectroscopy

  Spectroscopy class.

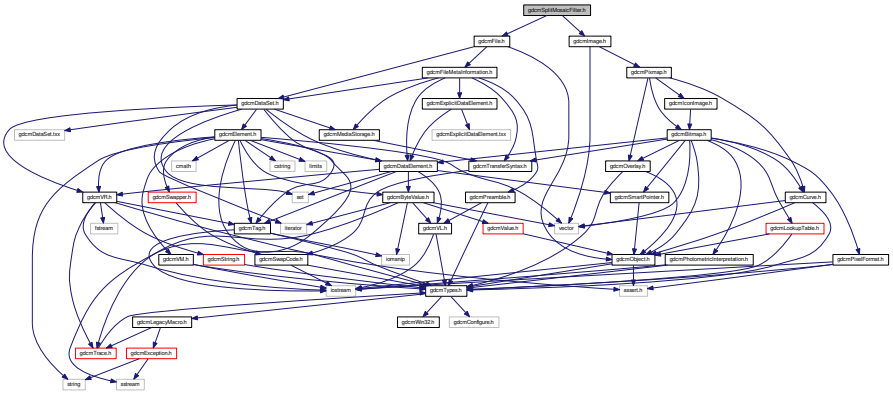
Namespaces

- gdcm

28.230 gdcmSplitMosaicFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```

Include dependency graph for gdcmSplitMosaicFilter.h:



Classes

- class gdcm::SplitMosaicFilter

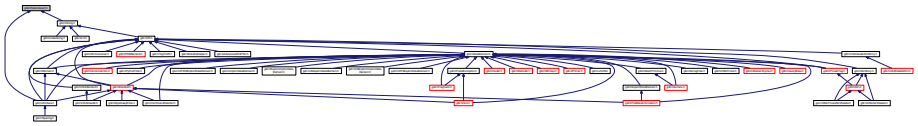
  SplitMosaicFilter class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA Image Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.

Namespaces

- gdcm

28.231 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< x >
- struct gdcm::STATIC_ASSERTION_FAILURE< true >

Namespaces

- gdcm

Macros

- #define GDCM_DO_JOIN(X, Y) GDCM_DO_JOIN2(X,Y)
- #define GDCM_DO_JOIN2(X, Y) X##Y
- #define GDCM_JOIN(X, Y) GDCM_DO_JOIN(X, Y)
- #define GDCM_STATIC_ASSERT(B)

  *The GDCM_JOIN + **LINE** is needed to create a uniq identifier.*

28.231.1 Macro Definition Documentation

28.231.1.1 #define GDCM_DO_JOIN(*X, Y*) GDCM_DO_JOIN2(X,Y)

28.231.1.2 #define GDCM_DO_JOIN2(*X, Y*) X##Y

28.231.1.3 #define GDCM_JOIN(*X, Y*) GDCM_DO_JOIN(X, Y)

28.231.1.4 #define GDCM_STATIC_ASSERT(*B*)

Value:

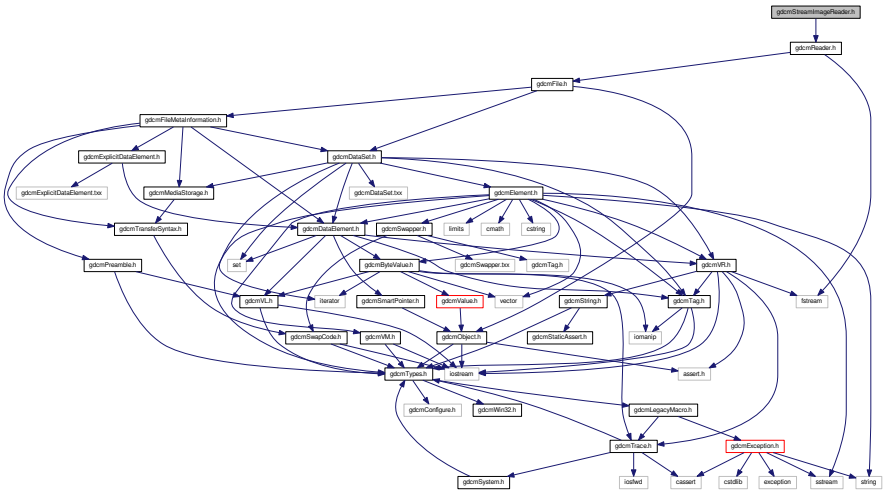
```
typedef ::gdcm::static_assert_test<\
    sizeof(::gdcm::STATIC_ASSERTION_FAILURE< (bool)( B ) >)>
      \
      GDCM_JOIN(gdcm_static_assert_typedef_, __LINE__)
```

The GDCM_JOIN + **LINE** is needed to create a uniq identifier.

28.232 gdcmStreamImageReader.h File Reference

```
#include "gdcmReader.h"
```

Include dependency graph for gdcmStreamImageReader.h:



Classes

- class gdcm::StreamImageReader

  StreamImageReader.

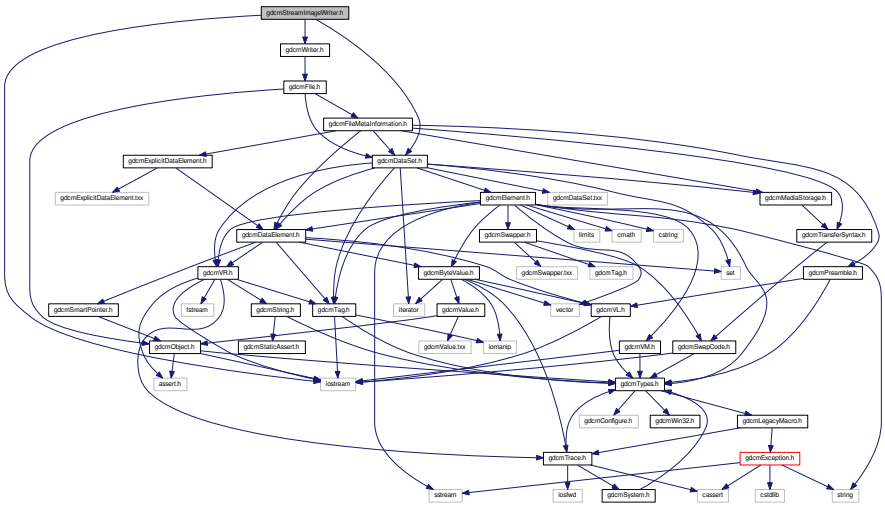
Namespaces

- gdcm

28.233 gdcmStreamImageWriter.h File Reference

```
#include "gdcmWriter.h"
#include <iostream>
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmStreamImageWriter.h:



Classes

- class gdcm::StreamImageWriter

  StreamImageReader.

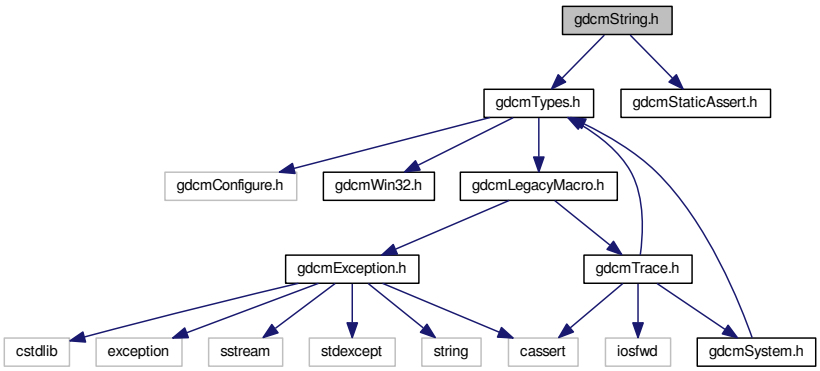
Namespaces

- gdcm

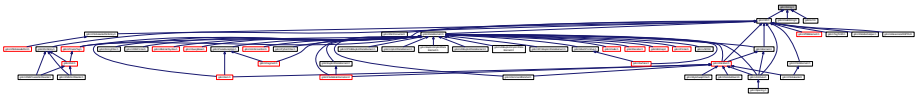
28.234 gdcmString.h File Reference

```
#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

- gdcm

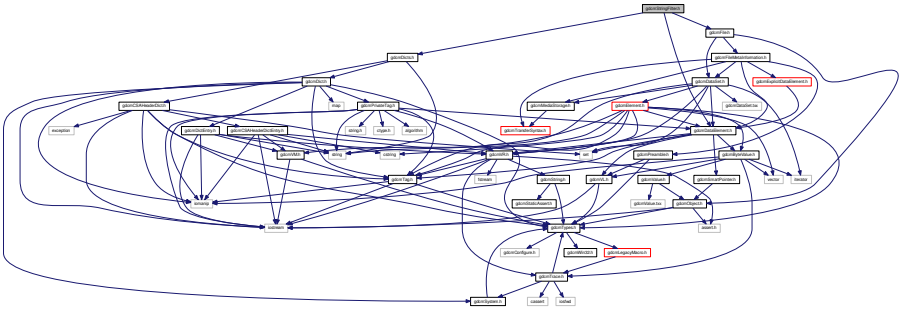
Functions

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

28.235 gdcmStringFilter.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"
```

Include dependency graph for gdcmStringFilter.h:



Classes

- class gdcm::StringFilter

  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.

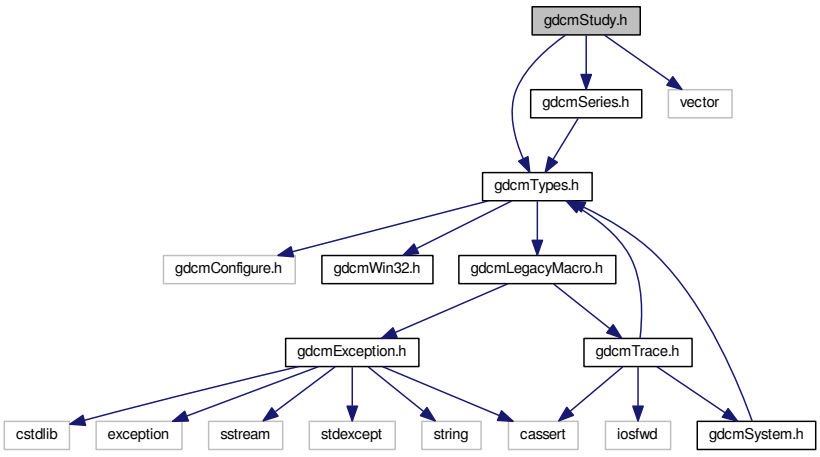
Namespaces

- gdcm

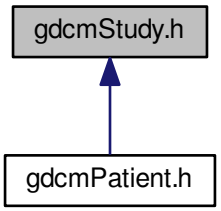
28.236 gdcmStudy.h File Reference

```
#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.

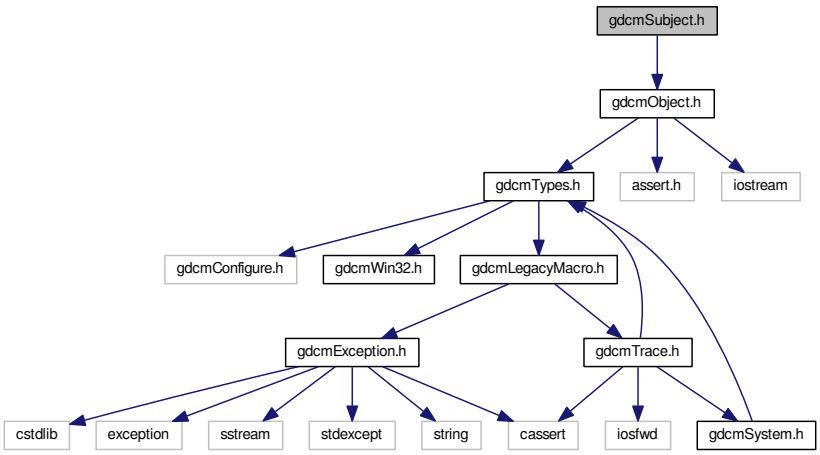
Namespaces

- gdcm

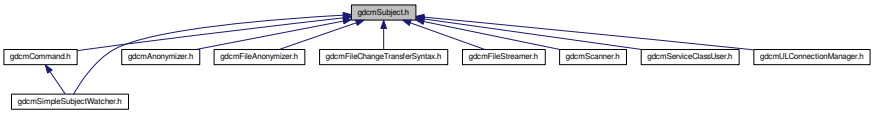
28.237 gdcmSubject.h File Reference

```
#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.

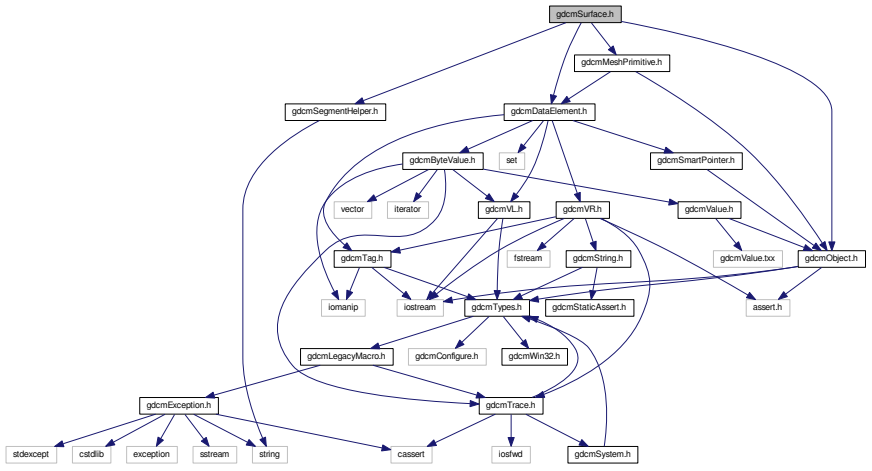
Namespaces

- gdcm

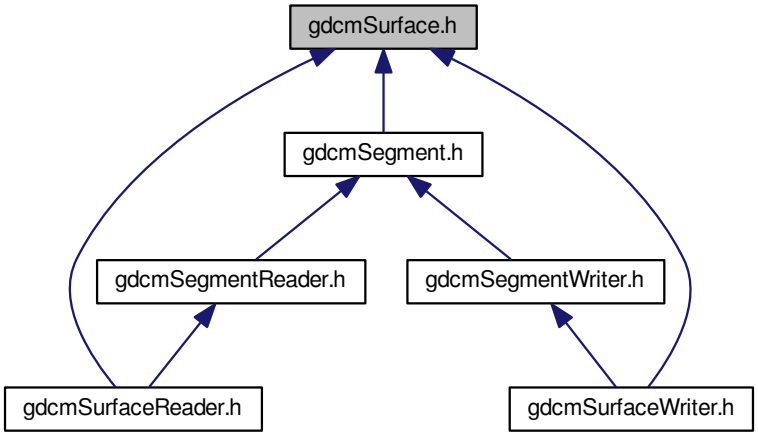
28.238 gdcmSurface.h File Reference

```
#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. This members are taken from required surface mesh module attributes.

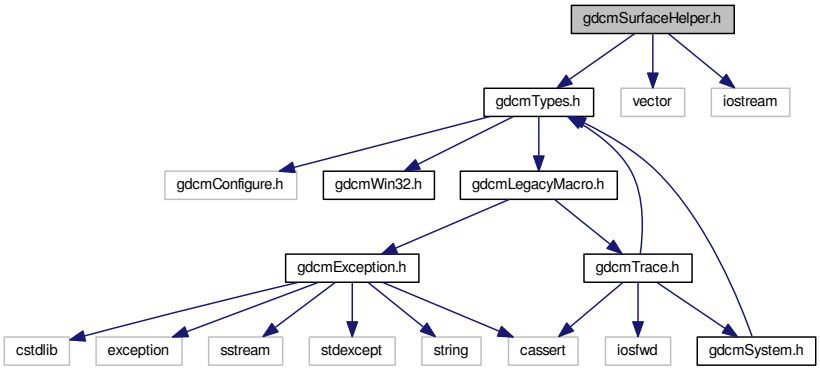
Namespaces

- gdcm

28.239 gdcmSurfaceHelper.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <iostream>
```

Include dependency graph for gdcmSurfaceHelper.h:



Classes

- class gdcm::SurfaceHelper

  SurfaceHelper Helper class for Surface object.

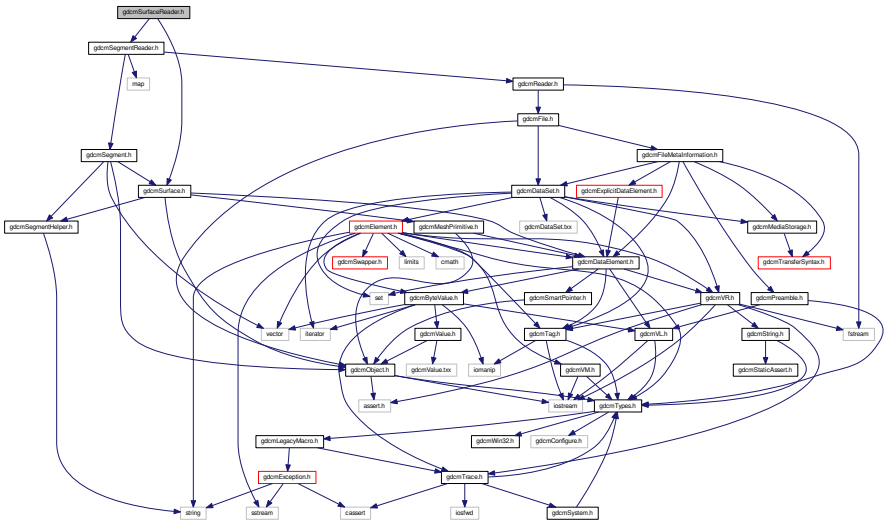
Namespaces

- gdcm

28.240 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. It reads surface mesh module attributes.

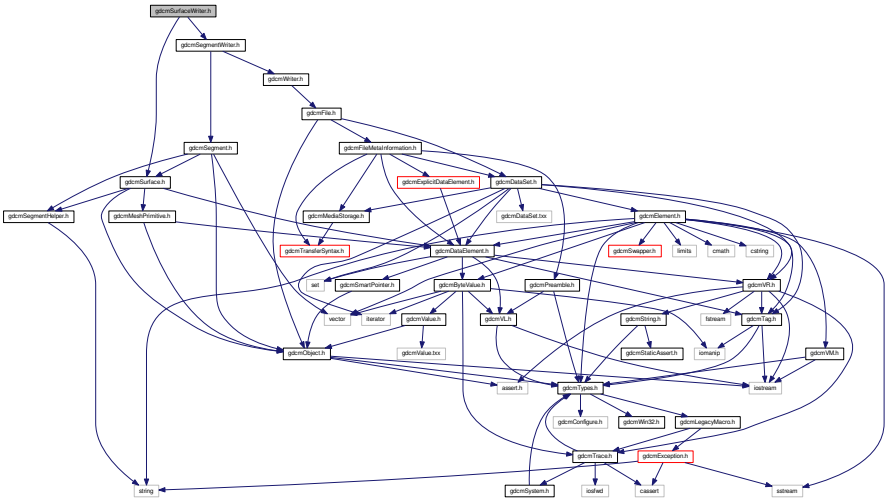
Namespaces

- gdcm

28.241 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. It writes surface mesh module attributes.

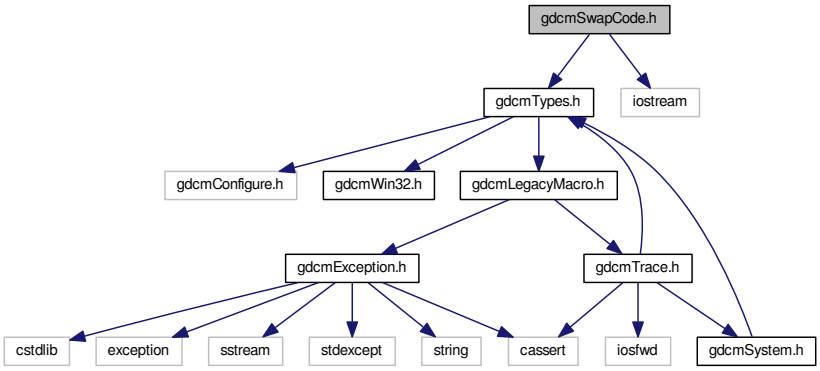
Namespaces

- gdcm

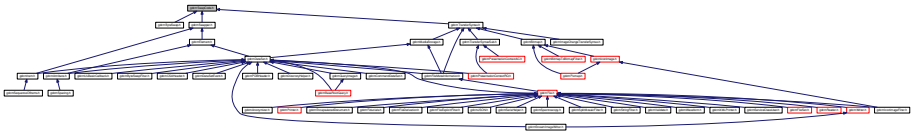
28.242 gdcmSwapCode.h File Reference

```
#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

- gdcm

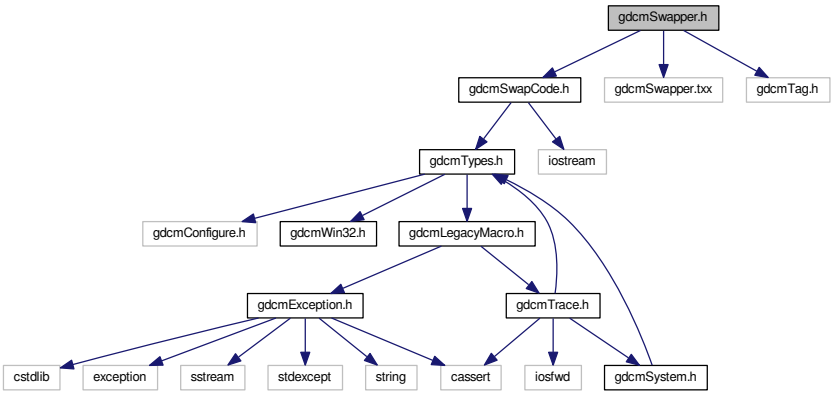
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)

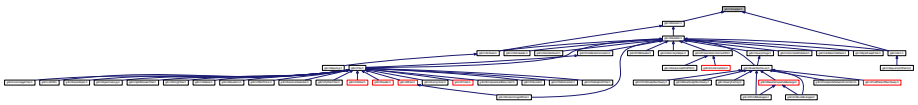
28.243 gdcmSwapper.h File Reference

```
#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

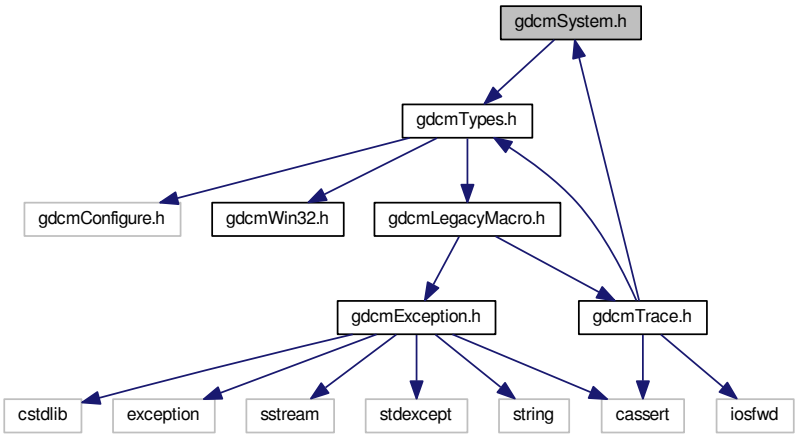
Namespaces

- gdcm

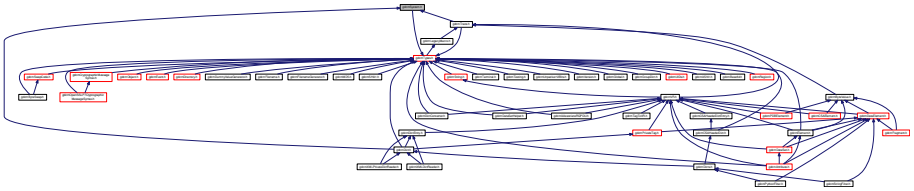
28.244 gdcmSystem.h File Reference

```
#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.

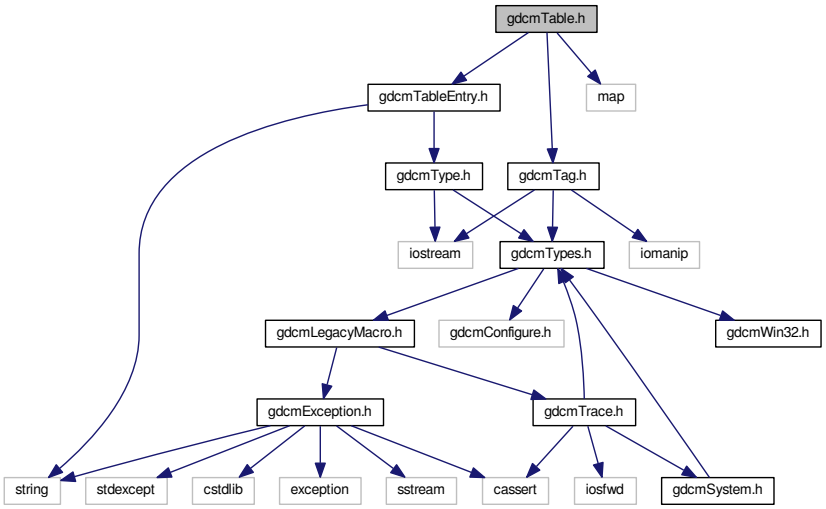
Namespaces

- gdcm

28.245 gdcmTable.h File Reference

```
#include "gdcmTableEntry.h"
#include "gdcmTag.h"
#include <map>
```

Include dependency graph for gdcmTable.h:



Classes

- class gdcm::Table

  Table.

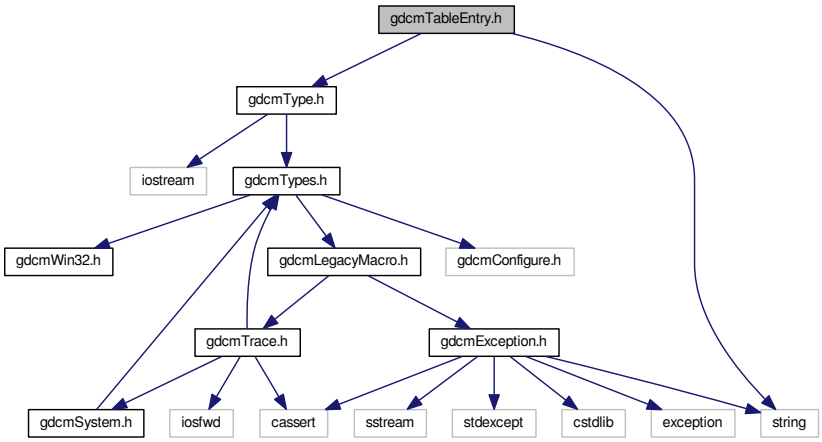
Namespaces

- gdcm

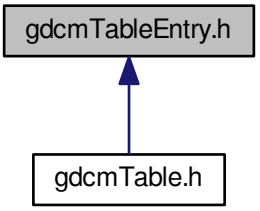
28.246 gdcmTableEntry.h File Reference

```
#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

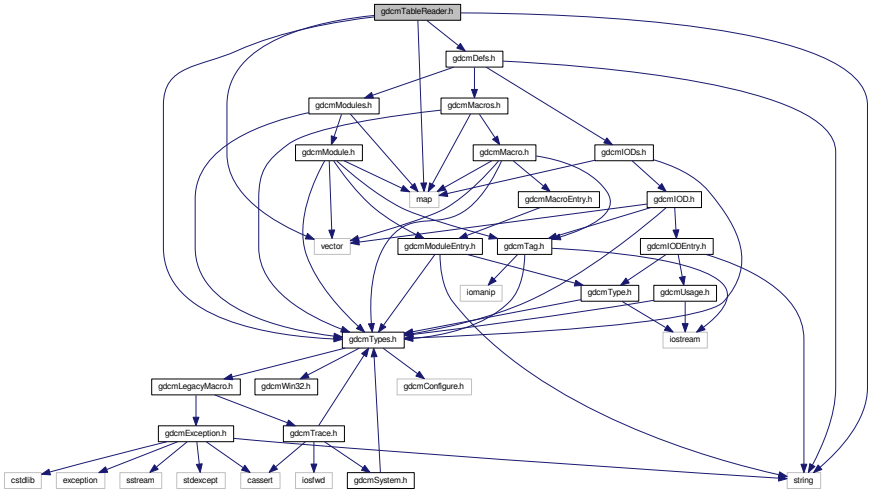
- gdcm

28.247 gdcmTableReader.h File Reference

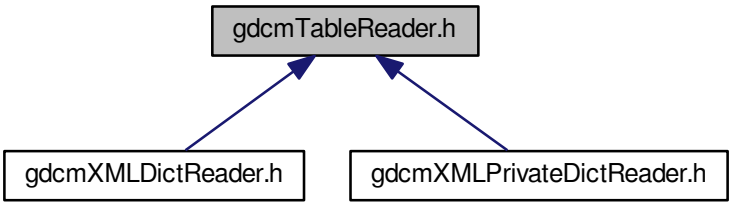
```
#include "gdcmTypes.h"
```

```
#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.

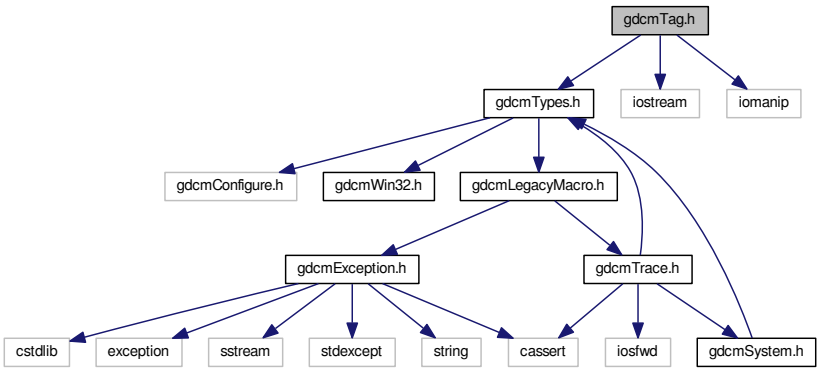
Namespaces

- gdcm

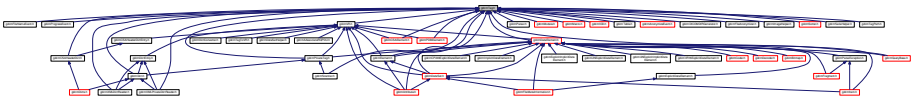
28.248 gdcmTag.h File Reference

```
#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). Basically an uint32_t which can also be expressed as two uint16_t (group and element)

Namespaces

- gdcm

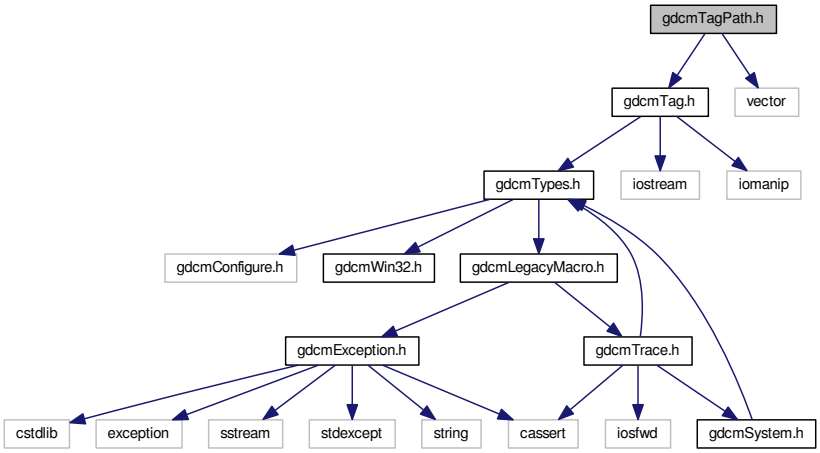
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)
- std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)

28.249 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.

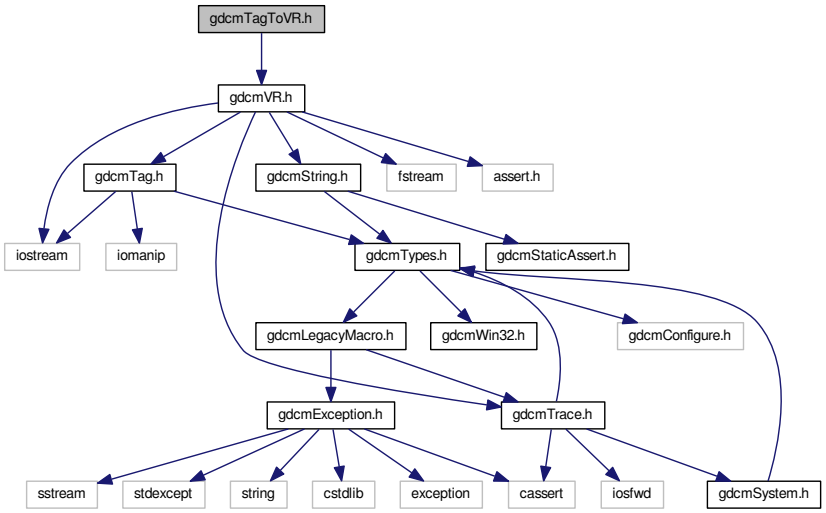
Namespaces

- gdcm

28.250 gdcmTagToVR.h File Reference

```
#include "gdcmVR.h"
```

Include dependency graph for gdcmTagToVR.h:



Namespaces

- gdcm

Functions

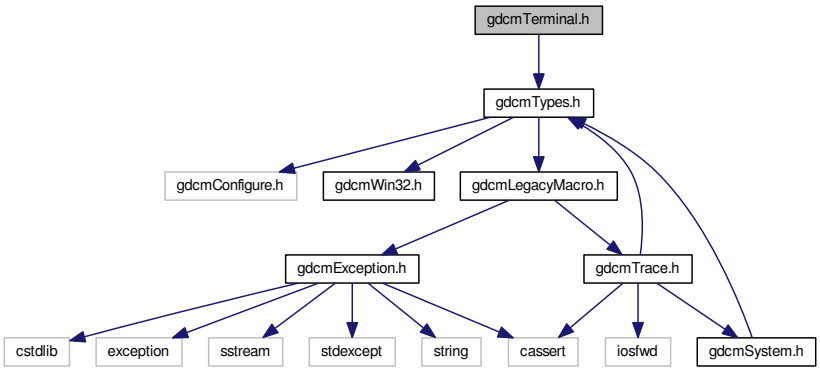
- VR::VRType gdcm::GetVRFromTag (Tag const &tag)

28.251 gdcmtar.dox File Reference

28.252 gdcmTerminal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmTerminal.h:



Namespaces

- gdcm
- gdcm::terminal

  Class for Terminal Allow one to print in color in a shell.

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

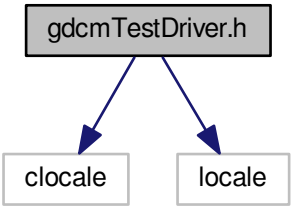
- GDCM_EXPORT std::string gdcm::terminal::setattribute (Attribute att)

- 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)

28.253 gdcmTestDriver.h File Reference

```
#include <clocale>
#include <locale>
```

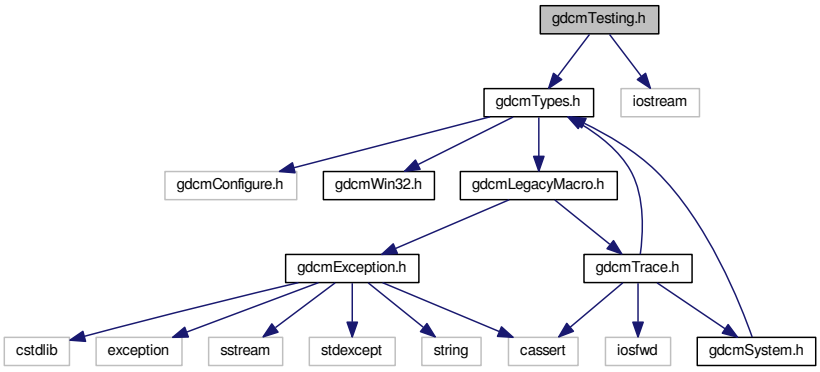
Include dependency graph for gdcmTestDriver.h:



28.254 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



Classes

- class gdcm::Testing

  class for testing

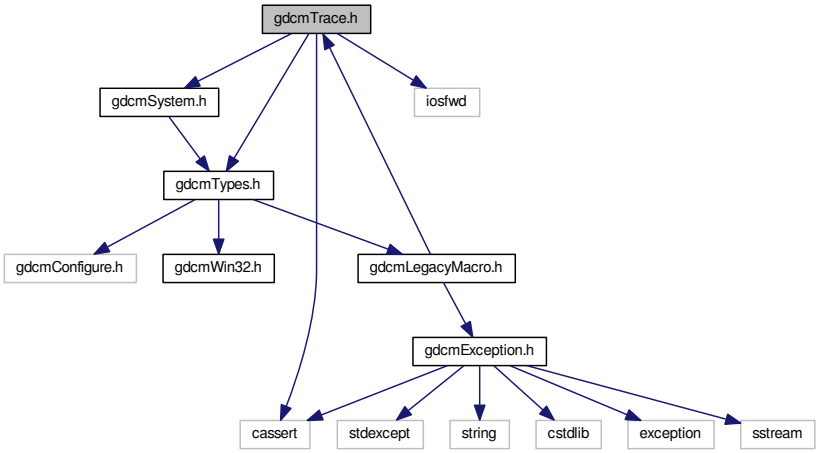
Namespaces

- gdcm

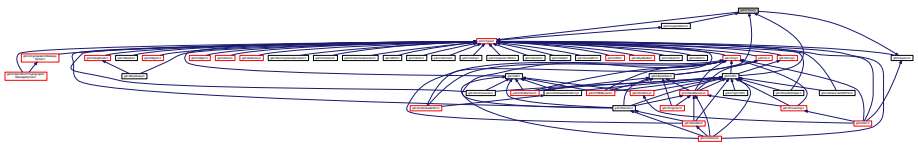
28.255 gdcmTrace.h File Reference

```
#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

- gdcm

Macros

- #define GDCM_FUNCTION "<unknow>"
- #define gdcmAssertAlwaysMacro(arg) gdcmAssertMacro(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.

28.255.1 Macro Definition Documentation

28.255.1.1 #define GDCM_FUNCTION "<unknow>"

28.255.1.2 #define gdcmAssertAlwaysMacro(*arg*) 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::SequenceOfFragments::ReadValue(), and gdcm::VR::Write().

28.255.1.3 #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 );                                                   \
   }                                                                 \
}
```

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().

28.255.1.4 #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;                         \
   }                                                                    \
}
```

Debug.

Parameters

msg	message part

Referenced by gdcm::ByteValue::ByteValue(), gdcm::OpenSSLCryptoFactory::OpenSSLCryptoFactory(), gdcm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory(), gdcm::BasicOffsetTable::Read(), gdcm::Item::Read(), gdcm::SequenceOfItems::Read(), gdcm::VR::Read(), gdcm::SequenceOfFragments::ReadPreValue(), gdcm::SequenceOfFragments::ReadValue(), and gdcm::ByteValue::SetLength().

28.255.1.5 #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;                                   \
   }                                                                    \
}
```

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::FileMetaInformation::Insert(), gdcm::DataSet::Insert(), gdcm::Item::Read(), and gdcm::Fragment::ReadBacktrack().

28.255.1.6 #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;                                \
   }                                                                 \
}
```

Warning.

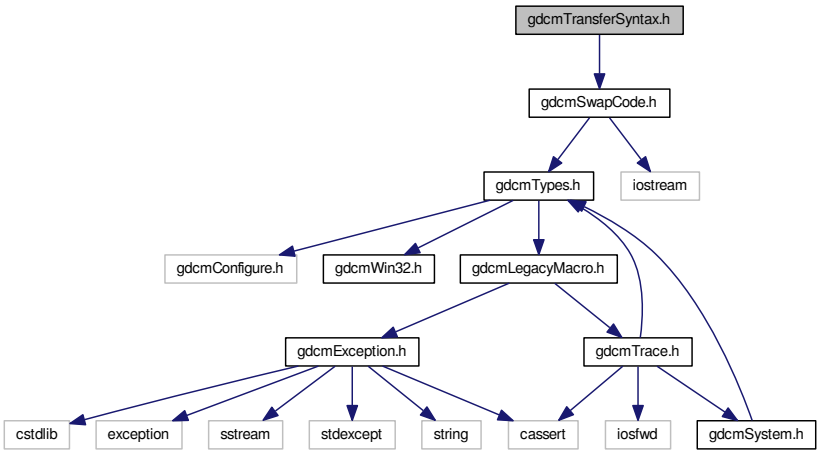
Parameters

msg	message part

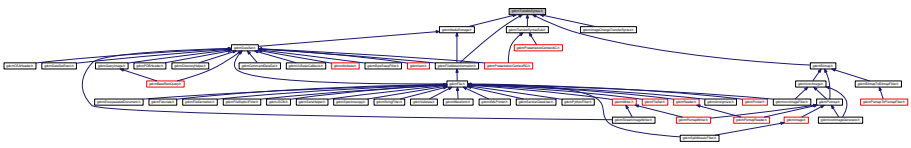
Referenced by gdcm::DataSet::InsertDataElement(), gdcm::Item::Read(), gdcm::SequenceOfItems::Read(), gdcm::Fragment::ReadBacktrack(), gdcm::Fragment::ReadValue(), gdcm::SequenceOfFragments::ReadValue(), gdcm::OpenSSLP7CryptographicMessageSyntax::SetPassword(), and gdcm::Item::Write().

28.256 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

- gdcm

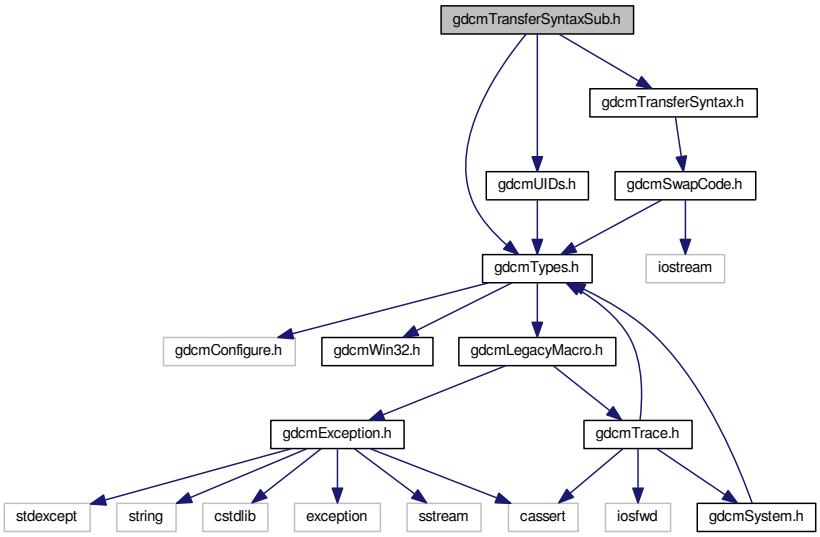
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)

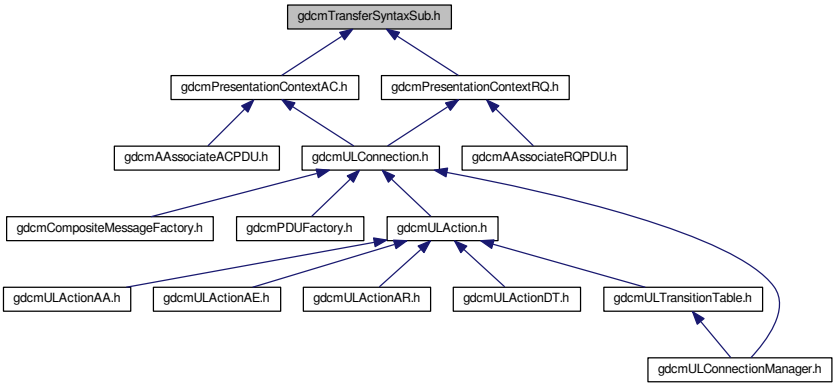
28.257 gdcmTransferSyntaxSub.h File Reference

```
#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 Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

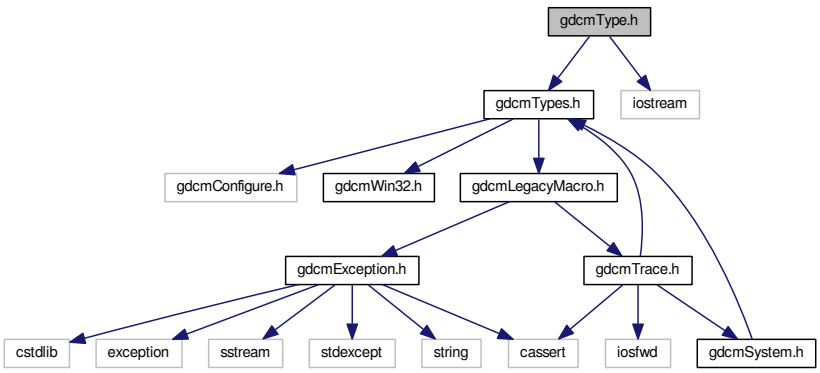
Namespaces

- gdcm
- gdcm::network

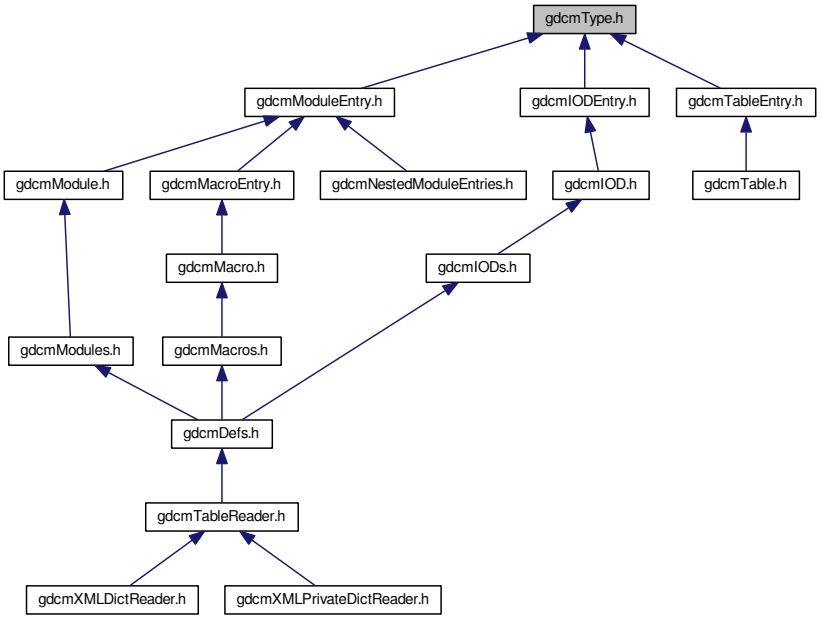
28.258 gdcmType.h File Reference

```
#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

- gdcm

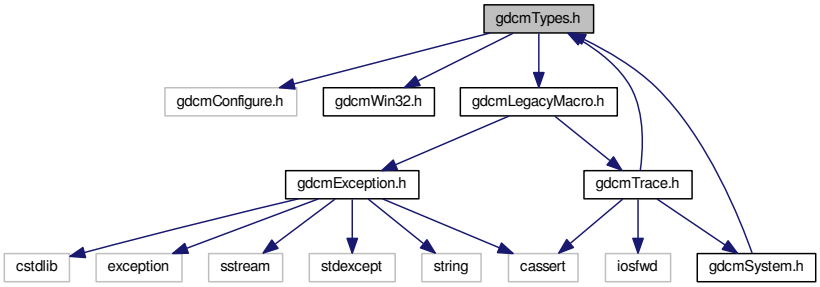
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)

28.259 gdcmTypes.h File Reference

```
#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
```

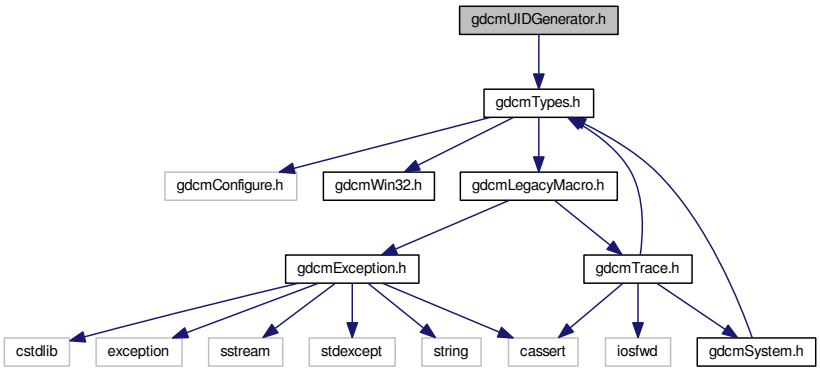
Include dependency graph for gdcmTypes.h:



28.260 gdcmUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUIDGenerator.h:



Classes

- class gdcm::UIDGenerator

  Class for generating unique UID.

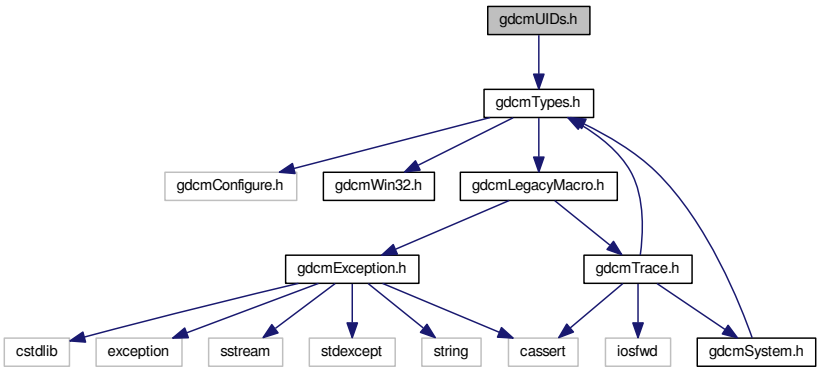
Namespaces

- gdcm

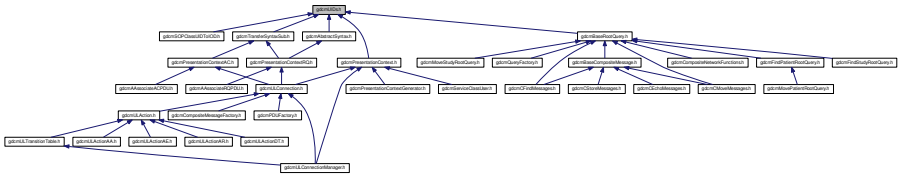
28.261 gdcmUIDs.h File Reference

```
#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

- gdcm

Functions

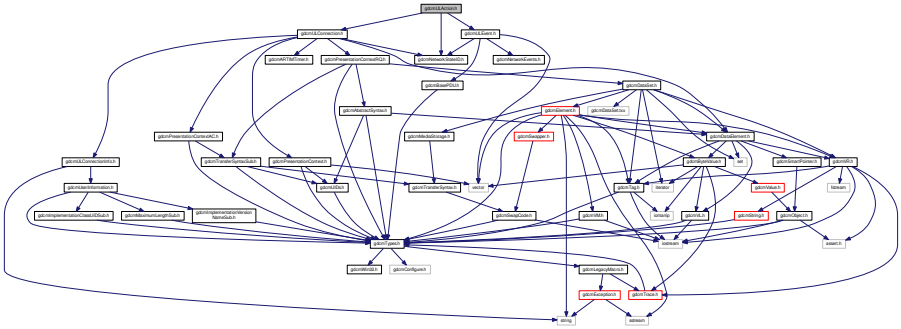
- std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)

28.262 gdcmULAction.h File Reference

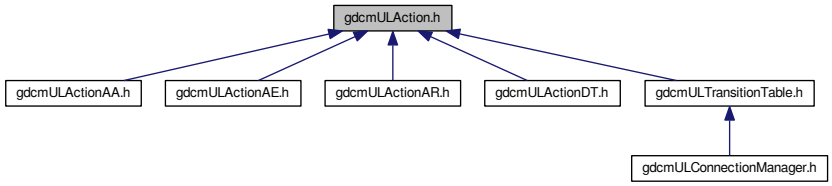
```
#include "gdcmNetworkStateID.h"
```

```
#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 A ULConnection in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given ULConnection.

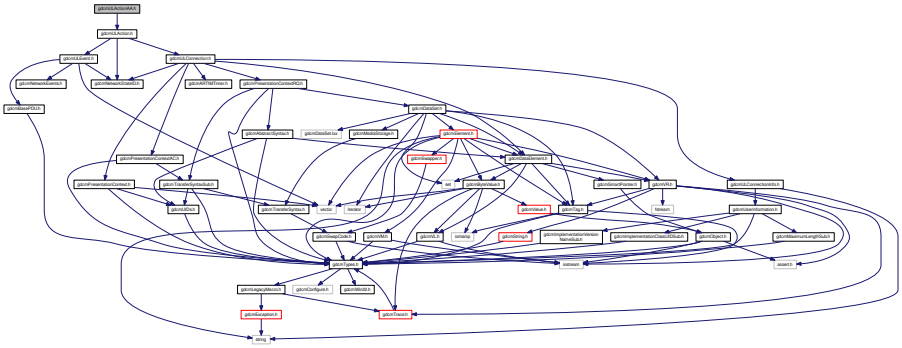
Namespaces

- gdcm
- gdcm::network

28.263 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

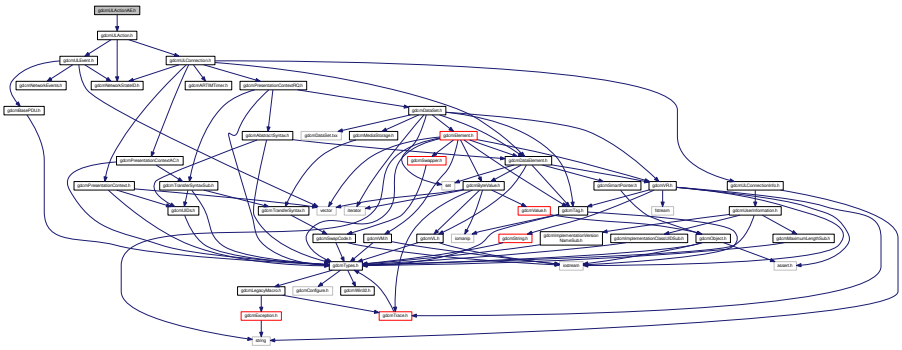
Namespaces

- gdcm
- gdcm::network

28.264 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

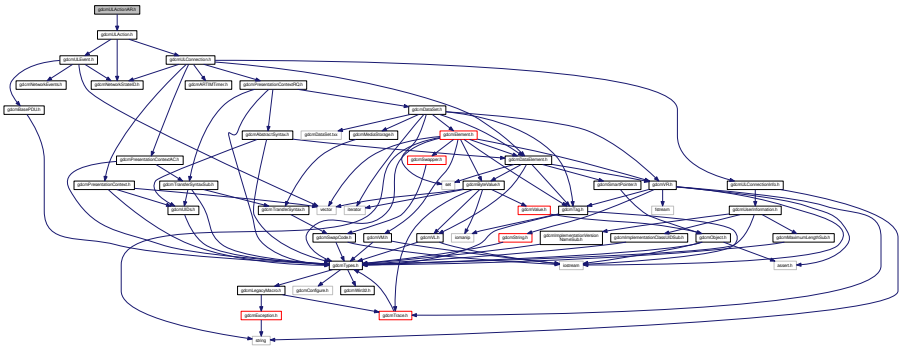
Namespaces

- gdcm
- gdcm::network

28.265 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

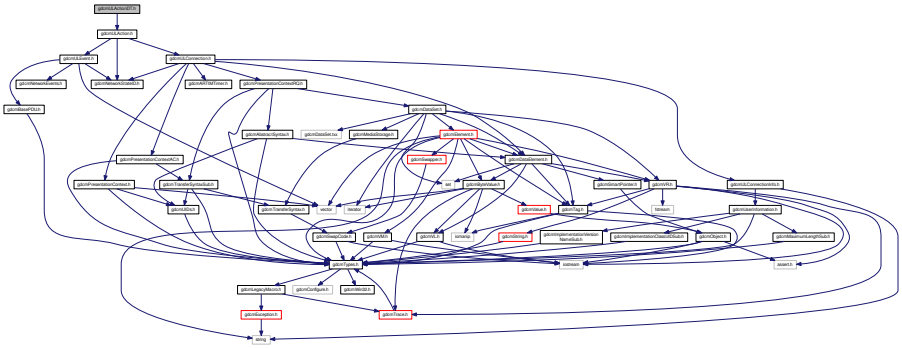
Namespaces

- gdcm
- gdcm::network

28.266 gdcmULActionDT.h File Reference

```
#include "gdcmULAction.h"
```

Include dependency graph for gdcmULActionDT.h:



Classes

- class gdcm::network::ULActionDT1
- class gdcm::network::ULActionDT2

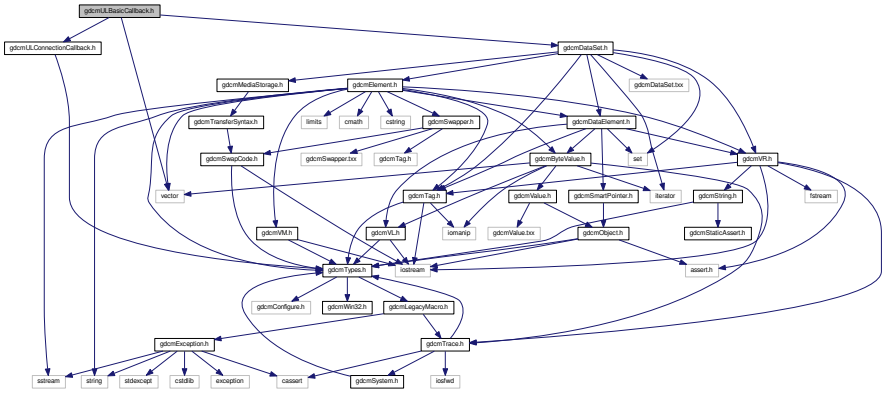
Namespaces

- gdcm
- gdcm::network

28.267 gdcmULBasicCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
#include "gdcmDataSet.h"
#include <vector>
```

Include dependency graph for gdcmULBasicCallback.h:



Classes

- class gdcm::network::ULBasicCallback

  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.

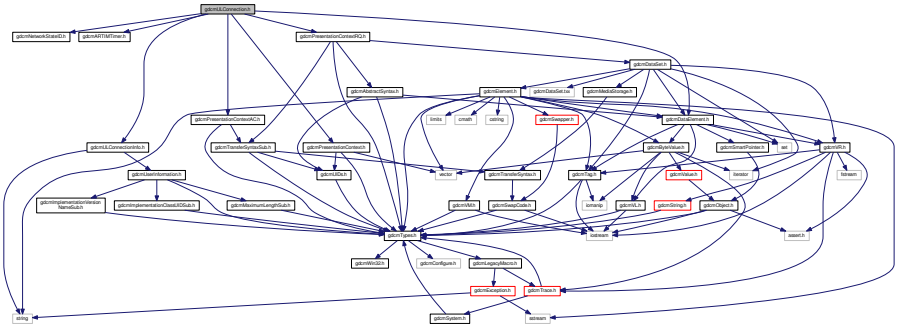
Namespaces

- gdcm
- gdcm::network

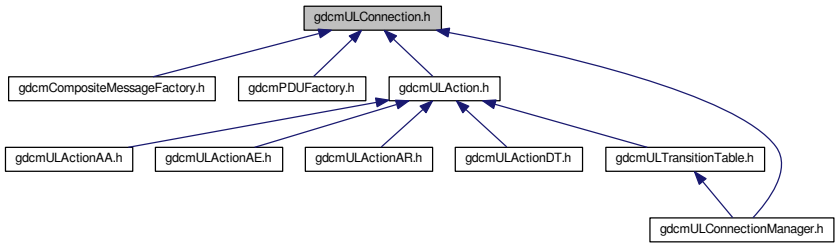
28.268 gdcmULConnection.h File Reference

```
#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 This is the class that contains the socket to another machine, and passes data through itself, as well as maintaining a sense of state.

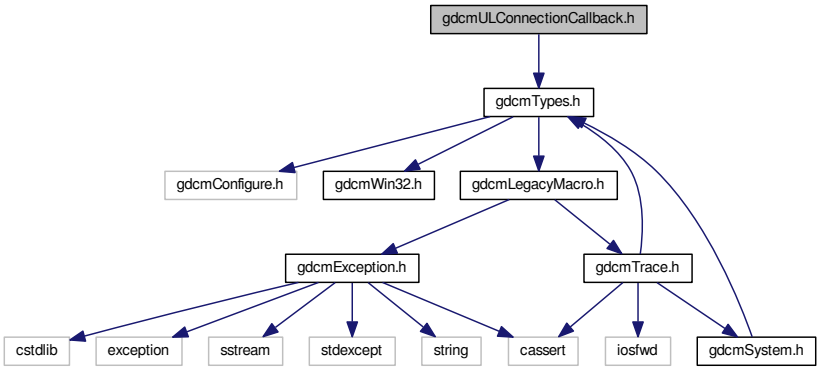
Namespaces

- gdcm
- gdcm::network

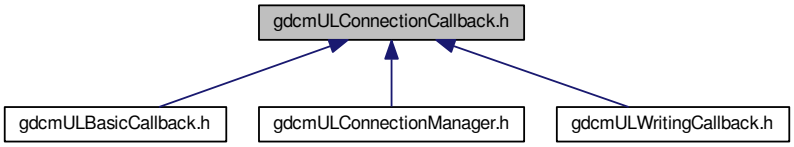
28.269 gdcmULConnectionCallback.h File Reference

```
#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

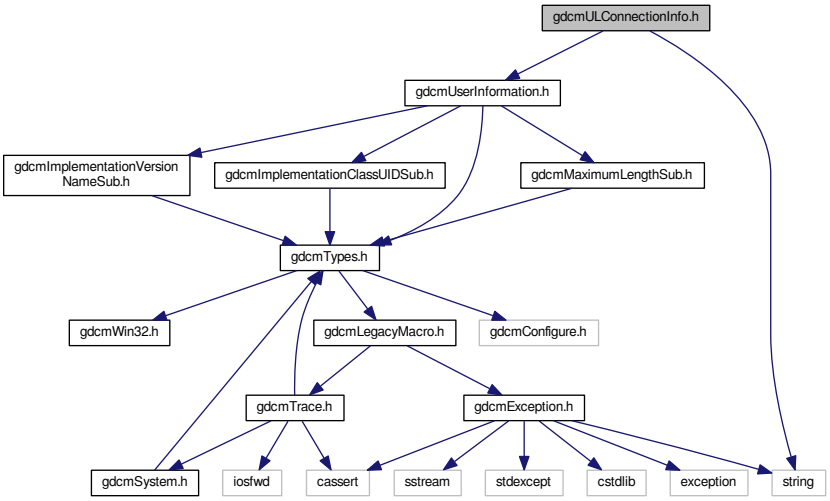
Namespaces

- gdcm
- gdcm::network

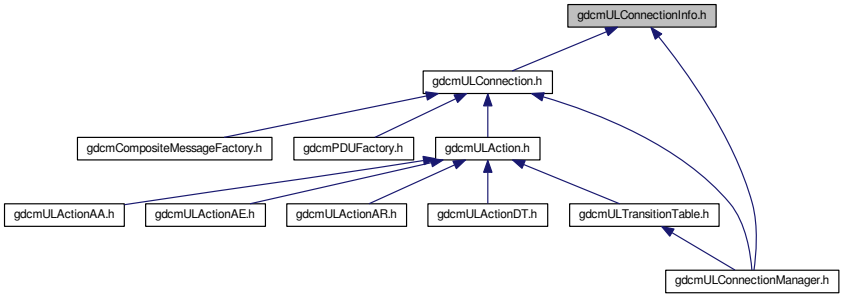
28.270 gdcmULConnectionInfo.h File Reference

```
#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 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.

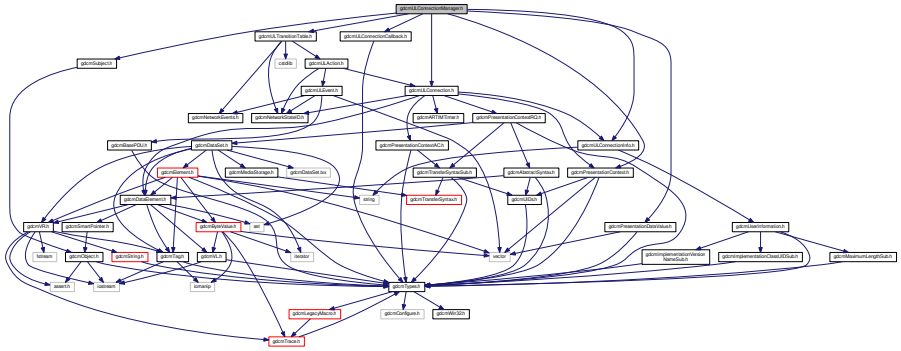
Namespaces

- gdcm
- gdcm::network

28.271 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 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).

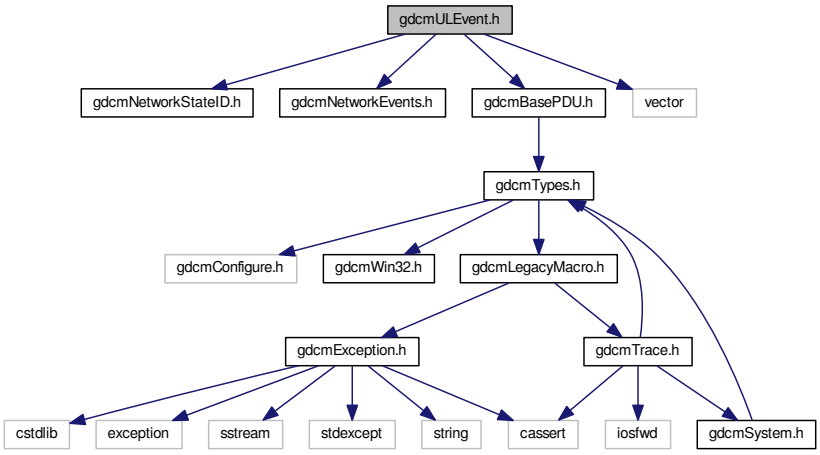
Namespaces

- gdcm
- gdcm::network

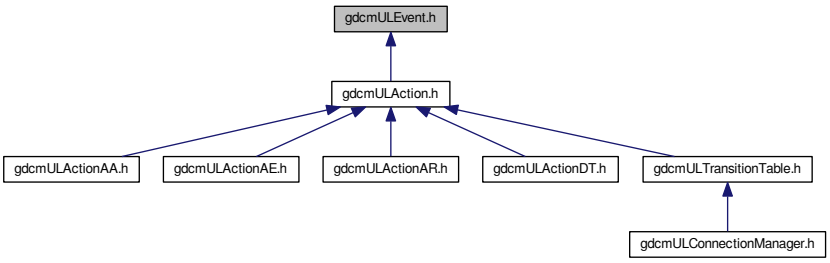
28.272 gdcmULEvent.h File Reference

```
#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 base class for network events.

Namespaces

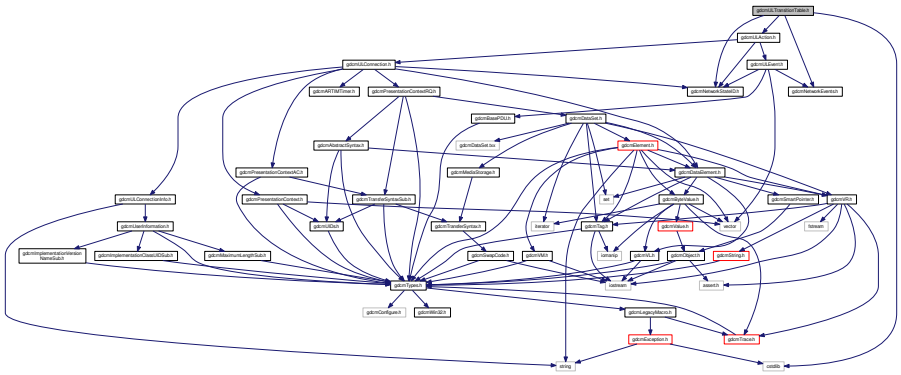
- gdcm
- gdcm::network

28.273 gdcmULTransitionTable.h File Reference

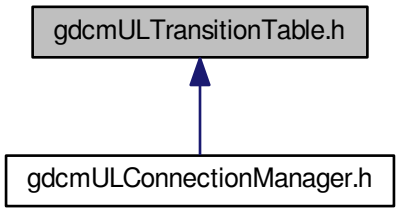
```
#include "gdcmNetworkStateID.h"
```

```
#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.

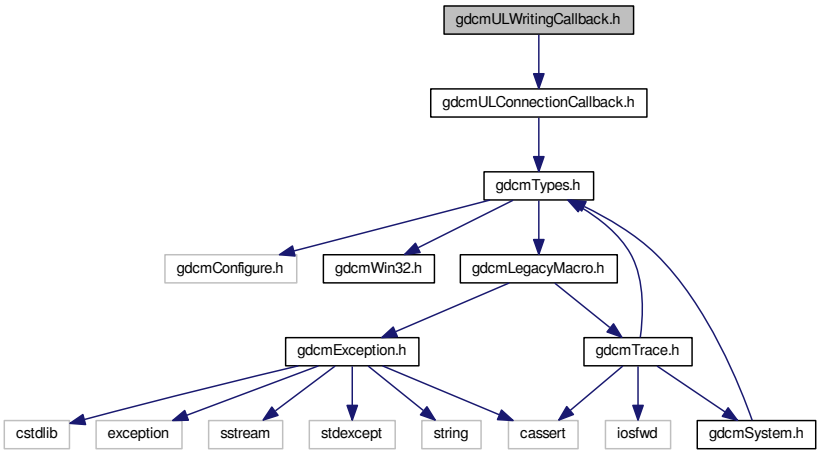
Namespaces

- gdcm
- gdcm::network

28.274 gdcmULWritingCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
```

Include dependency graph for gdcmULWritingCallback.h:



Classes

- class gdcm::network::ULWritingCallback

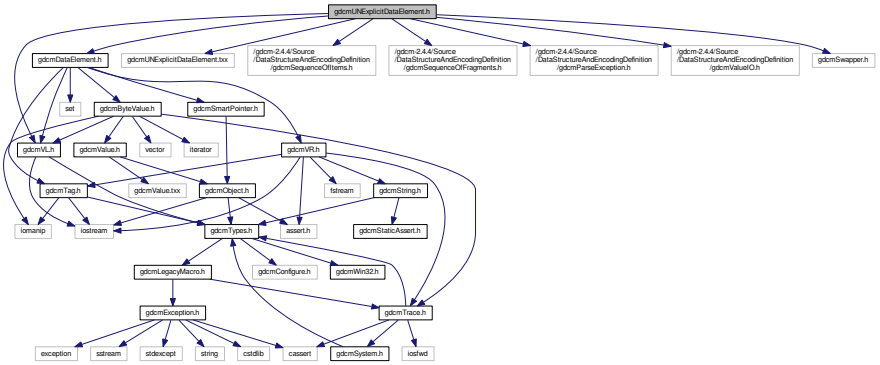
Namespaces

- gdcm
- gdcm::network

28.275 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.

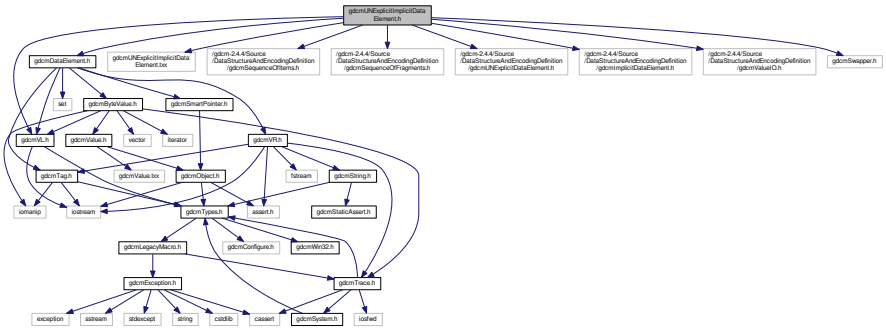
Namespaces

- gdcm

28.276 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 This class gather two known bugs:

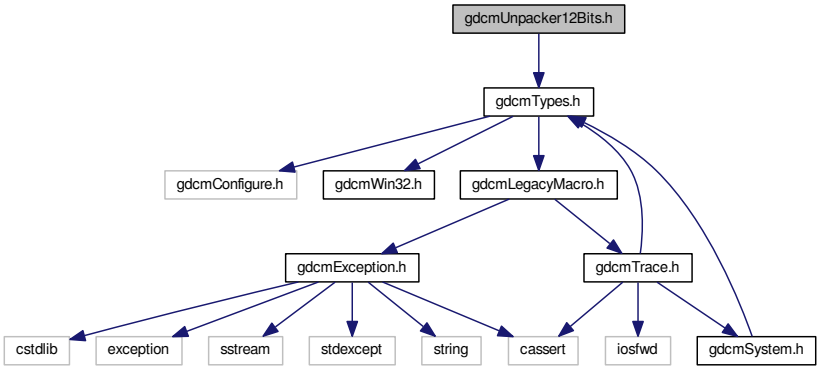
Namespaces

- gdcm

28.277 gdcmUnpacker12Bits.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUnpacker12Bits.h:



Classes

- class gdcm::Unpacker12Bits

  Pack/Unpack 12 bits pixel into 16bits.

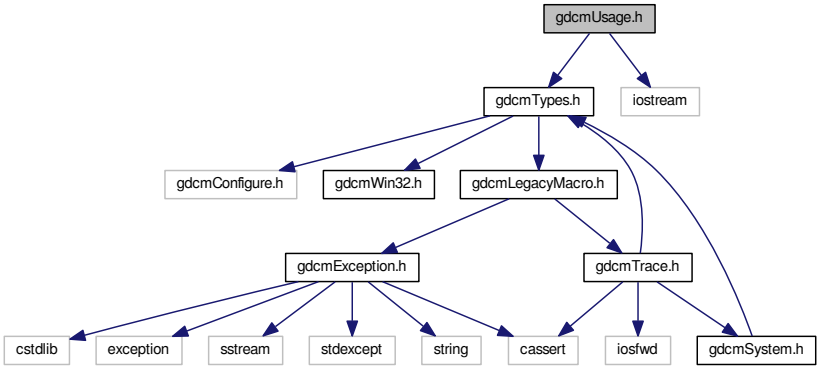
Namespaces

- gdcm

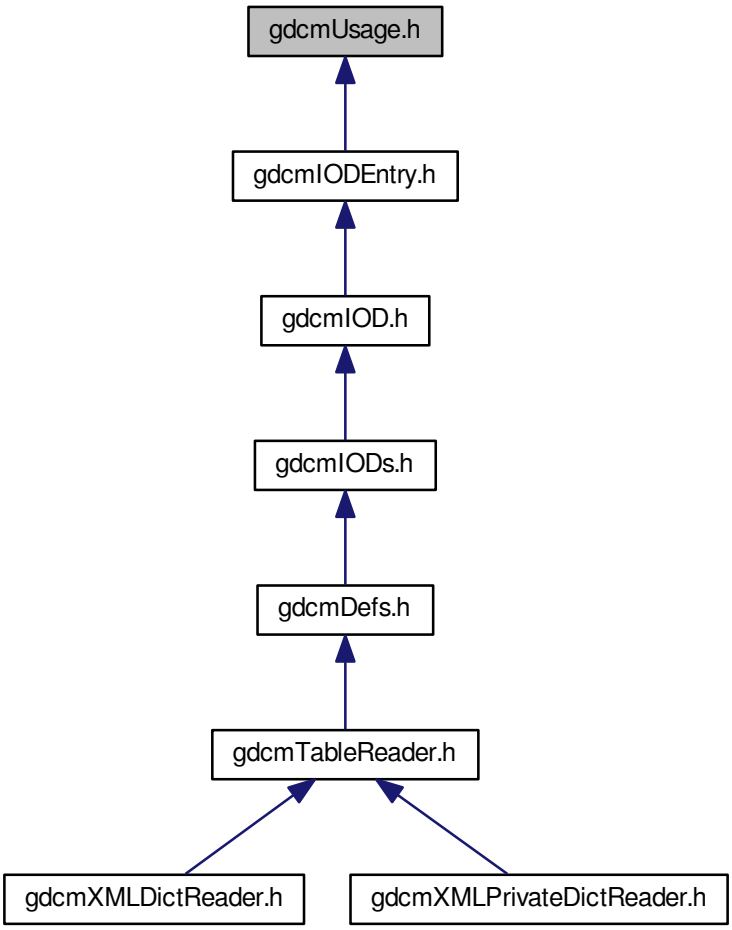
28.278 gdcmUsage.h File Reference

```
#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

- gdcm

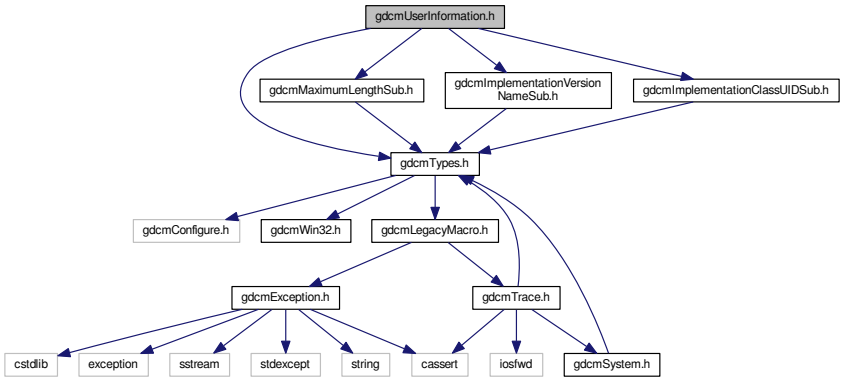
Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)

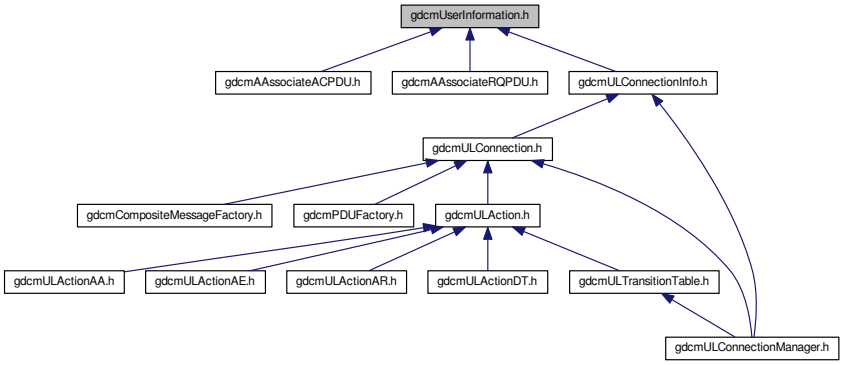
28.279 gdcmUserInformation.h File Reference

```
#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 Table 9-16 USER INFORMATION ITEM FIELDS.

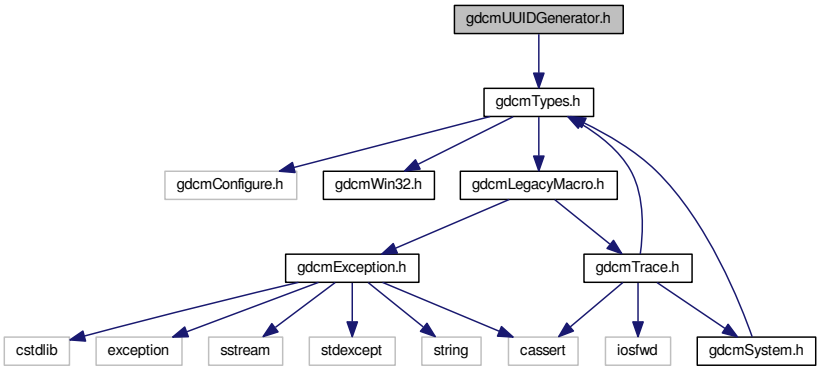
Namespaces

- gdcm
- gdcm::network

28.280 gdcmUUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUUIDGenerator.h:



Classes

- class gdcm::UUIDGenerator

  Class for generating unique UUID generate DCE 1.1 uid.

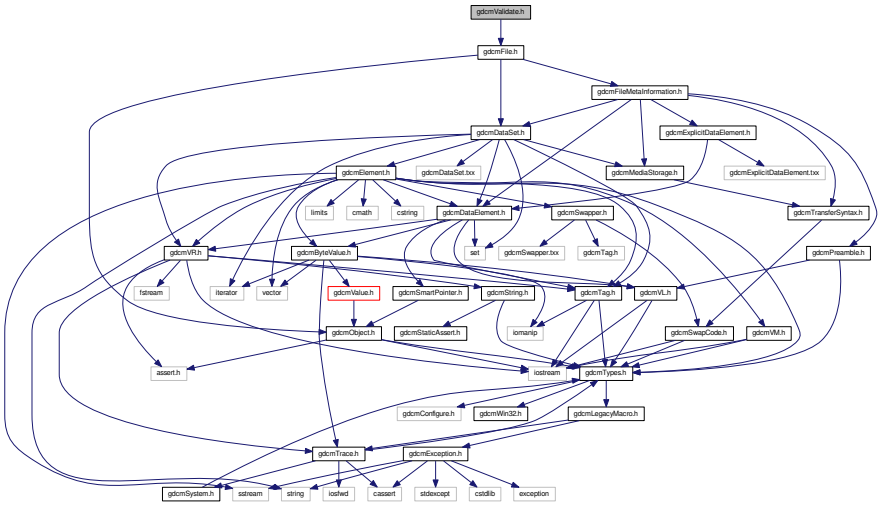
Namespaces

- gdcm

28.281 gdcmValidate.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmValidate.h:



Classes

- class gdcm::Validate

  Validate class.

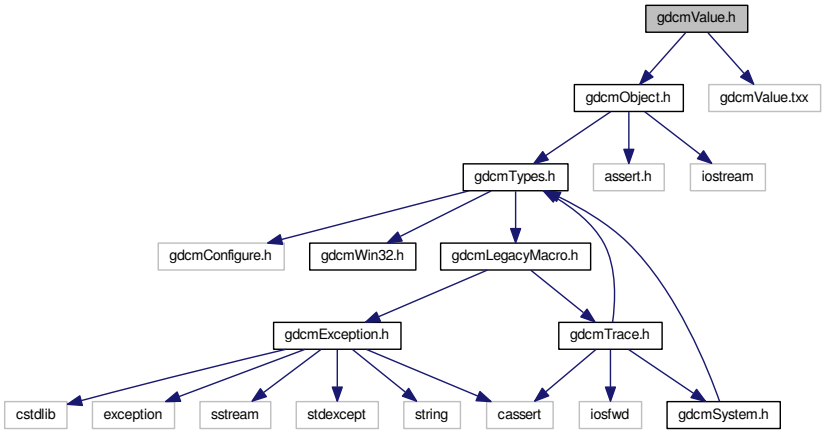
Namespaces

- gdcm

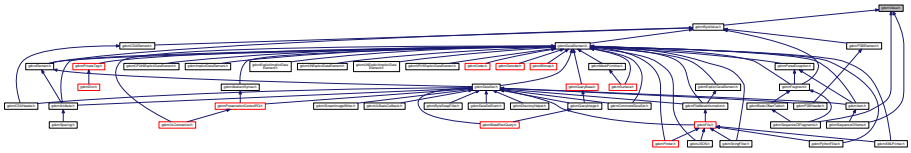
28.282 gdcmValue.h File Reference

```
#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.

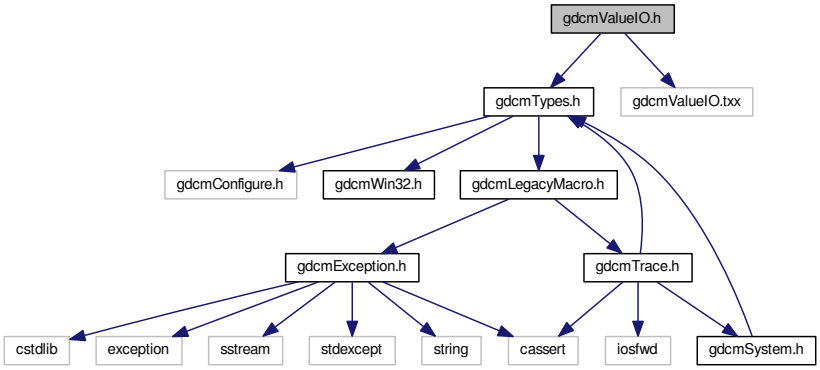
Namespaces

- gdcm

28.283 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.

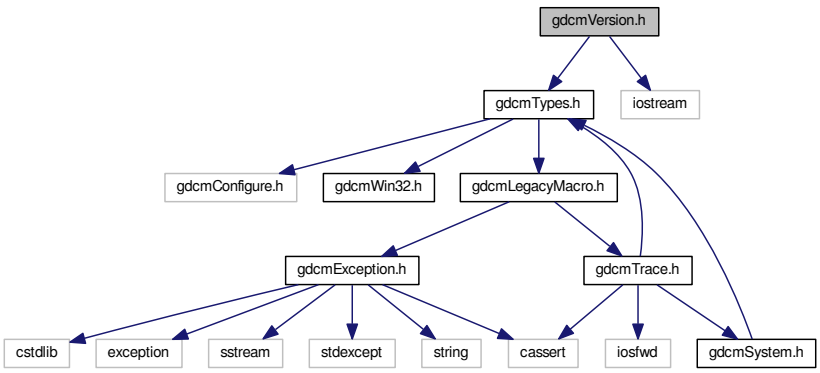
Namespaces

- gdcm

28.284 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

- gdcm

Functions

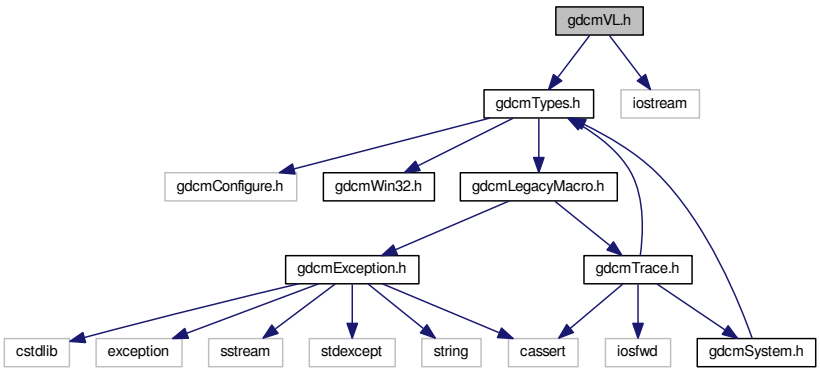
- std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)

28.285 gdcmviewer.dox File Reference

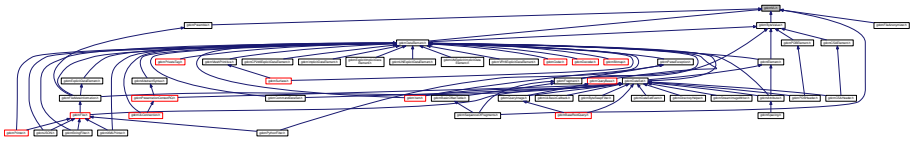
28.286 gdcmVL.h File Reference

```
#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

- gdcm

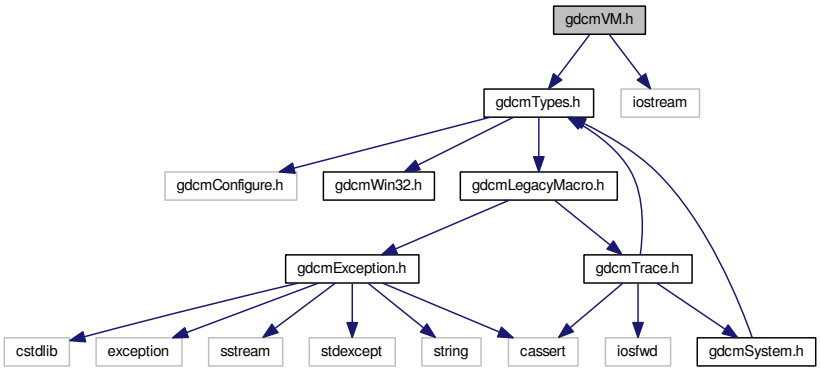
Functions

- std::ostream & gdcm::operator<< (std::ostream &os, const VL &val)

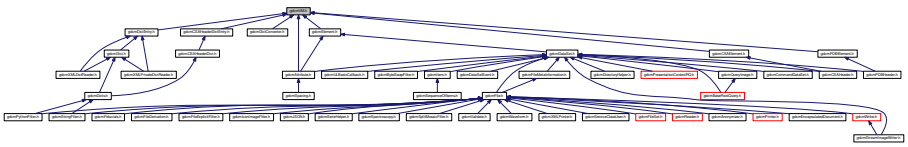
28.287 gdcmVM.h File Reference

```
#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.

- struct gdcm::VMToLength< T >

Namespaces

- gdcm

Macros

- #define TYPETOLENGTH(type, length)

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VM &_val)

28.287.1 Macro Definition Documentation

28.287.1.1 #define TYPETOLENGTH(*type, length*)

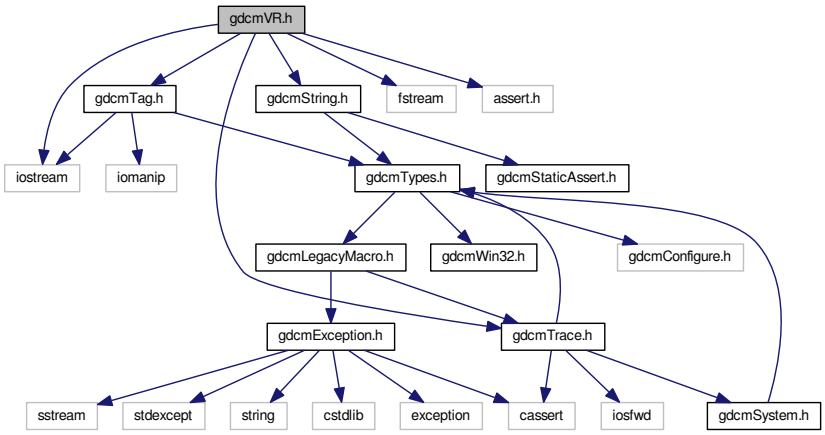
Value:

```
template<> struct VMToLength<VM::type> \
  { enum { Length = length }; };
```

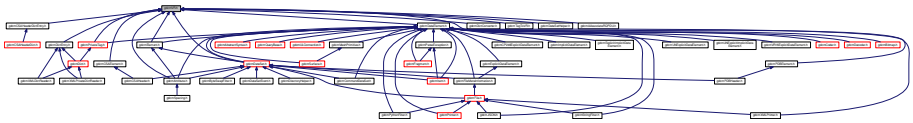
28.288 gdcmVR.h File Reference

```
#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 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.
- struct gdcm::VRToEncoding< T >
- struct gdcm::VRToType< T >

Namespaces

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

Functions

- std::ostream & gdcm::operator<< (std::ostream &_os, const VR &val)
- std::ostream & gdcm::operator<< (std::ostream &_os, const UI &_val)
- gdcm::TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN

Variables

- gdcm::VRBINARY

28.288.1 Macro Definition Documentation

28.288.1.1 #define TYPETOENCODING(*type, rep, rtype*)

Value:

```
template<> struct VRToEncoding<VR::type>    \
  { enum { Mode = VR::rep }; };               \
  template<> struct VRToType<VR::type>       \
  { typedef rtype Type; };
```

28.288.1.2 #define VRTypeTemplateCase(*type*)

Value:

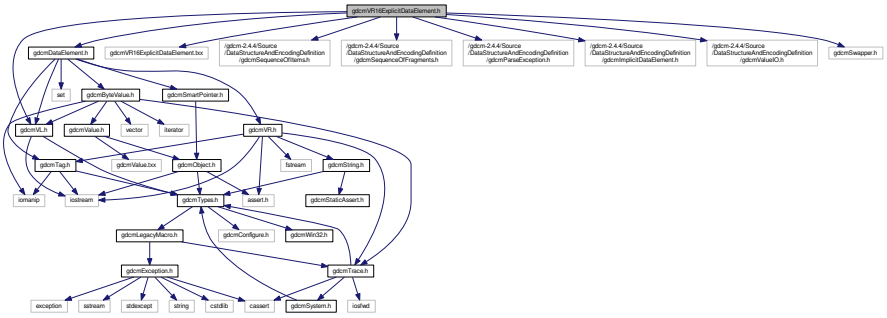
```
case VR::type: \
    return sizeof ( VRToType<VR::type>::Type );
```

Referenced by gdcm::VR::GetSize().

28.289 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.

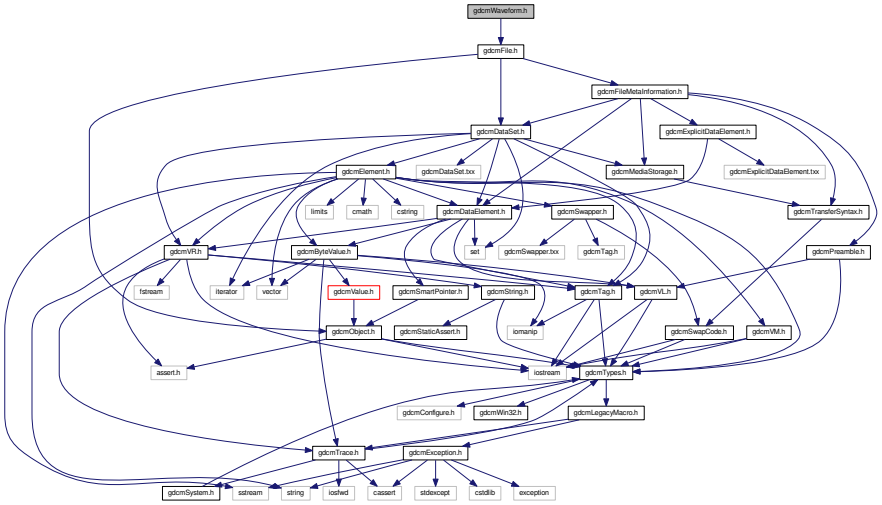
Namespaces

- gdcm

28.290 gdcmWaveform.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmWaveform.h:



Classes

- class gdcm::Waveform

  Waveform class.

Namespaces

- gdcm

28.291 gdcmWin32.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define GDCM_EXPORT

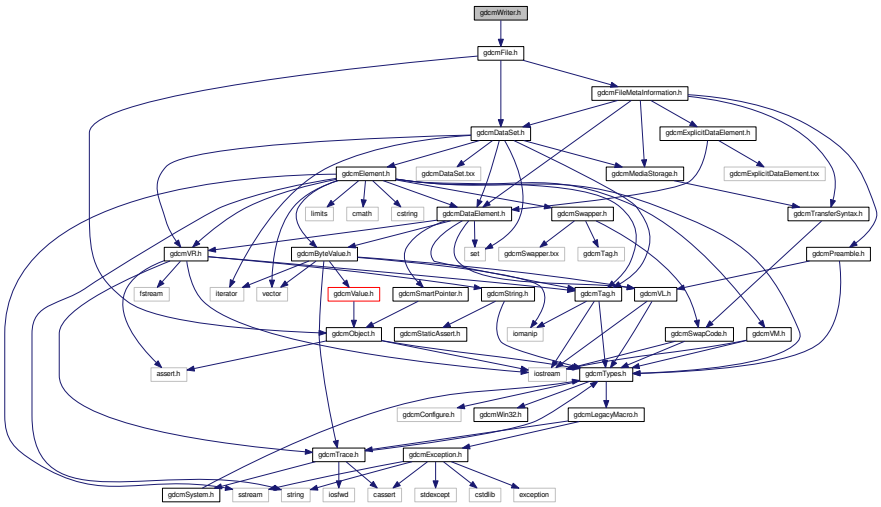
28.291.1 Macro Definition Documentation

28.291.1.1 #define GDCM_EXPORT

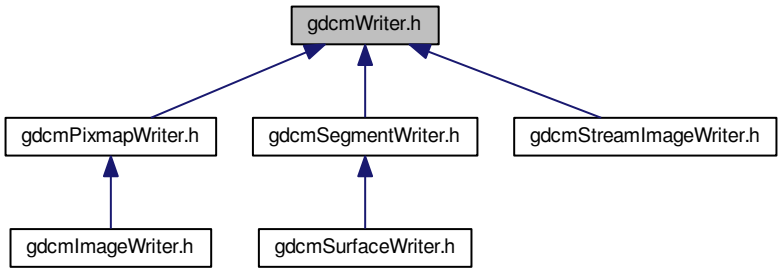
28.292 gdcmWriter.h File Reference

```
#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) This class is a non-validating writer, it will only performs well- formedness check only.

Namespaces

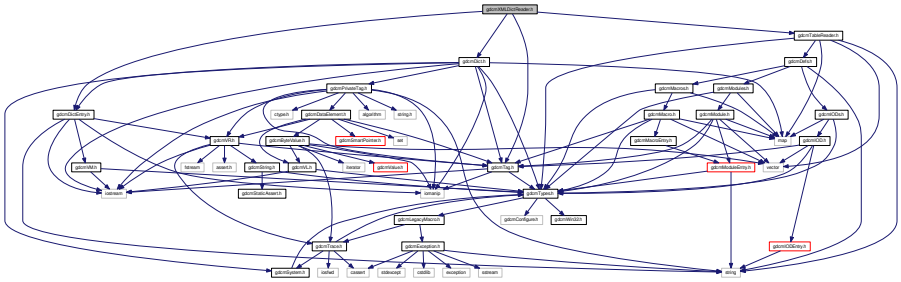
- gdcm

28.293 gdcmxml.dox File Reference

28.294 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.

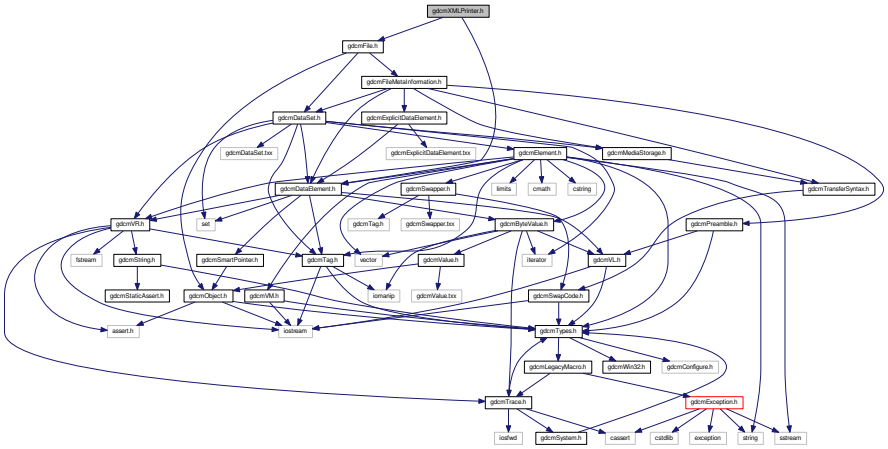
Namespaces

- gdcm

28.295 gdcmXMLPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

Include dependency graph for gdcmXMLPrinter.h:



Classes

- class gdcm::XMLPrinter

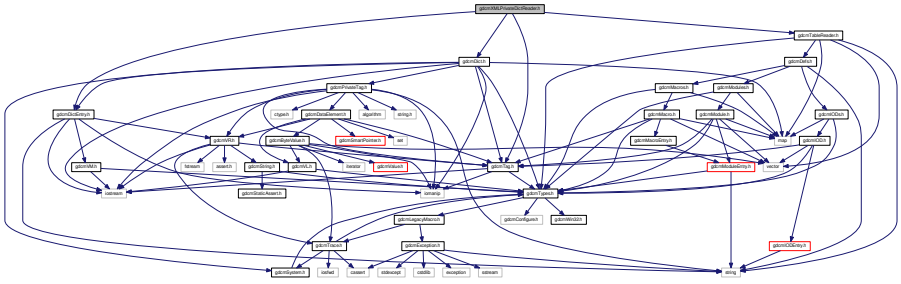
Namespaces

- gdcm

28.296 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

- gdcm

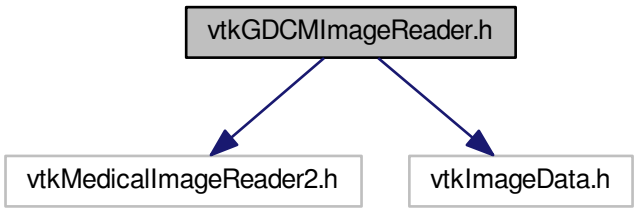
28.297 README.txt File Reference

28.298 TestsList.txt File Reference

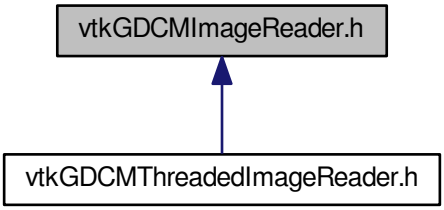
28.299 vtkGDCMImageReader.h File Reference

```
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
```

Include dependency graph for vtkGDCMImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vtkGDCMImageReader

Namespaces

- gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

28.299.1 Macro Definition Documentation

28.299.1.1 #define VTK_CMYK 8

28.299.1.2 #define VTK_INVERSE_LUMINANCE 5

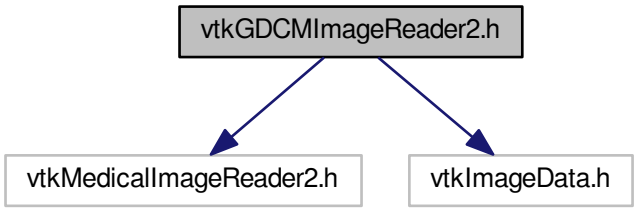
28.299.1.3 #define VTK_LOOKUP_TABLE 6

28.299.1.4 #define VTK_YBR 7

28.300 vtkGDCMImageReader2.h File Reference

```
#include "vtkMedicalImageReader2.h"
#include "vtkImageData.h"
```

Include dependency graph for vtkGDCMImageReader2.h:



Classes

- class vtkGDCMImageReader2

Namespaces

- gdcm

Macros

- #define VTK_CMYK 8
- #define VTK_INVERSE_LUMINANCE 5
- #define VTK_LOOKUP_TABLE 6
- #define VTK_YBR 7

28.300.1 Macro Definition Documentation

28.300.1.1 #define VTK_CMYK 8

28.300.1.2 #define VTK_INVERSE_LUMINANCE 5

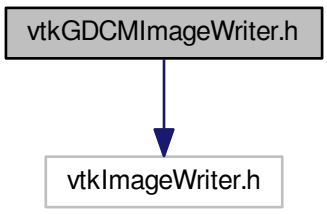
28.300.1.3 #define VTK_LOOKUP_TABLE 6

28.300.1.4 #define VTK_YBR 7

28.301 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



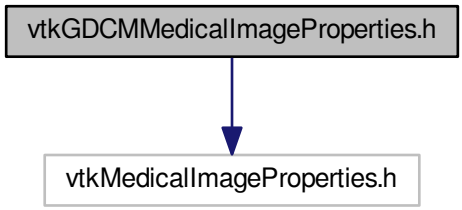
Classes

- class vtkGDCMImageWriter

28.302 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkMedicalImageProperties.h"
```

Include dependency graph for vtkGDCMMedicalImageProperties.h:



Classes

- class vtkGDCMMedicalImageProperties

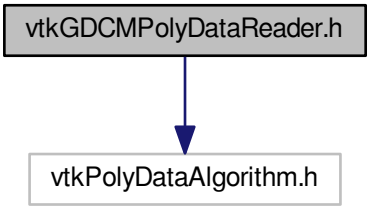
Namespaces

- gdcm

28.303 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:



Classes

- class vtkGDCMPolyDataReader

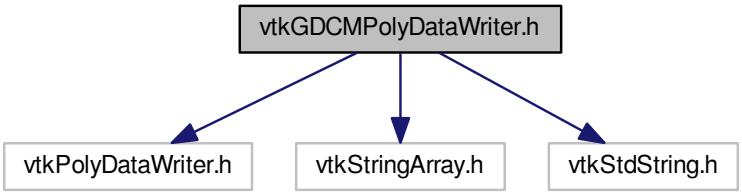
Namespaces

- gdcm

28.304 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkPolyDataWriter.h"
#include "vtkStringArray.h"
#include "vtkStdString.h"
```

Include dependency graph for vtkGDCMPolyDataWriter.h:



Classes

- class vtkGDCMPolyDataWriter

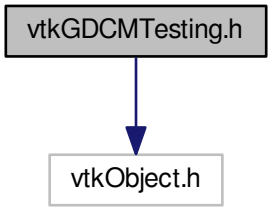
Namespaces

- gdcm

28.305 vtkGDCMTesting.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkGDCMTesting.h:



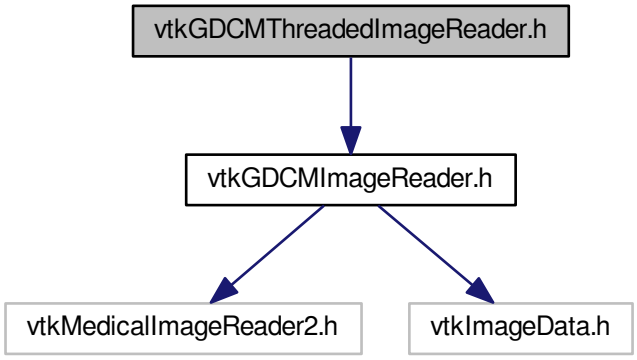
Classes

- class vtkGDCMTesting

28.306 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
```

Include dependency graph for vtkGDCMThreadedImageReader.h:

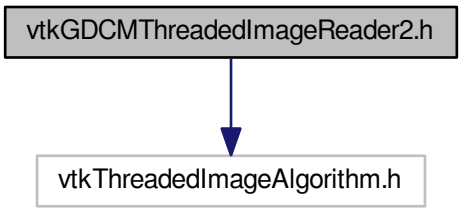


Classes

- class vtkGDCMThreadedImageReader

28.307 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```
Include dependency graph for vtkGDCMThreadedImageReader2.h:

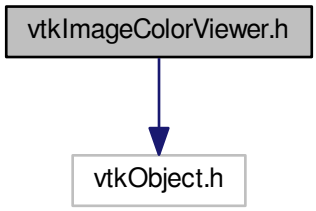


Classes

- class vtkGDCMThreadedImageReader2

28.308 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
```
Include dependency graph for vtkImageColorViewer.h:

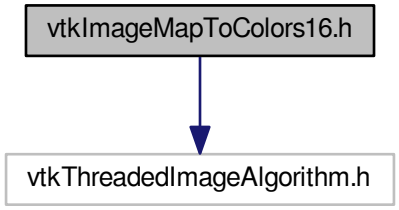


Classes

- class vtkImageColorViewer

28.309 vtkImageMapToColors16.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```
Include dependency graph for vtkImageMapToColors16.h:

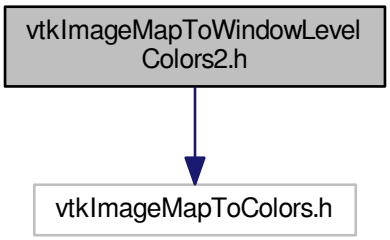


Classes

- class vtkImageMapToColors16

28.310 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkImageMapToColors.h"
```
Include dependency graph for vtkImageMapToWindowLevelColors2.h:



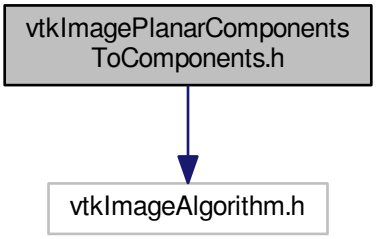
Classes

- class vtkImageMapToWindowLevelColors2

28.311 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



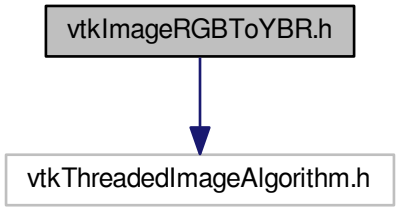
Classes

- class vtkImagePlanarComponentsToComponents

28.312 vtkImageRGBToYBR.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:

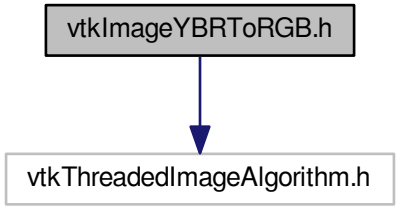


Classes

- class vtkImageRGBToYBR

28.313 vtkImageYBRToRGB.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```
Include dependency graph for vtkImageYBRToRGB.h:

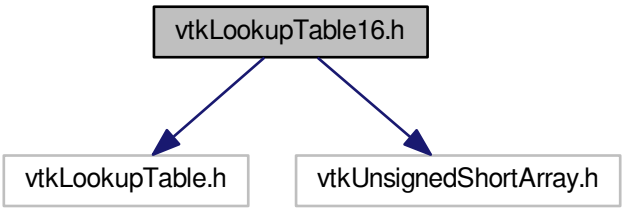


Classes

- class vtkImageYBRToRGB

28.314 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"
#include "vtkUnsignedShortArray.h"
```
Include dependency graph for vtkLookupTable16.h:



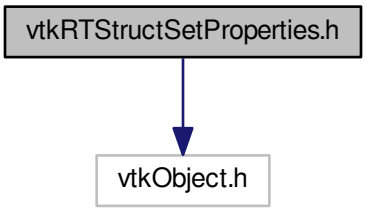
Classes

- class vtkLookupTable16

28.315 vtkRTStructSetProperties.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



Classes

- class vtkRTStructSetProperties

Chapter 29

Example Documentation

29.1 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
    // diferent coloration.

    // Start by creatin 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 saggitalColors = new vtkImageMapToColors();
  saggitalColors.SetInput(theImageData);
  saggitalColors.SetLookupTable(bwLut);
  vtkImageActor saggital = new vtkImageActor();
  saggital.SetInput(saggitalColors.GetOutput());
  saggital.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(saggital);
  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);
  }

}
```

29.2 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;
    }
}
```

29.3 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];

    // instanciate 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;
    }
}
```

29.4 CastConvertPhilips.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 Usage:
17
18  python --public /path/to/directory/
19 or
20  python --private /path/to/directory/
21
22  python --public --extension bak /path/to/directory/
23
24 rename -f 's/\.bak$//' *.bak
25
26 TODO:
27 http://docs.python.org/library/optparse.html#module-optparse
28 """
29
30 import vtkgdcm
31 import vtk
32 import sys
33 import gdcm
34
35 def ProcessOneFilePublic(filename, outfilename, tmpfile):
36   gdcm.ImageHelper.SetForceRescaleInterceptSlope(True)
37   vtkreader = vtkgdcm.vtkGDCMImageReader()
38   vtkreader.SetFileName( filename )
39   vtkreader.Update()
40
41   cast = vtk.vtkImageCast()
42   cast.SetInput( vtkreader.GetOutput() )
43   cast.SetOutputScalarTypeToUnsignedShort()
44
45   # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
46   # Some operation will actually be discarded (we simply need a temp storage)
47   vtkwriter = vtkgdcm.vtkGDCMImageWriter()
48   vtkwriter.SetFileName( tmpfile )
49   vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
50   vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
51   print "Format:",vtkreader.GetImageFormat()
52   vtkwriter.SetImageFormat( vtkreader.GetImageFormat() )
53   vtkwriter.SetInput( cast.GetOutput() )
54   #vtkwriter.Update()
55   vtkwriter.Write()
56
57   # ok now rewrite the exact same file as the original (keep all info)
58   # but use the Pixel Data Element from the written file
59   tmpreader = gdcm.ImageReader()
60   tmpreader.SetFileName( tmpfile )
61   if not tmpreader.Read():
62     sys.exit(1)
63
64   reader = gdcm.Reader()
65   reader.SetFileName( filename )
66   if not reader.Read():
67     sys.exit(1)
68
69   # Make sure to remove Slope/Rescale to avoid re-execution
70   ds = reader.GetFile().GetDataSet()
71   tags = [
72     gdcm.Tag(0x0028,0x1052),
73     gdcm.Tag(0x0028,0x1053),
74     gdcm.Tag(0x0028,0x1053),
75   ]
76   for tag in tags:
77     ds.Remove( tag )
78
```

```
79   writer = gdcm.ImageWriter()
80   writer.SetFileName( outfilename )
81   # Pass image from vtk written file
82   writer.SetImage( tmpreader.GetImage() )
83   # pass dataset from initial 'reader'
84   writer.SetFile( reader.GetFile() )
85   if not writer.Write():
86     sys.exit(1)
87
88 def ProcessOneFilePrivate(filename, outfilename, tmpfile):
89   vtkreader = vtkgdcm.vtkGDCMImageReader()
90   vtkreader.SetFileName( filename )
91   vtkreader.Update()
92
93
94   # (2005,1409)     DS      4       0.0
95   # (2005,140a)     DS      16      1.52283272283272
96
97   # (2005,0014)     LO      26      Philips MR Imaging DD 005
98   tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
99   tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
100
101
102
103   # Need to access some private tags, reread the file (for now):
104   reader = gdcm.Reader()
105   reader.SetFileName( filename )
106   if not reader.Read():
107     sys.exit(1)
108
109   ds = reader.GetFile().GetDataSet()
110
111   el1 = ds.GetDataElement( tag1 )
112   el2 = ds.GetDataElement( tag2 )
113
114
115   #pf = gdcm.PythonFilter()
116   #pf.SetFile( reader.GetFile() )
117   #print el1.GetTag()
118
119   print el1.GetByteValue()
120   v1 = eval(el1.GetByteValue().GetBuffer())
121   print el2.GetByteValue()
122   v2 = eval(el2.GetByteValue().GetBuffer())
123
124   print v1
125   shift = v1
126   print v2
127   scale = v2
128
129   ss = vtk.vtkImageShiftScale()
130   ss.SetInput( vtkreader.GetOutput() )
131   # because VTK image shift / scale convention is inverted from DICOM make sure shift is 0
132   assert shift == 0
133   ss.SetShift( shift )
134   ss.SetScale( scale )
135   ss.SetOutputScalarTypeToUnsignedShort ()
136   ss.Update()
137
138   # vtkGDCMImageWriter does not support Sequence, so let's write a tmp file first:
139   # Some operation will actually be discarded (we simply need a temp storage)
140   vtkwriter = vtkgdcm.vtkGDCMImageWriter()
141   vtkwriter.SetFileName( tmpfile )
142   vtkwriter.SetMedicalImageProperties( vtkreader.GetMedicalImageProperties() )
143   vtkwriter.SetDirectionCosines( vtkreader.GetDirectionCosines() )
144   vtkwriter.SetImageFormat( reader.GetImageFormat() )
145   # do not pass shift/scale again
146   vtkwriter.SetInput( ss.GetOutput() )
147   #vtkwriter.Update()
148   vtkwriter.Write()
149
150   # ok now rewrite the exact same file as the original (keep all info)
151   # but use the Pixel Data Element from the written file
152   tmpreader = gdcm.ImageReader()
153   tmpreader.SetFileName( tmpfile )
154   if not tmpreader.Read():
155     sys.exit(1)
156
157   writer = gdcm.ImageWriter()
158   writer.SetFileName( outfilename )
159   # Pass image from vtk written file
```

```
160   writer.SetImage( tmpreader.GetImage() )
161   # pass dataset from initial 'reader'
162   writer.SetFile( reader.GetFile() )
163   if not writer.Write():
164     sys.exit(1)
165
166 if __name__ == "__main__":
167
168   gdcm.Trace.DebugOff()
169   gdcm.Trace.WarningOff()
170   #filename = sys.argv[1]
171   #outfilename = sys.argv[2]
172   tmpfile = "/tmp/philips_rescaled.dcm"
173   #ProcessOneFile( filename, outfilename, tmpfile )
174   rescaletype = sys.argv[1]
175   assert rescaletype == "--public" or rescaletype == "--private"
176   dirname = sys.argv[2]
177   d = gdcm.Directory()
178   d.Load( dirname )
179
180   for f in d.GetFilenames():
181     #print f
182     ProcessOneFilePublic( f, f + ".bak", tmpfile )
183
184
185 print "success"
```

29.5 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;
}
```

29.6 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;
}
```

29.7 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;
}
```

29.8 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;
}
```

29.9 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);

  // Explicitely 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;
    }
}
```

29.10 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();

  const gdcm::Image &image = reader.GetImage();
  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;
}
```

29.11 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;
    }

}
```

29.12 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 "gdcmIPPSorter.h"

/*
 * 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;
}
```

29.13 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 "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;
}
```

29.14 ConvertMPL.py

```
1 ########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ########################################################################
14
15 """
16 display a DICOM image with matPlotLib via numpy
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Usage:
22
23  python ConvertNumpy.py "IM000000"
24
25 Thanks:
26   plotting example - Ray Schumacher 2009
27 """
28
29 import gdcm
30 import numpy
31 from pylab import *
32
33
34 def get_gdcm_to_numpy_typemap():
35     """Returns the GDCM Pixel Format to numpy array type mapping."""
36     _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
37                 gdcm.PixelFormat.INT8   :numpy.uint8,
```

```
38                 gdcm.PixelFormat.UINT16 :numpy.uint16,
39                 gdcm.PixelFormat.INT16  :numpy.int16,
40                 gdcm.PixelFormat.UINT32 :numpy.uint32,
41                 gdcm.PixelFormat.INT32  :numpy.int32,
42                 gdcm.PixelFormat.FLOAT32:numpy.float32,
43                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
44     return _gdcm_np
45
46 def get_numpy_array_type(gdcm_pixel_format):
47     """Returns a numpy array typecode given a GDCM Pixel Format."""
48     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
49
50 def gdcm_to_numpy(image):
51     """Converts a GDCM image to a numpy array.
52     """
53     pf = image.GetPixelFormat().GetScalarType()
54     print 'pf', pf
55     print image.GetPixelFormat().GetScalarTypeAsString()
56     assert pf in get_gdcm_to_numpy_typemap().keys(), \
57            "Unsupported array type %s"%pf
58     d = image.GetDimension(0), image.GetDimension(1)
59     print 'Image Size: %d x %d' % (d[0], d[1])
60     dtype = get_numpy_array_type(pf)
61     gdcm_array = image.GetBuffer()
62     ## use float for accurate scaling
63     result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
64     ## optional gamma scaling
65     #maxV = float(result[result.argmax()])
66     #result = result + .5*(maxV-result)
67     #result = numpy.log(result+50) ## apprx background level
68     result.shape = d
69     return result
70
71 if __name__ == "__main__":
72   import sys
73   r = gdcm.ImageReader()
74   filename = sys.argv[1]
75   r.SetFileName( filename )
76   if not r.Read():  sys.exit(1)
77   numpy_array = gdcm_to_numpy( r.GetImage() )
78
79   subplot(111)# one plot, on left
80   title(filename)
81   ## many colormaps are available
82   imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
83   ## set the plot sizes and placement
84   subplots_adjust(bottom=0.1, right=0.8, top=0.9)
85   cax = axes([0.85, 0.1, 0.075, 0.8])
86   colorbar(cax=cax)
87   title('values')
88   get_current_fig_manager().window.title('plot')
89   show()
```

29.15 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 "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;
}
```

29.16 ConvertNumpy.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
```

```
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 This module add support for converting a gdcm.Image to a numpy array.
17
18 Caveats:
19 - Does not support UINT12/INT12
20
21 Removed:
22 - float16 is defined in GDCM API but no implementation exist for it ...
23 """
24
25 import gdcm
26 import numpy
27
28 def get_gdcm_to_numpy_typemap():
29     """Returns the GDCM Pixel Format to numpy array type mapping."""
30     _gdcm_np = {gdcm.PixelFormat.UINT8  :numpy.int8,
31                 gdcm.PixelFormat.INT8   :numpy.uint8,
32                 #gdcm.PixelFormat.UINT12 :numpy.uint12,
33                 #gdcm.PixelFormat.INT12  :numpy.int12,
34                 gdcm.PixelFormat.UINT16 :numpy.uint16,
35                 gdcm.PixelFormat.INT16  :numpy.int16,
36                 gdcm.PixelFormat.UINT32 :numpy.uint32,
37                 gdcm.PixelFormat.INT32  :numpy.int32,
38                 #gdcm.PixelFormat.FLOAT16:numpy.float16,
39                 gdcm.PixelFormat.FLOAT32:numpy.float32,
40                 gdcm.PixelFormat.FLOAT64:numpy.float64 }
41     return _gdcm_np
42
43 def get_numpy_array_type(gdcm_pixel_format):
44     """Returns a numpy array typecode given a GDCM Pixel Format."""
45     return get_gdcm_to_numpy_typemap()[gdcm_pixel_format]
46
47 def gdcm_to_numpy(image):
48     """Converts a GDCM image to a numpy array.
49     """
50     pf = image.GetPixelFormat()
51
52     assert pf.GetScalarType() in get_gdcm_to_numpy_typemap().keys(), \
53            "Unsupported array type %s"%pf
54
55     shape = image.GetDimension(0) * image.GetDimension(1), pf.GetSamplesPerPixel()
56     if image.GetNumberOfDimensions() == 3:
57       shape = shape[0] * image.GetDimension(2), shape[1]
58
59     dtype = get_numpy_array_type(pf.GetScalarType())
60     gdcm_array = image.GetBuffer()
61     result = numpy.frombuffer(gdcm_array, dtype=dtype)
62     result.shape = shape
63     return result
64
65 if __name__ == "__main__":
66   import sys
67   r = gdcm.ImageReader()
68   filename = sys.argv[1]
69   r.SetFileName( filename )
70   if not r.Read():
71     sys.exit(1)
72
73   numpy_array = gdcm_to_numpy( r.GetImage() )
74   print numpy_array
```

29.17 ConvertPIL.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
```

```
#     This software is distributed WITHOUT ANY WARRANTY; without even
#     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
#     PURPOSE.  See the above copyright notice for more information.
#
#########################################################################

"""
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()])
    ## linear gamma adjust
    #result = result + .5*(maxV-result)
    ## log gamma
    result = numpy.log(result+50) ## 50 is apprx background level
    maxV = float(result[result.argmax()])
    result = result*(2.**8/maxV) ## histogram stretch
    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() )
  ## L is 8 bit grey
  ## http://www.pythonware.com/library/pil/handbook/concepts.htm
  pilImage = Image.frombuffer('L',
                              numpy_array.shape,
                              numpy_array.astype(numpy.uint8),
                              'raw','L',0,1)
  ## cutoff removes background noise and spikes
  pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
  pilImage.save(sys.argv[1]+'.jpg')
```

29.18 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 "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;
}
```

29.19 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"

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;
}
```

29.20 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;
}
```

29.21 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;
}
```

29.22 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;
}
```

29.23 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 "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->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;
}
```

29.24 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;
}
```

29.25 CreateRAWStorage.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16     <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"/>
17 """
18
19 import gdcm
20 import sys,os
21
22 if __name__ == "__main__":
23   r = gdcm.Reader()
24   # Will require Testing...
25   dataroot = gdcm.Testing.GetDataRoot()
26   filename = os.path.join( dataroot, '012345.002.050.dcm' )
27   r.SetFileName( filename )
28   r.Read()
```

```
29   f = r.GetFile()
30   ds = f.GetDataSet()
31
32   uid = "1.2.840.10008.5.1.4.1.1.66"
33 #  f = gdcm.File()
34 #  ds = f.GetDataSet()
35   de = gdcm.DataElement( gdcm.Tag(0x0008,0x0016) )
36   de.SetByteValue( uid, gdcm.VL(len(uid)) )
37   vr = gdcm.VR( gdcm.VR.UI )
38   de.SetVR( vr )
39   ds.Replace( de )
40
41   ano = gdcm.Anonymizer()
42   ano.SetFile( r.GetFile() )
43   ano.RemovePrivateTags()
44   ano.RemoveGroupLength()
45   taglist = [
46   gdcm.Tag(0x0008,0x0008),
47   gdcm.Tag(0x0008,0x0022),
48   gdcm.Tag(0x0008,0x0032),
49   gdcm.Tag(0x0008,0x2111),
50   gdcm.Tag(0x0008,0x1150),
51   gdcm.Tag(0x0008,0x1155),
52   gdcm.Tag(0x0008,0x0100),
53   gdcm.Tag(0x0008,0x0102),
54   gdcm.Tag(0x0008,0x0104),
55   gdcm.Tag(0x0040,0xa170),
56   gdcm.Tag(0x0008,0x2112),
57   gdcm.Tag(0x0008,0x0100),
58   gdcm.Tag(0x0008,0x0102),
59   gdcm.Tag(0x0008,0x0104),
60   gdcm.Tag(0x0008,0x9215),
61   gdcm.Tag(0x0018,0x0010),
62   gdcm.Tag(0x0018,0x0022),
63   gdcm.Tag(0x0018,0x0050),
64   gdcm.Tag(0x0018,0x0060),
65   gdcm.Tag(0x0018,0x0088),
66   gdcm.Tag(0x0018,0x0090),
67   gdcm.Tag(0x0018,0x1040),
68   gdcm.Tag(0x0018,0x1100),
69   gdcm.Tag(0x0018,0x1110),
70   gdcm.Tag(0x0018,0x1111),
71   gdcm.Tag(0x0018,0x1120),
72   gdcm.Tag(0x0018,0x1130),
73   gdcm.Tag(0x0018,0x1150),
74   gdcm.Tag(0x0018,0x1151),
75   gdcm.Tag(0x0018,0x1152),
76   gdcm.Tag(0x0018,0x1160),
77   gdcm.Tag(0x0018,0x1190),
78   gdcm.Tag(0x0018,0x1210),
79   gdcm.Tag(0x0020,0x0012),
80   gdcm.Tag(0x0020,0x0032),
81   gdcm.Tag(0x0020,0x0037),
82   gdcm.Tag(0x0020,0x1041),
83   gdcm.Tag(0x0020,0x4000),
84   gdcm.Tag(0x0028,0x0002),
85   gdcm.Tag(0x0028,0x0004),
86   gdcm.Tag(0x0028,0x0010),
87   gdcm.Tag(0x0028,0x0011),
88   gdcm.Tag(0x0028,0x0030),
89   gdcm.Tag(0x0028,0x0100),
90   gdcm.Tag(0x0028,0x0101),
91   gdcm.Tag(0x0028,0x0102),
92   gdcm.Tag(0x0028,0x0103),
93   gdcm.Tag(0x0028,0x1052),
94   gdcm.Tag(0x0028,0x1053),
95   gdcm.Tag(0x0028,0x2110),
96   gdcm.Tag(0x0028,0x2112),
97   gdcm.Tag(0x7fe0,0x0010),
98   gdcm.Tag(0x0018,0x0020),
99   gdcm.Tag(0x0018,0x0021),
100   gdcm.Tag(0x0018,0x0023),
101   gdcm.Tag(0x0018,0x0025),
102   gdcm.Tag(0x0018,0x0080),
103   gdcm.Tag(0x0018,0x0081),
104   gdcm.Tag(0x0018,0x0083),
105   gdcm.Tag(0x0018,0x0084),
106   gdcm.Tag(0x0018,0x0085),
107   gdcm.Tag(0x0018,0x0086),
108   gdcm.Tag(0x0018,0x0087),
109   gdcm.Tag(0x0018,0x0091),
```

```
110   gdcm.Tag(0x0018,0x0093),
111   gdcm.Tag(0x0018,0x0094),
112   gdcm.Tag(0x0018,0x0095),
113   gdcm.Tag(0x0018,0x1088),
114   gdcm.Tag(0x0018,0x1090),
115   gdcm.Tag(0x0018,0x1094),
116   gdcm.Tag(0x0018,0x1250),
117   gdcm.Tag(0x0018,0x1251),
118   gdcm.Tag(0x0018,0x1310),
119   gdcm.Tag(0x0018,0x1312),
120   gdcm.Tag(0x0018,0x1314),
121   gdcm.Tag(0x0018,0x1315),
122   gdcm.Tag(0x0018,0x1316),
123   gdcm.Tag(0x0020,0x0110),
124   gdcm.Tag(0x0028,0x0120),
125   gdcm.Tag(0x0028,0x1050),
126   gdcm.Tag(0x0028,0x1051)
127   ]
128   for tag in taglist:
129     #print tag
130     ano.Remove( tag )
131
132   # special handling
133   gen = gdcm.UIDGenerator()
134   ano.Replace( gdcm.Tag(0x0008,0x9123), gen.Generate() )
135   #ano.Empty( gdcm.Tag(0x0040,0x0555) )
136
137
138 #
139 #  uid = gen.Generate()
140 #  de.SetTag( gdcm.Tag(0x0008,0x0018) )
141 #  de.SetByteValue( uid, gdcm.VL(len(uid)) )
142 #  ds.Insert( de )
143
144   # init FMI now:
145   #fmi = f.GetHeader()
146   #ts = gdcm.TransferSyntax()
147   #print ts
148   #fmi.SetDataSetTransferSyntax( ts ) # default
149   #print fmi.GetDataSetTransferSyntax()
150   #de.SetTag( gdcm.Tag(0x0002,0x0010) )
151   #uid = "1.2.840.10008.1.2"
152   #de.SetByteValue( uid, gdcm.VL(len(uid)) )
153   #fmi.Insert( de )
154 #  f.SetHeader( r.GetFile().GetHeader() )
155
156   writer = gdcm.Writer()
157   writer.SetFile( ano.GetFile() )
158   writer.SetFileName( "rawstorage.dcm" );
159   writer.Write()
```

29.26 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;
}
```

29.27 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();
}
```

29.28 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;
      }

    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;
    }
}
```

29.29 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 );
      }

    }
}
```

29.30 DecompressImage.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 Usage:
17
18  python DecompressImage.py gdcmData/012345.002.050.dcm decompress.dcm
19 """
20
21 import gdcm
22 import sys
23
24 if __name__ == "__main__":
25
26   file1 = sys.argv[1]
27   file2 = sys.argv[2]
28
29   r = gdcm.ImageReader()
30   r.SetFileName( file1 )
31   if not r.Read():
32     sys.exit(1)
33
34   image = gdcm.Image()
35   ir = r.GetImage()
36
37   image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
38   dims = ir.GetDimensions();
39   print ir.GetDimension(0);
40   print ir.GetDimension(1);
41   print "Dims:",dims
42
43   #  Just for fun:
44   dircos =  ir.GetDirectionCosines()
45   t = gdcm.Orientation.GetType(dircos)
46   l = gdcm.Orientation.GetLabel(t)
47   print "Orientation label:",l
48
49   image.SetDimension(0, ir.GetDimension(0) );
50   image.SetDimension(1, ir.GetDimension(1) );
51
52   pixeltype = ir.GetPixelFormat();
53   image.SetPixelFormat( pixeltype );
54
55   pi = ir.GetPhotometricInterpretation();
56   image.SetPhotometricInterpretation( pi );
57
58   pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
59   str1 = ir.GetBuffer()
60   #print ir.GetBufferLength()
61   pixeldata.SetByteValue( str1, gdcm.VL( len(str1) ) )
62   image.SetDataElement( pixeldata )
63
64   w = gdcm.ImageWriter()
65   w.SetFileName( file2 )
66   w.SetFile( r.GetFile() )
67   w.SetImage( image )
68   if not w.Write():
69     sys.exit(1)
```

29.31 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 file are not garantee 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;
    }
}
```

29.32 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;
    }
}
```

29.33 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 );
      }

    }
}
```

29.34 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;
}
```

29.35 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 = NULL;
#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.c_str(), 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;
}
```

29.36 DumbAnonymizer.py

```
1 ########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ########################################################################
14
15 """
16 This example shows how one can use the gdcm.Anonymizer in 'dumb' mode.
17 This class becomes really handy when one knows which particular tag to fill in.
18
19 Usage:
20
21  python DumbAnonymizer.py gdcmData/012345.002.050.dcm out.dcm
22
23 """
24
25 import gdcm
26
27 # http://www.oid-info.com/get/1.3.6.1.4.17434
28 THERALYS_ORG_ROOT = "1.3.6.1.4.17434"
29
30 tag_rules={
31   # Value
32   (0x0012,0x0010):("Value","MySponsorName"),
33   (0x0012,0x0020):("Value","MyProtocolID"),
34   (0x0012,0x0021):("Value","MyProtocolName"),
35   (0x0012,0x0062):("Value","YES"),
36   (0x0012,0x0063):("Value","MyDeidentificationMethod"),
37
38   # Method
39   #(0x0002,0x0003):("Method","GenerateMSOPId"),
40   #(0x0008,0x1155):("Method","GenerateMSOPId"),
41   (0x0008,0x0018):("Method","GenerateMSOPId"),
42   (0x0010,0x0010):("Method","GetSponsorInitials"),
43   (0x0010,0x0020):("Method","GetSponsorId"),
```

```
44    (0x0012,0x0030):("Method","GetSiteId"),
45    (0x0012,0x0031):("Method","GetSiteName"),
46    (0x0012,0x0040):("Method","GetSponsorId"),
47    (0x0012,0x0050):("Method","GetTPId"),
48    (0x0018,0x0022):("Method","KeepIfExist"),
49    (0x0018,0x1315):("Method","KeepIfExist"),
50    (0x0020,0x000d):("Method","GenerateStudyId"),
51    (0x0020,0x000e):("Method","GenerateSeriesId"),
52    (0x0020,0x1002):("Method","GetNumberOfFrames"),
53    (0x0020,0x0020):("Method","GetPatientOrientation"),
54    # Other:
55    (0x0012,0x0051):("Patient Field","Type Examen"),
56    (0x0018,0x1250):("Sequence Field","Receive Coil"),
57    (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
58    (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
59    (0x0018,0x0082):("Sequence Field","Invertion Time"),
60  }
61
62  class MyAnon:
63    def __init__(self):
64      self.studyuid = None
65      self.seriesuid = None
66      generator = gdcm.UIDGenerator()
67      if not self.studyuid:
68        self.studyuid = generator.Generate()
69      if not self.seriesuid:
70        self.seriesuid = generator.Generate()
71    def GetSponsorInitials(self):
72      return "dummy^foobar"
73    def GenerateStudyId(self):
74      return self.studyuid
75    def GenerateSeriesId(self):
76      return self.seriesuid
77    #def GenerateMSOPId(self):
78    def GenerateMSOPId(self):
79      generator = gdcm.UIDGenerator()
80      return generator.Generate()
81    def GetSiteId(self):
82      return "MySiteId"
83    def GetSiteName(self):
84      return "MySiteName"
85    def GetSponsorId(self):
86      return "MySponsorId"
87    def GetTPId(self):
88      return "MyTP"
89
90  if __name__ == "__main__":
91    import sys
92    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle
       ( "DumbAnonymizer" )
93    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
94
95    r = gdcm.Reader()
96    filename = sys.argv[1]
97    r.SetFileName( filename )
98    if not r.Read(): sys.exit(1)
99
100    obj = MyAnon()
101
102    w = gdcm.Writer()
103    ano = gdcm.Anonymizer()
104    ano.SetFile( r.GetFile() )
105    ano.RemoveGroupLength()
106    for tag,rule in tag_rules.items():
107      if rule[0] == 'Value':
108        print tag,rule
109        ano.Replace( gdcm.Tag( tag[0], tag[1] ), rule[1] )
110      elif rule[0] == 'Method':
111        print tag,rule
112        # result = locals()[rule[1]]()
113        methodname = rule[1]
114        if hasattr(obj, methodname):
115          _member = getattr(obj, methodname)
116          result = _member()
117          ano.Replace( gdcm.Tag( tag[0], tag[1] ), result )
118        else:
119          print "Problem with: ", methodname
120
121    outfilename = sys.argv[2]
122    w.SetFileName( outfilename )
123    w.SetFile( ano.GetFile() )
```

```
124   if not w.Write(): sys.exit(1)
```

29.37 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, NULL }
};

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;
}
```

29.38 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 [1.3.46.670589.11.17240.5.23.4.1.3012.20100324094825680018]         # 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
  "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_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,
  param_string,
  param_3, // ??
  param_enum,
} param_type;

static const char *gettypenamefromtype( int i)
{
  const char *ret = NULL;
  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));
    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];
  int8_t boolean;
  int32_t type;
  uint32_t dim;
  uint32_t v4;
  /*int32_t*/ std::streamoff offset;
  param_type gettype() const { return (param_type)type; }
  uint32_t getdim() const { return dim; }
  void read( std::istream & is )
    {
    is.read( name, 32 + 1);
    //assert( name[32] == 0 ); // fails sometimes...
    // This is always the same issue the string can contains garbarge from previous run,
    // we need to print only until the first \0 character:
    assert( strlen( name ) <= 32 ); // sigh
    is.read( (char*)&boolean,1);
    assert( boolean == 0 || boolean == 1 ); // some kind of bool...
    is.read( (char*)&type, sizeof( type ) );
    assert( gettypenamefromtype( type ) );
    is.read( (char*)&dim, sizeof( dim ) );
    is.read( (char*)&v4, sizeof( v4 ) );
    //assert( v4 == 0 ); // always 0 ? sometimes not...
    const std::streamoff cur = is.tellg();
    is.read( (char*)&offset, sizeof( offset ) );
    offset += cur;
    }

  void print( std::ostream & os ) const
    {
    os << name << ",";
```

```
    os << (int)boolean << ",";
    os << type << ",";
    os << dim << ",";
    os << v4 << ",";
    os << offset << std::endl;
    }
  void printvalue( std::ostream & os, std::istream & is ) const
    {
    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:
        {
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os << ",";
          int32_t v;
          is.read( (char*)&v, sizeof(v) );
          os << v;
          }
        }
      break;
    case param_string:
        {
        std::string v;
        v.resize( dim );
        is.read( &v[0], dim );
        os << v;
        }
      break;
    case param_enum:
        {
        for( uint32_t idx = 0; idx < dim; ++idx )
          {
          if( idx ) os << ",";
          int32_t v;
          is.read( (char*)&v, sizeof(v) );
          os << v;
          }
        }
      break;
      }

    }
  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 ?
  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 ?
  const gdcm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
  if( !ds.FindDataElement( pt1 ) ) return false;
  const gdcm::DataElement &de1 = ds.GetDataElement( pt1 );

  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();
    std::string s1( bv1->GetPointer() , bv1->GetLength() );

    if( s1 == "IEEE_PDF" )
      {
      //  std::cout << "Len= " << bv->GetLength() << std::endl;
#if 0
      std::string fn = gdcm::LOComp::Trim( s.c_str() ); // remove trailing space
      std::ofstream out( fn.c_str() );
      out.write( bv->GetPointer(), bv->GetLength() );
      out.close();
#endif

      std::istringstream is;
      std::string dup( bv->GetPointer(), bv->GetLength() );
      is.str( dup );

      header h;
      h.read( is );
#if 0
      std::cout << s0.c_str() << std::endl;
      h.print( std::cout );
#endif

      assert( is.tellg() == std::streampos(0x20) );
      is.seekg( 0x20 );

      std::vector< param > params;
      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
      assert( isvalidpdfstring( fn.c_str() ) );
      fn += ".csv";
      //fn += ".xml";
      std::ofstream csv( fn.c_str() );

      // let's do some bookeeping:
      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\5f\64\6
          # 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 ) ) return 1;
    }

  return 0;
}
```

29.39 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( tvaluesl3 ) )
      {
      const DataElement & value = ds.GetDataElement( tvaluesl3 );
      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 1;
  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 )
```

```
{
  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 1;
    }
  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 1;
      }
    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;
}
```

29.40 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 );

  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.size() )
    {
    is.read( &bytes[0], l - 16 );
    }
  //os << "pos:" << is.tellg() << std::endl;

  if( strcmp(str, "TUSREMEASUREMENT" ) == 0 )
    {
    const char *p = &bytes[0];
    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);
    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;
}
```

29.41 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, 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[0];
    assert( is.tellg() == offsets[r] + 16 );
    const Bytef *source = (Bytef*)buf + offsets[r] + 16;
    uLong sourceLen;
    if( r + 1 == nframes )
      sourceLen = totalsize - offsets[r] - 16;
    else
      sourceLen = offsets[r+1] - 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[0], 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, 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[0];

  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;
}
```

29.42 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(0);

  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(0);

  std::cout << "Finished loading data from : " << nfiles << " files" << std::endl;

//  MappingType const &mappings = s.GetMappings();


  sqlite3* db;
  sqlite3_open("./dicom.db", &db);

  if(db == 0)
    {
    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, 0, 0, &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,
      0  // 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(0);

  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;
}
```

29.43 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 explicitely 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;
}
```

29.44 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 )
{
  short * buffer = (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;
}
```

29.45 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;
}
```

29.46 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;
    }
}
```

29.47 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;
}
```

29.48 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;
}
```

29.49 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;
    }
}
```

29.50 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");
        }
      }

    }
}
```

29.51 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;
    }
}
```

29.52 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 <openjpeg.h>
#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 "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;
}
```

29.53 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;
    }
}
```

29.54 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;
}
```

29.55 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;
    }
}
```

29.56 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" );
    }
}
```

29.57 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;
    }
}
```

29.58 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;
    }
}
```

29.59 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;
    }
}
```

29.60 FindAllPatientName.py

```
1 ######################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ######################################################################
14 """
15 This example shows how one can use the gdcm.CompositeNetworkFunctions class
16 for executing a C-FIND query
17 It will print the list of patient name found
18
19 Usage:
20
21  python FindAllPatientName.py
22
23 """
24
25 import gdcm
26
27 # Patient Name
28 tag = gdcm.Tag(0x10,0x10)
29 de = gdcm.DataElement(tag)
30
31 # Search all patient name where string match 'F*'
32 de.SetByteValue('F*',gdcm.VL(2))
33
34 ds = gdcm.DataSet()
35 ds.Insert(de)
36
37 cnf = gdcm.CompositeNetworkFunctions()
38 theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)
39
40 #print theQuery.ValidateQuery()
41
42 # prepare the variable for output
43 ret = gdcm.DataSetArrayType()
44
45 # Execute the C-FIND query
46 cnf.CFind('dicom.example.com',11112,theQuery,ret,'GDCM_PYTHON','ANY-SCP')
47
48 for i in range(0,ret.size()):
49   print "Patient #",i
50   print ret[i]
```

29.61 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 attemps 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".
 */
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);

  const gdcm::ByteValue *bv = frag0.GetByteValue();
  const char *ptr = bv->GetPointer();
  size_t len = bv->GetLength();

  const char sig[] = "\x00\x00\x00\x00\x6A\x70\x32\x63";
  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;
  const 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;
}
```

29.62 FixCommaBug.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 Using LC_NUMERIC set to something not compatible with "C" it is possible to write out "," instead of
17 "." as required by the DICOM standard
18 Issue is still current (IMHO) with gdcm 2.0.9
19 """
20
21 import gdcm
22 import sys
23
24 filename = sys.argv[1]
25 outname = sys.argv[2]
26
27 # read
28 r = gdcm.Reader()
29 r.SetFileName( filename )
30 if not r.Read():
31   print "not valid"
32   sys.exit(1)
33
```

```
34 file = r.GetFile()
35 dataset = file.GetDataSet()
36
37 ano = gdcm.Anonymizer()
38 ano.SetFile( file )
39
40 tags = [
41 gdcm.Tag(0x0018,0x1164),
42 gdcm.Tag(0x0018,0x0088),
43 gdcm.Tag(0x0018,0x0050),
44 gdcm.Tag(0x0028,0x0030),
45 ]
46
47 for tag in tags:
48   print tag
49   if dataset.FindDataElement( tag ):
50     pixelspacing = dataset.GetDataElement( tag )
51     #print pixelspacing
52     bv = pixelspacing.GetByteValue()
53     str = bv.GetBuffer()
54     #print bv.GetLength()
55     #print len(str)
56     new_str = str.replace(",",".")
57     # Need to explicitly pass bv.GetLength() to remove any trailing garbage
58     ano.Replace( tag, new_str, bv.GetLength() )
59
60 #print dataset
61
62 w = gdcm.Writer()
63 w.SetFile( file )
64 w.SetFileName( outname )
65 if not w.Write():
66   print "Cannot write"
67   sys.exit(1)
68
69 # paranoid:
70 image_reader = gdcm.ImageReader()
71 image_reader.SetFileName( outname )
72 if not image_reader.Read():
73   print "there is still a comma"
74   sys.exit(1)
75
76 print "Sucess!"
77 sys.exit(0) # success
```

29.63 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<BYTE> 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[0];
    bv->GetBuffer(buffer, totalLen);
    const BYTE* pbyteCompressed0 = (const BYTE*)buffer;
    while( totalLen > 0 && pbyteCompressed0[totalLen-1] != 0xd9 )
      {
      totalLen--;
      }

    JlsParameters metadata;
    if (JpegLsReadHeader(buffer, totalLen, &metadata) != OK)
      {
      std::cerr << "Cant parse jpegls" << std::endl;
      return false;
      }

    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 = NULL;
    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 << "Cant 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[0];
    size_t cbyteCompressed = vbuffer.size(); // updated legnth

    JlsParameters params;
    JpegLsReadHeader(pbyteCompressed, cbyteCompressed, &params);
```

```
    std::vector<BYTE> rgbyteOut;
    //rgbyteOut.resize( image.GetBufferLength() );
    rgbyteOut.resize(params.height *params.width * ((params.bitspersample + 7)
        / 8) * params.components);

    JLS_ERROR result =
      JpegLsDecode(&rgbyteOut[0], rgbyteOut.size(), pbyteCompressed, cbyteCompressed, &params );
    if (result != OK)
      {
      std::cerr << "Could not patch JAI-JPEGLS" << 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[0], (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;
}
```

29.64 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 "gdcmSystem.h"
#include "gdcmDirectory.h"
#include "gdcmIPPSorter.h"

#ifndef vtkFloatingPointType
#define vtkFloatingPointType float
#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' postions
//
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;
}
```

29.65 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"

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 ); // cant 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;
}
```

29.66 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 "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\36.10\38.62\44.88\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\3
       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( (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\44.6\28.9\44.2\29.7\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7\40.
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;
}
```

29.67 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 "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\36.10\38.62\44.88\44.88\44.88\45.00\45.00\45.00\45.66\45.66\46.42\39.77\39.77\39.77\39.77\3
       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( (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\44.6\28.9\44.2\29.7\43.9\31.5\43.5\33.0\42.8\33.7\42.4\35.2\41.3\38.2\40.4\39.6\39.7\40.
      (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;
}
```

29.68 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"


// 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;
}
```

29.69 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"

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 ); // cant get it to work
  cube->GetAssembly()->SetUserTransform( transform ); // cant 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;
}
```

29.70 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 "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#include "vtkVolumeTextureMapper3D.h"
#include "vtkFixedPointVolumeRayCastMapper.h"
#include "vtkInteractorStyleTrackballCamera.h"
#include "vtkRenderer.h"
#include "vtkRenderWindow.h"
#include "vtkImageClip.h"
#include "vtkRenderWindowInteractor.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;
}
```

29.71 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)(1 << 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;
}
```

29.72 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;
    }
}
```

29.73 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 <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];
        sprintf(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;
}
```

29.74 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;
}
```

29.75 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;
}
```

29.76 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 pupose of this execise 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;
}
```

29.77 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;
}
```

29.78 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;
}
```

29.79 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;
    }
}
```

29.80 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;
}
```

29.81 GetPortionCSAHeader.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 Usage:
17
18  python GetPortionCSAHeader.py input.dcm
19
20 Footnote:
21   SIEMENS is not publishing any information on the CSA header. So any info extracted
22   is at your own risk.
23 """
24
25 import sys
26 import gdcm
27
28 if __name__ == "__main__":
29
30   file = sys.argv[1]
31
32   r = gdcm.Reader()
33   r.SetFileName( file )
34   if not r.Read():
35     sys.exit(1)
36
37   ds = r.GetFile().GetDataSet()
38   csa_t1 = gdcm.CSAHeader()
39   csa_t2 = gdcm.CSAHeader()
40   #print csa
41   t1 = csa_t1.GetCSAImageHeaderInfoTag();
42   print t1
43   t2 = csa_t2.GetCSASeriesHeaderInfoTag();
44   print t2
45   # Let's do it for t1:
46   if ds.FindDataElement( t1 ):
47     csa_t1.LoadFromDataElement( ds.GetDataElement( t1 ) )
48     print csa_t1
49
50   # Now let's pretend we are only interested in B_value and DiffusionGradientDirection entries:
51   bvalues = csa_t1.GetCSAElementByName( "B_value" ) # WARNING: it is case sensitive !
52   print bvalues
```

```
53
54   diffgraddir = csa_t1.GetCSAElementByName( "DiffusionGradientDirection" ) # WARNING: it is case sensitive
       !
55   print diffgraddir
56
57   # repeat for t2 if you like it:
58   if ds.FindDataElement( t2 ):
59     csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
60     # print csa_t2
61
62   gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
63   print gdt
64
65   bv = gdt.GetByteValue();
66   #print bv
67   str = bv.GetPointer()
68   print str.split("\\")
```

29.82 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;
}
```

29.83 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 1;
  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 1;
  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 1;
    const DataElement& de8 = subds4.GetDataElement( tseq8 );
    Element<VR::UL,VM::VM1> ldimz;
    ldimz.SetFromDataElement( de8 );
    dimz += ldimz.GetValue();
    if( !subds4.FindDataElement( tseq4 ) ) return 1;
    const DataElement& seq4 = subds4.GetDataElement( tseq4 );
    if( !subds4.FindDataElement( tseq5 ) ) return 1;
    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[0], (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;
}
```

29.84 headsq2dcm.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 Usage:
17  python headsq2dcm.py -D /path/to/VTKData
18 """
19
20 import vtk
21 import vtkgdcm
22 from vtk.util.misc import vtkGetDataRoot
```

```
23 VTK_DATA_ROOT = vtkGetDataRoot()
24
25 reader = vtk.vtkVolume16Reader()
26 reader.SetDataDimensions(64, 64)
27 reader.SetDataByteOrderToLittleEndian()
28 reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter")
29 reader.SetImageRange(1, 93)
30 reader.SetDataSpacing(3.2, 3.2, 1.5)
31
32 cast = vtk.vtkImageCast()
33 cast.SetInput( reader.GetOutput() )
34 cast.SetOutputScalarTypeToUnsignedChar()
35
36 # By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
37 writer = vtkgdcm.vtkGDCMImageWriter()
38 writer.SetFileName( "headsq.dcm" )
39 writer.SetInput( reader.GetOutput() )
40 # cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
41 #writer.SetInput( cast.GetOutput() )
42 writer.SetFileDimensionality( 3 )
43 writer.Write()
```

29.85 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;
    }
}
```

29.86 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;
    }
}
```

29.87 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;
    }
}
```

29.88 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;
    }
}
```

29.89 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;
    }
}
```

29.90 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 );
    }
}
```

29.91 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];

  // Instanciate 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;
}
```

29.92 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() ); // new
    writer.SetInput( reader.GetOutput() ); // old
    writer.Write();

    return 0;
    }
}
```

29.93 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' eventhough I (MM) would have prefered
// 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 );
    }
}
```

29.94 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.SetInput( reader.GetOutput() );
    cast.SetOutputScalarTypeToUnsignedChar();

    // By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
```

```
    vtkGDCMImageWriter writer = vtkGDCMImageWriter.
      New();
    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();

    return 0;
    }
}
```

29.95 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];

  // Instanciate 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();

  // Contruct 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.
 */
```

29.96 HelloWorld.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 Hello World !
17 """
18
19 import gdcm
20 import sys
21
22 if __name__ == "__main__":
23
24   # verbosity:
25   #gdcm.Trace.DebugOn()
26   #gdcm.Trace.WarningOn()
27   #gdcm.Trace.ErrorOn()
28
29   # Get the filename from the command line
30   filename = sys.argv[1]
31
32   # Instanciate a gdcm.Reader
33   # This is the main class to handle any type of DICOM object
34   # You should check for gdcm.ImageReader for reading specifically DICOM Image file
35   r = gdcm.Reader()
36   r.SetFileName( filename )
37   # If the reader fails to read the file, we should stop !
38   if not r.Read():
39     print "Not a valid DICOM file"
40     sys.exit(1)
41
42   # Get the DICOM File structure
43   file = r.GetFile()
44
45   # Get the DataSet part of the file
46   dataset = file.GetDataSet()
47
48   # Ok let's print it !
49   print dataset
50
51   # Use StringFilter to print a particular Tag:
52   sf = gdcm.StringFilter()
53   sf.SetFile(r.GetFile())
54
55   # Check if Attribute exist
56   print dataset.FindDataElement( gdcm.Tag(0x0028,0x0010))
57
```

```
58   # Let's print it as string pair:
59   print sf.ToStringPair(gdcm.Tag(0x0028,0x0010))
```

29.97 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;
}
```

29.98 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[0], 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[0], (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;
}
```

29.99 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 "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;
}
```

29.100 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;
    }
}
```

29.101 ManipulateFile.py

```
1 ######################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ######################################################################
14
15 """
16 Usage:
17
18  python ManipulateFile.py input.dcm output.dcm
19
20 Footnote:
21   GDCM 1.2.x would create incorrect Multiframes MR Image Storage file. Try to recover from
22   the issues to recreate a MultiframeGrayscaleByteSecondaryCaptureImageStorage file.
23   e.g:
24
25   python ManipulateFile.py Insight/Testing/Temporary/itkGDCMImageIOTest5-j2k.dcm manipulated.dcm
26 """
27
28 import sys
29 import gdcm
```

```
30
31 if __name__ == "__main__":
32
33   file1 = sys.argv[1]
34   file2 = sys.argv[2]
35
36   r = gdcm.Reader()
37   r.SetFileName( file1 )
38   if not r.Read():
39     sys.exit(1)
40
41   ano = gdcm.Anonymizer()
42   ano.SetFile( r.GetFile() )
43   ano.RemovePrivateTags()
44   ano.Remove( gdcm.Tag(0x0032,0x1030) )
45   ano.Remove( gdcm.Tag(0x008,0x14) )
46   ano.Remove( gdcm.Tag(0x008,0x1111) )
47   ano.Remove( gdcm.Tag(0x008,0x1120) )
48   ano.Remove( gdcm.Tag(0x008,0x1140) )
49   ano.Remove( gdcm.Tag(0x10,0x21b0) )
50   ano.Empty( gdcm.Tag(0x10,0x10) )
51   ano.Empty( gdcm.Tag(0x10,0x20) )
52   ano.Empty( gdcm.Tag(0x10,0x30) )
53   ano.Empty( gdcm.Tag(0x20,0x10) )
54   ano.Empty( gdcm.Tag(0x32,0x1032) )
55   ano.Empty( gdcm.Tag(0x32,0x1033) )
56   ano.Empty( gdcm.Tag(0x40,0x241) )
57   ano.Empty( gdcm.Tag(0x40,0x254) )
58   ano.Empty( gdcm.Tag(0x40,0x253) )
59   ano.Empty( gdcm.Tag(0x40,0x1001) )
60   ano.Empty( gdcm.Tag(0x8,0x80) )
61   ano.Empty( gdcm.Tag(0x8,0x50) )
62   ano.Empty( gdcm.Tag(0x8,0x1030) )
63   ano.Empty( gdcm.Tag(0x8,0x103e) )
64   ano.Empty( gdcm.Tag(0x18,0x1030) )
65   ano.Empty( gdcm.Tag(0x38,0x300) )
66   g = gdcm.UIDGenerator()
67   ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
68   ano.Replace( gdcm.Tag(0x0020,0x00d), g.Generate() )
69   ano.Replace( gdcm.Tag(0x0020,0x00e), g.Generate() )
70   ano.Replace( gdcm.Tag(0x0020,0x052), g.Generate() )
71   #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
72   """
73   ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
74   ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
75   ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
76   ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
77   ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
78   ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
79   ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
80   ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
81   ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
82   ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing
83
84   ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
85   ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
86   ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
87   ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator
88
89   ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation
90
91   ano.Empty( gdcm.Tag(0x0020,0x0020) )
92
93   ano.Remove( gdcm.Tag(0x7fe0,0x0000) )
94
95   #ano.Empty( gdcm.Tag(0x0028,0x0009) ) # Frame Increment Pointer
96
97   #ano.Empty( gdcm.Tag(0x0028,0x1052) )  #<entry group="0028" element="1052" vr="DS" vm="1" name="Rescale
       Intercept"/>
98   #ano.Empty( gdcm.Tag(0x0028,0x1053) )  #<entry group="0028" element="1053" vr="DS" vm="1" name="Rescale
       Slope"/>
99   #ano.Replace( gdcm.Tag(0x0028,0x1054), "US" )  #<entry group="0028" element="1054" vr="LO" vm="1" name="
      Rescale Type"/>
100
101   ano.Replace( gdcm.Tag(0x2050, 0x0020), "IDENTITY")
102   """
103
104   w = gdcm.Writer()
105   w.SetFile( ano.GetFile() )
106   w.SetFileName( file2 )
107   if not w.Write():
```

```
108     sys.exit(1)
```

29.102 ManipulateSequence.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 Usage:
17
18  python ManipulateSequence.py input.dcm output.dcm
19
20 This was tested using:
21
22  python ManipulateSequence.py gdcmData/D_CLUNIE_CT1_J2KI.dcm myoutput.dcm
23
24 This is a dummy example on how to modify a value set in a nested-nested dataset
25
26 WARNING:
27 Do not use as-is in production, this is just an example
28 This example works in an undefined length Item only (you need to explicitely recompute the length
    otherwise)
29 """
30
31 import sys
32 import gdcm
33
34 if __name__ == "__main__":
35
36   file1 = sys.argv[1]
37   file2 = sys.argv[2]
38
39   r = gdcm.Reader()
40   r.SetFileName( file1 )
41   if not r.Read():
42     sys.exit(1)
43
44   f = r.GetFile()
45   ds = f.GetDataSet()
46   tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
47   if ds.FindDataElement( tsis ):
48     sis = ds.GetDataElement( tsis )
49     #sqsis = sis.GetSequenceOfItems()
50     # GetValueAsSQ handle more cases
51     sqsis = sis.GetValueAsSQ()
52     if sqsis.GetNumberOfItems():
53       item1 = sqsis.GetItem(1)
54       nestedds = item1.GetNestedDataSet()
55       tprcs = gdcm.Tag(0x0040,0xa170) # PurposeOfReferenceCodeSequence
56       if nestedds.FindDataElement( tprcs ):
57         prcs = nestedds.GetDataElement( tprcs )
58         sqprcs = prcs.GetSequenceOfItems()
59         if sqprcs.GetNumberOfItems():
60           item2 = sqprcs.GetItem(1)
61           nestedds2 = item2.GetNestedDataSet()
62           # (0008,0104) LO [Uncompressed predecessor]               #  24, 1 CodeMeaning
63           tcm = gdcm.Tag(0x0008,0x0104)
64           if nestedds2.FindDataElement( tcm ):
65             cm = nestedds2.GetDataElement( tcm )
66             mystr = "GDCM was here"
67             cm.SetByteValue( mystr, gdcm.VL( len(mystr) ) )
68
69   w = gdcm.Writer()
70   w.SetFile( f )
71   w.SetFileName( file2 )
72   if not w.Write():
```

```
73     sys.exit(1)
```

29.103 MergeFile.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 Usage:
17
18  python MergeFile.py input1.dcm input2.dcm
19
20   It will produce a 'merge.dcm' output file, which contains all meta information from input1.dcm
21   and copy the Stored Pixel values from input2.dcm
22   This script even works when input2.dcm is a Secondary Capture and does not contains information
23   such as IOP and IPP...
24 """
25
26 import sys
27 import gdcm
28
29 if __name__ == "__main__":
30
31   file1 = sys.argv[1]
32   file2 = sys.argv[2]
33
34   r1 = gdcm.ImageReader()
35   r1.SetFileName( file1 )
36   if not r1.Read():
37     sys.exit(1)
38
39   r2 = gdcm.ImageReader()
40   r2.SetFileName( file2 )
41   if not r2.Read():
42     sys.exit(1)
43
44   # Image from r2 could be Secondary Capture and thus would not contains neither IPP nor IOP
45   # Instead always prefer to only copy the Raw Data Element.
46   # Warning ! Image need to be identical ! Only the value of Stored Pixel can be different.
47   r1.GetImage().SetDataElement( r2.GetImage().GetDataElement() )
48
49   w = gdcm.ImageWriter()
50   w.SetFile( r1.GetFile() )
51   #w.SetImage( r2.GetImage() )  # See comment above
52   w.SetImage( r1.GetImage() )
53
54   w.SetFileName( "merge.dcm" )
55   if not w.Write():
56     sys.exit(1)
57
58   sys.exit(0)
```

29.104 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;
}
```

29.105 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;
    }
}
```

29.106 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.SetInput(change.GetOutput());

    //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();
    }
}
```

29.107 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;
    }
}
```

29.108 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();
    }
}
```

29.109 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 wextent[] = image.GetWholeExtent();
    int xMin = wextent[0];
    int xMax = wextent[1];
    int yMin = wextent[2];
    int yMax = wextent[3];
    int zMin = wextent[4];
    int zMax = wextent[5];
```

```
  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.SetInput(change.GetOutput());
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.SetInput(change.GetOutput());
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.SetInput(change.GetOutput());
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 );
    }
}
```

29.110 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                         = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004658"
tReferenceImage1                         = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004635"
tReferenceImage2                         = "1.3.12.2.1107.5.2.9.16041.30000007062106100181200004683"
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.""T1mapFunctor"">
      {
        <Class> ""T1mapFunctor@IceImagePostProcFunctors""

        <ParamBool.""EXECUTE"">  { }
        <ParamDouble.""Flip1_deg"">  { <Precision> 16  14.7378520000000000  }
...

*/
  // Below is an attemp 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;
    }

  return 0;
}
```

29.111 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;
    }
}
```

29.112 NewSequence.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 Usage:
17
18   python NewSequence.py input.dcm output.dcm
19
20
21 Thanks to Robert Irie for code
22 """
23
24 import sys
25 import gdcm
26
27 if __name__ == "__main__":
28
29   file1 = sys.argv[1]
30   file2 = sys.argv[2]
31
32   r = gdcm.Reader()
33   r.SetFileName( file1 )
```

```
34   if not r.Read():
35     sys.exit(1)
36
37   f = r.GetFile()
38   ds = f.GetDataSet()
39   #tsis = gdcm.Tag(0x0008,0x2112) # SourceImageSequence
40
41   # Create a dataelement
42   de = gdcm.DataElement(gdcm.Tag(0x0010, 0x2180))
43   de.SetByteValue("Occupation", gdcm.VL(len("Occupation")))
44   de.SetVR(gdcm.VR(gdcm.VR.SH))
45
46   # Create an item
47   it=gdcm.Item()
48   it.SetVLToUndefined()       # Needed to not popup error message
49   #it.InsertDataElement(de)
50   nds=it.GetNestedDataSet()
51   nds.Insert(de)
52
53   # Create a Sequence
54   sq=gdcm.SequenceOfItems().New()
55   sq.SetLengthToUndefined()
56   sq.AddItem(it)
57
58   # Insert sequence into data set
59   des=gdcm.DataElement(gdcm.Tag(0x0400,0x0550))
60   des.SetVR(gdcm.VR(gdcm.VR.SQ))
61   des.SetValue(sq.__ref__())
62   des.SetVLToUndefined()
63
64   ds.Insert(des)
65
66   w = gdcm.Writer()
67   w.SetFile( f )
68   w.SetFileName( file2 )
69   if not w.Write():
70     sys.exit(1)
```

29.113 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"

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;
}
```

29.114 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 << "Sucess to rewrite image !" << std::endl;
  image.Print( std::cout );
  return 0;
}
```

29.115 PhilipsPrivateRescaleInterceptSlope.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 """
16 Usage:
17
18  python
19 """
20
21 import gdcm
22 import sys
23
24 filename = sys.argv[1]
25 tmpfile = "/tmp/philips_rescaled.dcm"
26
27
28 # Need to access some private tags, read the file :
29 reader = gdcm.Reader()
30 reader.SetFileName( filename )
31 if not reader.Read():
32   sys.exit(1)
33
34 ds = reader.GetFile().GetDataSet()
35
36 #print ds
37 # (2005,1409)     DS      4       0.0
38 # (2005,140a)     DS      16      1.52283272283272
39
40 # (2005,0014)     LO      26      Philips MR Imaging DD 005
41 tag1 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
42 tag2 = gdcm.PrivateTag(0x2005,0x0a,"Philips MR Imaging DD 005")
43 print tag1
44 print tag2
45
46 # make sure to do a copy, we want the private tag to remain
47 # otherwise gdcm gives us a reference
48 el1 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
49 print el1
50 el2 = gdcm.DataElement( ds.GetDataElement( tag2 ) )
51 print el2
52
```

```
# (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"
```

29.116 PlaySound.py

```
################################################################################
#
#  Program: GDCM (Grassroots DICOM). A DICOM library
#
#  Copyright (c) 2006-2011 Mathieu Malaterre
#  All rights reserved.
#  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
#
#     This software is distributed 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:

 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)
```

```
58
59 #print bv.GetPointer()
60 print bv.GetLength()
61 l = 116838
62
63 file='test.wav'
64 myfile = open(file, "wb")
65 s = bv.GetPointer()
66 for i in range(0, l):
67   myfile.write(s[i])
68 myfile.close()
69
70 # http://mail.python.org/pipermail/python-list/2004-October/288905.html
71 if sys.platform.startswith('win'):
72    from winsound import PlaySound, SND_FILENAME, SND_ASYNC
73    PlaySound(file, SND_FILENAME|SND_ASYNC)
74 elif sys.platform.find('linux')>-1:
75    from wave import open as waveOpen
76    from ossaudiodev import open as ossOpen
77    s = waveOpen(file,'rb')
78    (nc,sw,fr,nf,comptype, compname) = s.getparams( )
79    dsp = ossOpen('/dev/dsp','w')
80    try:
81      from ossaudiodev import AFMT_S16_NE
82    except ImportError:
83      if byteorder == "little":
84        AFMT_S16_NE = ossaudiodev.AFMT_S16_LE
85      else:
86        AFMT_S16_NE = ossaudiodev.AFMT_S16_BE
87    dsp.setparameters(AFMT_S16_NE, nc, fr)
88    data = s.readframes(nf)
89    s.close()
90    dsp.write(data)
91    dsp.close()
```

29.117 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
};

byte* src = (byte*)data_in;
byte* dest = (byte*)&new_stream[0];
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 ther 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 swithing 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[0], (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;
}
```

29.118 PrivateDict.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 """
```

```
17
18 import gdcm
19 import sys,os
20
21 if __name__ == "__main__":
22   #gdcm.Trace.DebugOn()
23   globInst = gdcm.Global.GetInstance()
24   # Try to load Part3.xml file
25   # This fils is too big for being accessible directly at runtime.
26   globInst.LoadResourcesFiles()
27
28
29   # Get a private tag from the runtime dicts. LoadResourcesFiles could
30   # have failed but this has no impact on the private dict
31
32   d = globInst.GetDicts()
33   print d.GetDictEntry( gdcm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
34   pd = d.GetPrivateDict()
35   print pd.GetDictEntry( gdcm.PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER") )
```

29.119 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 differents way 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 = 0;
  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;
}
```

29.120 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;
}
```

29.121 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;
```

}

29.122 ReadAndDumpDICOMDIR.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 #  File: ReadAndDumpDICOMDIR.py
14 #
15 #  Author: Lukas Batteau (lbatteau gmail)
16 #
17 #  This example shows how to read and dump a DICOMDIR File.
18 #  Based on Tom Marynowski's (lordglub gmail) example.
19 #
20 #  Usage:
21 #  python ReadAndDumpDICOMDIR.py [DICOMDIR file]
22 ############################################################################
23
24
25
26 import sys
27 import gdcm
28
29 if __name__ == "__main__":
30     # Check arguments
31     if (len(sys.argv) < 2):
32         # No filename passed
33         print "No input filename found"
34         quit()
35
36     filename = sys.argv[1]
37
38
39     # Read file
40     reader = gdcm.Reader()
41     reader.SetFileName(filename)
42     if (not reader.Read()):
43         print "Unable to read %s" % (filename)
44         quit()
45
46     file = reader.GetFile()
47
48     # Retrieve header information
49     fileMetaInformation = file.GetHeader()
50     print fileMetaInformation
51
52     # Retrieve data set
53     dataSet = file.GetDataSet()
54     #print dataSet
55
56     # Check media storage
57     mediaStorage = gdcm.MediaStorage()
58     mediaStorage.SetFromFile(file)
59     if (gdcm.MediaStorage.GetMSType(str(mediaStorage)) !=
    gdcm.MediaStorage.MediaStorageDirectoryStorage):
60         # File is not a DICOMDIR
61         print "This file is not a DICOMDIR (Media storage type: %s)" % (str(mediaStorage))
62         quit()
63
64     # Check Media Storage SOP Class
65     if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
66         sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0x0002, 0x0002)).GetValue())
67         # Check SOP UID
68         if (sopClassUid != "1.2.840.10008.1.3.10"):
69             # File is not a DICOMDIR
70             print "This file is not a DICOMDIR"
71     else:
72         # Not present
```

```
73          print "Media Storage SOP Class not present"
74          quit()
75
76      # Iterate through the DICOMDIR data set
77      iterator = dataSet.GetDES().begin()
78      while (not iterator.equal(dataSet.GetDES().end())):
79          dataElement = iterator.next()
80
81          # Check the element tag
82          if (dataElement.GetTag() == gdcm.Tag(0x004, 0x1220)):
83              # The 'Directory Record Sequence' element
84              sequence = dataElement.GetValueAsSQ()
85
86              # Loop through the sequence items
87              itemNr = 1
88              while (itemNr < sequence.GetNumberOfItems()):
89                  item = sequence.GetItem(itemNr)
90
91                  # Check the element tag
92                  if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
93                      # The 'Directory Record Type' element
94                      value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
95
96                      # PATIENT
97                      while (value.strip() == "PATIENT"):
98                          print value.strip()
99                          # Print patient name
100                         if (item.FindDataElement(gdcm.Tag(0x0010, 0x0010))):
101                             value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0010)).GetValue())
102                             print value
103
104                         # Print patient ID
105                         if (item.FindDataElement(gdcm.Tag(0x0010, 0x0020))):
106                             value = str(item.GetDataElement(gdcm.Tag(0x0010, 0x0020)).GetValue())
107                             print value
108
109                         # Next
110                         itemNr = itemNr + 1
111                         item = sequence.GetItem(itemNr)
112                         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
113                             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).GetValue())
114
115                         # STUDY
116                         while (value.strip() == "STUDY"):
117                             print value.strip()
118
119                             # Print study UID
120                             if (item.FindDataElement(gdcm.Tag(0x0020, 0x000d))):
121                                 value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000d)).GetValue(
    ))
122                                 print value
123
124                             # Print study date
125                             if (item.FindDataElement(gdcm.Tag(0x0008, 0x0020))):
126                                 value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0020)).GetValue(
    ))
127                                 print value
128
129                             # Print study description
130                             if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
131                                 value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue(
    ))
132                                 print value
133
134                             # Next
135                             itemNr = itemNr + 1
136                             item = sequence.GetItem(itemNr)
137                             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
138                                     value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).
    GetValue())
139
140                             # SERIES
141                             while (value.strip() == "SERIES"):
142                                 print value.strip()
143
144                                 # Print series UID
145                                 if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
146                                     value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).
    GetValue())
147                                     print value
148
```

```
149                                 # Print series modality
150                                 if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
151                                     value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060)).
      GetValue())
152                                     print "Modality"
153                                     print value
154
155                                 # Print series description
156                                 if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
157                                     value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e)).
      GetValue())
158                                     print "Description"
159                                     print value
160
161                                 # Next
162                                 itemNr = itemNr + 1
163                                 item = sequence.GetItem(itemNr)
164                                 if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
165                                     value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).
      GetValue())
166
167                                 # IMAGE
168                                 while (value.strip() == "IMAGE"):
169                                     print value.strip()
170
171                                     # Print image UID
172                                     if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
173                                         value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1511)).
      GetValue())
174                                         print value
175
176                                     # Next
177                                     if (itemNr < sequence.GetNumberOfItems()):
178                                         itemNr = itemNr + 1
179                                     else:
180                                         break
181
182                                     item = sequence.GetItem(itemNr)
183                                     if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
184                                         value = str(item.GetDataElement(
      gdcm.Tag(0x0004, 0x1430)).GetValue())
185
186                 # Next
187                 itemNr = itemNr + 1
```

29.123 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];

  // Instanciate 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;
}
```

29.124 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 "gdcmReader.h"
#include "gdcmImplicitDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmByteValue.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;
}
```

29.125 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);
      }
    }
}
```

29.126 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;
}
```

29.127 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)
      {
      std::cerr << "BAD ALLOC Exception caught!" << std::endl;
      }
    catch (...)
      {
      std::cerr << "Exception caught!" << std::endl;
      }
    }

  return 0;
}
```

29.128 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' eventhough I (MM) would have prefered
// 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.SetInput( reader.GetOutput() );
    writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
    writer.Write();

    System.out.println("Success writing: " + writer.GetFileName() );
    }
}
```

29.129 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 explicitely 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;
}
```

29.130 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;
    }
}
```

29.131 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 pupose of this execise 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;
    }
}
```

29.132 RemovePrivateTags.py

```
1 ######################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
```

```
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 Usage:
17
18  python RemovePrivateTags.py input.dcm output.dcm
19 """
20
21 import sys
22 import gdcm
23
24
25 if __name__ == "__main__":
26
27   file1 = sys.argv[1]
28   file2 = sys.argv[2]
29
30   # Instanciate the reader.
31   r = gdcm.Reader()
32   r.SetFileName( file1 )
33   if not r.Read():
34     sys.exit(1)
35
36   # Remove private tags
37   ano = gdcm.Anonymizer()
38   ano.SetFile( r.GetFile() )
39   if not ano.RemovePrivateTags():
40     sys.exit(1)
41
42   # Write DICOM file
43   w = gdcm.Writer()
44   w.SetFile( ano.GetFile() )
45   #w.CheckFileMetaInformationOff() # Do not attempt to check meta header
46   w.SetFileName( file2 )
47   if not w.Write():
48     sys.exit(1)
49
50   # It is usually a good idea to exit the script with an error, as gdcm does not remove partial (incorrect)
       DICOM file
51   # (application level)
```

29.133 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;
    }
}
```

29.134 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 <strstream>
#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 "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();

  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::strstream 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());

        _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;
}
```

29.135 ReWriteSCAsMR.py

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 GDCM 1.x would write out MR Image Storage as Secondary Capture Object while still setting Rescale
       Slope/Intercept
17 and saving the Pixel Spacing in (0028,0030)
18 """
19
20 import gdcm
21 import sys,os
22
23 def CheckSecondaryCaptureObjectIsMRImageStorage(r):
24   ds = r.GetFile().GetDataSet()
25   # Check Source Image Sequence
26   if ds.FindDataElement( gdcm.Tag(0x0008,0x2112) ):
27     sis = ds.GetDataElement( gdcm.Tag(0x0008,0x2112) )
28     sqsis = sis.GetSequenceOfItems()
29     if sqsis.GetNumberOfItems():
30       item1 = sqsis.GetItem(1)
31       nestedds = item1.GetNestedDataSet()
32       if nestedds.FindDataElement( gdcm.Tag(0x0008,0x1150) ):
33         ReferencedSOPClassUID = nestedds.GetDataElement( gdcm.Tag(0x0008,0x1150) )
34         raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
35         uids = gdcm.UIDs()
36         # what is the actual object we are looking at ?
37         ms = gdcm.MediaStorage()
38         ms.SetFromDataSet(ds)
39         msuid = ms.GetString()
40         uids.SetFromUID( msuid )
41         msuidname = uids.GetName() # real Media Storage Name
42         uids.SetFromUID( raw )
43         sqmsuidname = uids.GetName() # Source Image Sequence Media Storage Name
44         # If object is SC and Source derivation is MRImageStorage then we can assume 'Pixel Spacing' is
       correct
45         if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary Capture Image Storage' ):
46           return True
```

```
47   # in all other case simply return the currentspacing:
48   return False
49
50 if __name__ == "__main__":
51   r = gdcm.ImageReader()
52   filename = sys.argv[1]
53   r.SetFileName( filename )
54   if not r.Read():
55     sys.exit(1)
56   f = r.GetFile()
57
58   if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
59     # Special handling of the spacing:
60     # GDCM 1.2.0 would not rewrite correcly DICOM Object and would always set them as 'Secondary Capture
       Image Storage'
61     # while we would rather have 'MR Image Storage'
62     gdcm.ImageHelper.SetForcePixelSpacing( True )
63     mrspacing = gdcm.ImageHelper.GetSpacingValue( r.GetFile() )
64     # TODO: I cannot do simply the following:
65     #image.SetSpacing( mrspacing )
66     image.SetSpacing(0, mrspacing[0] )
67     image.SetSpacing(1, mrspacing[1] )
68     image.SetSpacing(2, mrspacing[2] )
69     gdcm.ImageHelper.SetForceRescaleInterceptSlope( True )
70     ris = gdcm.ImageHelper.GetRescaleInterceptSlopeValue(
      r.GetFile() )
71     image.SetIntercept( ris[0] )
72     image.SetSlope( ris[1] )
73
74   outfilename = sys.argv[2]
75   w = gdcm.ImageWriter()
76   w.SetFileName( outfilename )
77   w.SetFile( r.GetFile() )
78   w.SetImage( image )
79   if not w.Write():
80     sys.exit(1)
81
82   sys.exit(0)
```

29.136 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;

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<0x0028,0x0010> at1;
at1.SetFromDataSet( ds );
gdcm::Attribute<0x0028,0x0011> at2;
at2.SetFromDataSet( ds );

gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
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[0], (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
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 = "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;
}
```

29.137 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"

/*
 * 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;
}
```

29.138 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:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ScanDirectory.exe /path/to/gdcmData/
 */
using System;
using gdcm;

public class ScanDirectory
{
  public static int Main(string[] args)
    {
    string directory = args[0];
    Tag t = new Tag(0x8,0x8);

    Directory d = new Directory();
    uint nfiles = d.Load( directory );
    if(nfiles == 0) return 1;
    //System.Console.WriteLine( "Files:\n" + d.toString() );

    //Scanner s = new Scanner();
    SmartPtrScan sscan = Scanner.New();
    Scanner s = sscan.__ref__();
    SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
    s.AddTag( t );
    bool b = s.Scan( d.GetFilenames() );
    if(!b) return 1;

    System.Console.WriteLine( "Scan:\n" + s.toString() );

    System.Console.WriteLine( "success" );
    return 0;
    }
}
```

29.139 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");
  }

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" );
    }
}
```

29.140 ScanDirectory.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
14
15 import gdcm
16 import sys,os
17
18 class ProgressWatcher(gdcm.SimpleSubjectWatcher):
19   def ShowProgress(self, sender, event):
20     pe = gdcm.ProgressEvent.Cast(event)
21     print pe.GetProgress()
22   def EndFilter(self):
23     print "Yay ! I am done"
24
25 if __name__ == "__main__":
26   directory = sys.argv[1]
27
28   # Define the set of tags we are interested in
29   t1 = gdcm.Tag(0x8,0x8);
30   t2 = gdcm.Tag(0x10,0x10);
31
32   # Iterate over directory
33   d = gdcm.Directory();
34   nfiles = d.Load( directory );
35   if(nfiles == 0): sys.exit(1);
36   # System.Console.WriteLine( "Files:\n" + d.toString() );
37
38   filenames = d.GetFilenames()
39
40   #  Get rid of any Warning while parsing the DICOM files
41   gdcm.Trace.WarningOff()
42
43   # instanciate Scanner:
44   sp = gdcm.Scanner.New();
45   s = sp.__ref__()
46   w = ProgressWatcher(s, 'Watcher')
47
48   s.AddTag( t1 );
49   s.AddTag( t2 );
50   b = s.Scan( filenames );
51   if(not b): sys.exit(1);
52
53   print "success" ;
54   #print s
55
56   pttv = gdcm.PythonTagToValue( s.GetMapping( filenames[1] ) )
57   pttv.Start()
58   # iterate until the end:
59   while( not pttv.IsAtEnd() ):
60     # get current value for tag and associated value:
61     # if tag was not found, then it was simply not added to the internal std::map
62     # Warning value can be None
63     tag = pttv.GetCurrentTag()
64     value = pttv.GetCurrentValue()
65     print tag,"->",value
66     # increment iterator
67     pttv.Next()
```

```
68
69   sys.exit(0)
```

29.141 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;
    }
}
```

29.142 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;
    }
}
```

29.143 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;
      }
    }
}
```

29.144 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 "gdcmScanner.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)
    {
    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::Scanner> sp = new
      gdcm::Scanner;
  gdcm::Scanner &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),
  };
  s.AddTag( tag_array[0] );
  s.AddTag( tag_array[1] );
  s.AddTag( tag_array[2] );

  gdcm::Directory::FilenamesType filenames;
  filenames.push_back( filename );
  filenames.push_back( filename_invalid );

  if( !s.Scan( filenames ) )
    {
    return 1;
    }

  //s.Print( std::cout );


  if( s.IsKey( filename ) )
    {
    std::cout << "INFO:" << filename << " is a proper Key for the Scanner (this is a DICOM file)" <<
      std::endl;
    }

  if( !s.IsKey( filename_invalid ) )
    {
    std::cout << "INFO:" << filename_invalid << " is not a proper Key for the Scanner (this is either not a
       DICOM file or file does not exist)" << std::endl;
    }

  gdcm::Scanner::TagToValue const &ttv = s.GetMapping(filename);

  const gdcm::Tag *ptag = tag_array;
  for( ; ptag != tag_array + 3; ++ptag )
    {
    gdcm::Scanner::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;
}
```

29.145 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;
}
```

29.146 SortImage.py

```
1 ############################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ############################################################################
```

```
14
15 """
16 Usage:
17
18  python SortImage.py dirname
19 """
20
21 import gdcm
22 import sys
23
24 def PrintProgress(object, event):
25   assert event == "ProgressEvent"
26   print "Progress:", object.GetProgress()
27
28 def MySort(ds1, ds2):
29   # compare ds1
30   return False
31
32 if __name__ == "__main__":
33
34   dirname = sys.argv[1]
35   d = gdcm.Directory()
36   d.Load( dirname )
37
38   print d
39
40   sorter = gdcm.Sorter()
41   sorter.SetSortFunction( MySort )
42   #sorter.AddObserver( "ProgressEvent", PrintProgress )
43   sorter.Sort( d.GetFilenames() )
44
45   print "Sorter:"
46   print sorter
```

29.147 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;
    }
}
```

29.148 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 alread 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;
    }
}
```

29.149 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;
}
```

29.150 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 <string.h> // memcpy

int myfunc()
{
  char vl_str[4];
  const char raw[] = "\000\000\000\004";
  memcpy(vl_str, raw, 4);
  uint32_t vl;
  gdcm::ByteSwap<uint32_t>::SwapRangeFromSwapCodeIntoSystem
      ((uint32_t*)(&vl_str), gdcm::SwapCode::BigEndian, 1);
  memcpy(&vl, vl_str, 4);
  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);
  std::cout << std::hex << "vl: " << vl << std::endl;
  if( vl != 0x4000000 )
    {
    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 )
  {
  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,2);
   if ( array[0] != 0x3412 )
     {
     return 1;
     }

  return 0;
}
```

29.151 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;
}
```

29.152 TestReader.py

This is a C++ example on how to use gdcm::Reader

```
1 ##########################################################################
2 #
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 import gdcm
16 import os,sys
17
18 def TestRead(filename, verbose = False):
19   r = gdcm.Reader()
20   r.SetFileName( filename )
21   sucess = r.Read()
22   #if verbose: print r.GetFile()
23   if verbose: print(r.GetFile().GetDataSet())
24   return sucess
25
26 if __name__ == "__main__":
27   sucess = 0
28   try:
29     filename = os.sys.argv[1]
30     sucess += TestRead( filename, True )
31   except:
32     # loop over all files:
33     gdcm.Trace.DebugOff()
34     gdcm.Trace.WarningOff()
35     t = gdcm.Testing()
36     nfiles = t.GetNumberOfFileNames()
37     for i in range(0,nfiles):
38       filename = t.GetFileName(i)
39       sucess += TestRead( filename )
40
41
```

```
42   # Test succeed ?
43   sys.exit(sucess == 0)
```

29.153 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 <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;
}
```

29.154 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;
}
```

29.155 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(1)
    {
    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 );
    }
}
```

29.156 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.c_str(), 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;
}
```

29.157 WriteBuffer.py

```
1 ##########################################################################
2 #
```

```
3 #  Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #  Copyright (c) 2006-2011 Mathieu Malaterre
6 #  All rights reserved.
7 #  See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #     This software is distributed WITHOUT ANY WARRANTY; without even
10 #     the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 ##########################################################################
14
15 """
16 Usage:
17
18 http://chuckhahm.com/Ischem/Zurich/XX_0134
19
20 (2005,1132) SQ (Sequence with undefined length #=8)     # u/l, 1 Unknown Tag & Data
21   (fffe,e000) na (Item with undefined length #=9)         # u/l, 1 Item
22     (2005,0011) LO [Philips MR Imaging DD 002]              #  26, 1 PrivateCreator
23     (2005,1137) PN [PDF_CONTROL_GEN_PARS]                   #  20, 1 Unknown Tag & Data
24     (2005,1138) PN (no value available)                     #   0, 0 Unknown Tag & Data
25     (2005,1139) PN [IEEE_PDF]                               #   8, 1 Unknown Tag & Data
26     (2005,1140) PN (no value available)                     #   0, 0 Unknown Tag & Data
27     (2005,1141) PN (no value available)                     #   0, 0 Unknown Tag & Data
28     (2005,1143) SL 3103                                     #   4, 1 Unknown Tag & Data
29     (2005,1144) OW 0566\0000\013b\0000\0a4a\0000\000e\0000\0a7a\0000\0195\0000\0008... # 3104, 1 Unknown
       Tag & Data
30     (2005,1147) CS [Y]                                      #   2, 1 Unknown Tag & Data
31   (fffe,e00d) na (ItemDelimitationItem)                   #   0, 0 ItemDelimitationItem
32   (fffe,e000) na (Item with undefined length #=9)         # u/l, 1 Item
33     (2005,0011) LO [Philips MR Imaging DD 002]              #  26, 1 PrivateCreator
34     (2005,1137) PN [PDF_CONTROL_PREP_PARS]                  #  22, 1 Unknown Tag & Data
35     (2005,1138) PN (no value available)                     #   0, 0 Unknown Tag & Data
36     (2005,1139) PN [IEEE_PDF]                               #   8, 1 Unknown Tag & Data
37     (2005,1140) PN (no value available)                     #   0, 0 Unknown Tag & Data
38     (2005,1141) PN (no value available)                     #   0, 0 Unknown Tag & Data
39     (2005,1143) SL 7934                                     #   4, 1 Unknown Tag & Data
40     (2005,1144) OW 19b6\0000\005f\0000\1b2a\0000\00f3\0000\1eee\0000\0000\0000\0008... # 7934, 1 Unknown
       Tag & Data
41     (2005,1147) CS [Y]                                      #   2, 1 Unknown Tag & Data
42   (fffe,e00d) na (ItemDelimitationItem)                   #   0, 0 ItemDelimitationItem
43 ...
44 """
45
46 import sys
47 import gdcm
48
49 if __name__ == "__main__":
50
51   file1 = sys.argv[1]
52   file2 = sys.argv[2]
53
54   r = gdcm.Reader()
55   r.SetFileName( file1 )
56   if not r.Read():
57     sys.exit(1)
58
59   fg = gdcm.FilenameGenerator()
60   f = r.GetFile()
61   ds = f.GetDataSet()
62   tsis = gdcm.Tag(0x2005,0x1132) #
63   if ds.FindDataElement( tsis ):
64     sis = ds.GetDataElement( tsis )
65     #sqsis = sis.GetSequenceOfItems()
66     # GetValueAsSQ handle more cases
67     sqsis = sis.GetValueAsSQ()
68     if sqsis.GetNumberOfItems():
69       nitems = sqsis.GetNumberOfItems();
70       fg.SetNumberOfFilenames( nitems )
71       fg.SetPrefix( file2 )
72       if not fg.Generate():
73         print "problem"
74         sys.exit(1)
75       for i in range(0,nitems):
76         item1 = sqsis.GetItem(i+1) # Item start at 1
77         nestedds = item1.GetNestedDataSet()
78         tprcs = gdcm.Tag(0x2005,0x1144) #
79         if nestedds.FindDataElement( tprcs ):
80           prcs = nestedds.GetDataElement( tprcs )
81           bv = prcs.GetByteValue()
```

```
82          print bv
83          f = open( fg.GetFilename(i) , "w" )
84          f.write( bv.WriteBuffer() )
```

Index

AE
  gdcm::VR, 854
AES128_CIPHER
  gdcm::CryptographicMessageSyntax, 261
AES192_CIPHER
  gdcm::CryptographicMessageSyntax, 262
AES256_CIPHER
  gdcm::CryptographicMessageSyntax, 262
ALGOType_END
  gdcm::Segment, 651
ARGB
  gdcm::PhotometricInterpretation, 572
AS
  gdcm::VR, 854
AT
  gdcm::VR, 854
AUTOMATIC
  gdcm::Segment, 651
AXIAL
  gdcm::Orientation, 546
AmbulatoryECGWaveformStorage
  gdcm::MediaStorage, 510
  gdcm::UIDs, 776
Audio
  gdcm::MediaStorage, 511
AudioSRStorageTrialRetired
  gdcm::UIDs, 777

BLUE
  gdcm::LookupTable, 498
backslash
  gdcm, 126
BadBigEndian
  gdcm::SwapCode, 731
BadLittleEndian
  gdcm::SwapCode, 731
BasicAnnotationBoxSOPClass
  gdcm::UIDs, 775
BasicColorImageBoxSOPClass
  gdcm::UIDs, 775
BasicColorPrintManagementMetaSOPClass
  gdcm::UIDs, 775
BasicFilmBoxSOPClass
  gdcm::UIDs, 775
BasicFilmSessionSOPClass
  gdcm::UIDs, 775
BasicGrayscaleImageBoxSOPClass
  gdcm::UIDs, 775
BasicGrayscalePrintManagementMetaSOPClass
  gdcm::UIDs, 775
BasicPrintImageOverlayBoxSOPClassRetired
  gdcm::UIDs, 776
BasicStudyContentNotificationSOPClassRetired
  gdcm::UIDs, 775
BasicTextSR
  gdcm::MediaStorage, 510
BasicTextSRStorage
  gdcm::UIDs, 777
BasicVoiceAudioWaveformStorage
  gdcm::MediaStorage, 510
  gdcm::UIDs, 776
BigEndian
  gdcm::SwapCode, 731
black
  gdcm::terminal, 137
BlendingSoftcopyPresentationStateStorageSOPClass
  gdcm::UIDs, 777
blink
  gdcm::terminal, 137
blue
  gdcm::terminal, 138
BreastImagingRelevantPatientInformationQuery
  gdcm::UIDs, 778
BreastTomosynthesisImageStorage
  gdcm::MediaStorage, 511
  gdcm::UIDs, 780
bright
  gdcm::terminal, 137

C_CANCEL_RQ
  gdcm::network::DIMSE, 325
C_ECHO_RQ
  gdcm::network::DIMSE, 325
C_ECHO_RSP
  gdcm::network::DIMSE, 325
C_FIND_RQ
  gdcm::network::DIMSE, 324
C_FIND_RSP
  gdcm::network::DIMSE, 324
C_GET_RQ
  gdcm::network::DIMSE, 324
C_GET_RSP

gdcm::network::DIMSE, 324
C_MOVE_RQ
gdcm::network::DIMSE, 324
C_MOVE_RSP
gdcm::network::DIMSE, 325
C_STORE_RQ
gdcm::network::DIMSE, 324
C_STORE_RSP
gdcm::network::DIMSE, 324
CALIBRATED
gdcm::Spacing, 696
CAPI
gdcm::CryptoFactory, 260
CMYK
gdcm::PhotometricInterpretation, 572
CONDENSED_STYLE
gdcm::Printer, 604
CONSOLE
gdcm::terminal, 138
CORONAL
gdcm::Orientation, 546
CS
gdcm::VR, 854
CSANonImageStorage
gdcm::MediaStorage, 510
CT_private_ELE
gdcm::TransferSyntax, 761
CTImageStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
CardiacElectrophysiologyWaveformStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 776
CardiacRelevantPatientInformationQuery
gdcm::UIDs, 779
ChestCADSRStorage
gdcm::UIDs, 778
ColorSoftcopyPresentationStateStorageSOPClass
gdcm::UIDs, 777
ComprehensiveSR
gdcm::MediaStorage, 510
ComprehensiveSRStorage
gdcm::UIDs, 777
ComprehensiveSRStorageTrialRetired
gdcm::UIDs, 777
ComputedRadiographyImageStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
Conditional
gdcm::Usage, 837
cyan
gdcm::terminal, 138

DA
gdcm::VR, 854
DATASET_FORMAT
gdcm::CSAHeader, 269
DEFAULT
gdcm::CryptoFactory, 260
DES3_CIPHER
gdcm::CryptographicMessageSyntax, 261
DETECTOR
gdcm::Spacing, 696
DICOMApplicationContextName
gdcm::UIDs, 775
DICOMControlledTerminology
gdcm::UIDs, 775
DICOMUIDRegistry
gdcm::UIDs, 775
DICT_DEBUG
gdcm::DictConverter, 316
DICT_DEFAULT
gdcm::DictConverter, 316
DICT_XML
gdcm::DictConverter, 316
DS
gdcm::VR, 855
DT
gdcm::VR, 855
DeflatedExplicitVRLittleEndian
gdcm::TransferSyntax, 760
gdcm::UIDs, 773
DeformableSpatialRegistrationStorage
gdcm::UIDs, 777
DetachedInterpretationManagementSOPClassRetired
gdcm::UIDs, 775
DetachedPatientManagementMetaSOPClassRetired
gdcm::UIDs, 775
DetachedPatientManagementSOPClass
gdcm::MediaStorage, 510
DetachedPatientManagementSOPClassRetired
gdcm::UIDs, 775
DetachedResultsManagementMetaSOPClassRetired
gdcm::UIDs, 775
DetachedResultsManagementSOPClassRetired
gdcm::UIDs, 775
DetachedStudyManagementMetaSOPClassRetired
gdcm::UIDs, 775
DetachedStudyManagementSOPClass
gdcm::MediaStorage, 510
DetachedStudyManagementSOPClassRetired
gdcm::UIDs, 775
DetachedVisitManagementSOPClass
gdcm::MediaStorage, 510
DetachedVisitManagementSOPClassRetired
gdcm::UIDs, 775
DetailSRStorageTrialRetired
gdcm::UIDs, 777

dicomAETitle
gdcm::UIDs, 779
dicomApplicationCluster
gdcm::UIDs, 779
dicomAssociationAcceptor
gdcm::UIDs, 779
dicomAssociationInitiator
gdcm::UIDs, 779
dicomAuthorizedNodeCertificateReference
gdcm::UIDs, 779
dicomConfigurationRoot
gdcm::UIDs, 779
dicomDescription
gdcm::UIDs, 779
dicomDevice
gdcm::UIDs, 779
dicomDeviceName
gdcm::UIDs, 779
dicomDeviceSerialNumber
gdcm::UIDs, 779
dicomDevicesRoot
gdcm::UIDs, 779
dicomHostname
gdcm::UIDs, 779
dicomInstalled
gdcm::UIDs, 779
dicomInstitutionAddress
gdcm::UIDs, 779
dicomInstitutionDepartmentName
gdcm::UIDs, 779
dicomInstitutionName
gdcm::UIDs, 779
dicomIssuerOfPatientID
gdcm::UIDs, 779
dicomManufacturer
gdcm::UIDs, 779
dicomManufacturerModelName
gdcm::UIDs, 779
dicomNetworkAE
gdcm::UIDs, 779
dicomNetworkConnection
gdcm::UIDs, 780
dicomNetworkConnectionReference
gdcm::UIDs, 779
dicomPort
gdcm::UIDs, 779
dicomPreferredCalledAETitle
gdcm::UIDs, 779
dicomPreferredCallingAETitle
gdcm::UIDs, 779
dicomPrimaryDeviceType
gdcm::UIDs, 779
dicomRelatedDeviceReference
gdcm::UIDs, 779
dicomSOPClass
gdcm::UIDs, 779
dicomSoftwareVersion
gdcm::UIDs, 779
dicomStationName
gdcm::UIDs, 779
dicomSupportedCharacterSet
gdcm::UIDs, 779
dicomTLSCyphersuite
gdcm::UIDs, 779
dicomThisNodeCertificateReference
gdcm::UIDs, 779
dicomTransferCapability
gdcm::UIDs, 780
dicomTransferRole
gdcm::UIDs, 779
dicomTransferSyntax
gdcm::UIDs, 779
dicomUniqueAETitle
gdcm::UIDs, 780
dicomUniqueAETitlesRegistryRoot
gdcm::UIDs, 779
dicomVendorData
gdcm::UIDs, 779
DigitalIntraoralXRayImageStorageForPresentation
gdcm::UIDs, 776
DigitalIntraoralXRayImageStorageForProcessing
gdcm::MediaStorage, 509
gdcm::UIDs, 776
DigitalIntraoralXrayImageStorageForPresentation
gdcm::MediaStorage, 509
DigitalMammographyImageStorageForPresentation
gdcm::MediaStorage, 509
DigitalMammographyImageStorageForProcessing
gdcm::MediaStorage, 509
DigitalMammographyXRayImageStorageForPresentation
gdcm::UIDs, 776
DigitalMammographyXRayImageStorageForProcessing
gdcm::UIDs, 776
DigitalXRayImageStorageForPresentation
gdcm::MediaStorage, 509
gdcm::UIDs, 776
DigitalXRayImageStorageForProcessing
gdcm::MediaStorage, 509
gdcm::UIDs, 776
dim
gdcm::terminal, 137
DuplicateAttributeError
gdcm::Parser, 556

eAABORTPDUReceivedOpen
gdcm::network, 135
eAABORTRequest
gdcm::network, 135

eAASSOCIATE_RQPDUreceived
gdcm::network, 135
eAASSOCIATERequestLocalUser
gdcm::network, 135
eAASSOCIATEresponseAccept
gdcm::network, 135
eAASSOCIATEresponseReject
gdcm::network, 135
eARELEASE_RPPDUReceived
gdcm::network, 135
eARELEASE_RQPDUReceivedOpen
gdcm::network, 135
eARELEASERequest
gdcm::network, 135
eARELEASEResponse
gdcm::network, 135
eARTIMTimerExpired
gdcm::network, 135
eASSOCIATE_ACPDUreceived
gdcm::network, 135
eASSOCIATE_RJPDUreceived
gdcm::network, 135
eArabic
gdcm, 125
eCyrillic
gdcm, 125
EDGE
gdcm::MeshPrimitive, 519
eEventDoesNotExist
gdcm::network, 135
eFind
gdcm, 125
eGB18030
gdcm, 125
eGreek
gdcm, 125
eHebrew
gdcm, 125
eImage
gdcm, 125
eJapanese
gdcm, 125
eJapaneseKanjiMultibyte
gdcm, 125
eJapaneseSupplementaryKanjiMultibyte
gdcm, 125
eKoreanHangulHanjaMultibyte
gdcm, 125
eLatin1
gdcm, 125
eLatin2
gdcm, 125
eLatin3
gdcm, 125
eLatin4
gdcm, 125
eLatin5
gdcm, 125
eMove
gdcm, 125
ePDATATFPDU
gdcm::network, 135
ePDATArequest
gdcm::network, 135
ePatient
gdcm, 125
ePatientRootType
gdcm, 126
eSeries
gdcm, 125
eSta10ReleaseCollisionAc
gdcm::network, 136
eSta11ReleaseCollisionRq
gdcm::network, 136
eSta12ReleaseCollisionAcLocal
gdcm::network, 136
eSta13AwaitingClose
gdcm::network, 136
eSta1Idle
gdcm::network, 135
eSta2Open
gdcm::network, 135
eSta3WaitLocalAssoc
gdcm::network, 135
eSta4LocalAssocDone
gdcm::network, 135
eSta5WaitRemoteAssoc
gdcm::network, 135
eSta6TransferReady
gdcm::network, 135
eSta7WaitRelease
gdcm::network, 135
eSta8WaitLocalRelease
gdcm::network, 136
eSta9ReleaseCollisionRqLocal
gdcm::network, 136
eStaDoesNotExist
gdcm::network, 135
eStudy
gdcm, 125
eStudyRootType
gdcm, 126
eThai
gdcm, 125
eTransportConnConfirmLocal
gdcm::network, 135
eTransportConnIndicLocal
gdcm::network, 135

eTransportConnectionClosed
gdcm::network, 135
eUTF8
gdcm, 125
eUnrecognizedPDUReceived
gdcm::network, 135
EncapsulatedCDAStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 778
EncapsulatedPDFStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 778
EnhancedCTImageStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
EnhancedMRImageStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
EnhancedSR
gdcm::MediaStorage, 510
EnhancedSRStorage
gdcm::UIDs, 777
EnhancedUSVolumeStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 780
EnhancedXAImageStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 777
EnhancedXRFImageStorage
gdcm::UIDs, 777
Explicit
gdcm::TransferSyntax, 760
ExplicitVRBigEndian
gdcm::TransferSyntax, 760
gdcm::UIDs, 773
ExplicitVRLittleEndian
gdcm::TransferSyntax, 760
gdcm::UIDs, 773

FACET
gdcm::MeshPrimitive, 519
FD
gdcm::VR, 855
FL
gdcm::VR, 855
FLOAT16
gdcm::PixelFormat, 575
FLOAT32
gdcm::PixelFormat, 575
FLOAT64
gdcm::PixelFormat, 575
FujiPrivateCRImageStorage
gdcm::MediaStorage, 511

GDCM_DIFFERENT
gdcm, 124
GDCM_EQUAL
gdcm, 124
GDCM_GREATER
gdcm, 125
GDCM_GREATEROREQUAL
gdcm, 125
GDCM_LESS
gdcm, 125
GDCM_LESSOREQUAL
gdcm, 125
GEMS
gdcm::Dicts, 322
GEPrivate3DModelStorage
gdcm::MediaStorage, 510
GRAY
gdcm::LookupTable, 498
GREEN
gdcm::LookupTable, 498
gdcm, 109
backslash, 126
eArabic, 125
eCyrillic, 125
eFind, 125
eGB18030, 125
eGreek, 125
eHebrew, 125
eImage, 125
eJapanese, 125
eJapaneseKanjiMultibyte, 125
eJapaneseSupplementaryKanjiMultibyte, 125
eKoreanHangulHanjaMultibyte, 125
eLatin1, 125
eLatin2, 125
eLatin3, 125
eLatin4, 125
eLatin5, 125
eMove, 125
ePatient, 125
ePatientRootType, 126
eSeries, 125
eStudy, 125
eStudyRootType, 126
eThai, 125
eUTF8, 125
GDCM_DIFFERENT, 124
GDCM_EQUAL, 124
GDCM_GREATER, 125
GDCM_GREATEROREQUAL, 125
GDCM_LESS, 125
GDCM_LESSOREQUAL, 125
LD_ALL, 126
LD_NOSEQ, 126
LD_NOSHADOW, 126

LD_NOSHADOWSEQ, 126
operator!=, 126
operator<<, 126–130
operator>>, 130
operator==, 130
gdcm::Attribute
VMType, 171
gdcm::Attribute< Group, Element, TVR, VM::VM1 >
VMType, 177
gdcm::CSAHeader
DATASET_FORMAT, 269
INTERFILE, 269
NOMAGIC, 269
SV10, 269
UNKNOWN, 269
ZEROED_OUT, 269
gdcm::CryptoFactory
CAPI, 260
DEFAULT, 260
OPENSSL, 260
OPENSSLP7, 260
gdcm::CryptographicMessageSyntax
AES128_CIPHER, 261
AES192_CIPHER, 262
AES256_CIPHER, 262
DES3_CIPHER, 261
gdcm::DictConverter
DICT_DEBUG, 316
DICT_DEFAULT, 316
DICT_XML, 316
gdcm::Dicts
GEMS, 322
PHILIPS, 322
SIEMENS, 322
gdcm::LookupTable
BLUE, 498
GRAY, 498
GREEN, 498
RED, 498
UNKNOWN, 498
gdcm::MediaStorage
AmbulatoryECGWaveformStorage, 510
Audio, 511
BasicTextSR, 510
BasicVoiceAudioWaveformStorage, 510
BreastTomosynthesisImageStorage, 511
CSANonImageStorage, 510
CTImageStorage, 509
CardiacElectrophysiologyWaveformStorage, 510
ComprehensiveSR, 510
ComputedRadiographyImageStorage, 509
DetachedPatientManagementSOPClass, 510
DetachedStudyManagementSOPClass, 510
DetachedVisitManagementSOPClass, 510
DigitalIntraoralXRayImageStorageForProcessing, 509
DigitalIntraoralXrayImageStorageForPresentation, 509
DigitalMammographyImageStorageForPresentation, 509
DigitalMammographyImageStorageForProcessing, 509
DigitalXRayImageStorageForPresentation, 509
DigitalXRayImageStorageForProcessing, 509
EncapsulatedCDAStorage, 510
EncapsulatedPDFStorage, 510
EnhancedCTImageStorage, 509
EnhancedMRImageStorage, 509
EnhancedSR, 510
EnhancedUSVolumeStorage, 511
EnhancedXAImageStorage, 511
FujiPrivateCRImageStorage, 511
GEPrivate3DModelStorage, 510
GeneralECGWaveformStorage, 510
GeneralElectricMagneticResonanceImageStorage, 510
GrayscaleSoftcopyPresentationStateStorageSOP↩ Class, 510
HangingProtocolStorage, 511
HardcopyGrayscaleImageStorage, 510
HemodynamicWaveformStorage, 510
KeyObjectSelectionDocument, 510
LeadECGWaveformStorage, 510
MRImageStorage, 509
MRSpectroscopyStorage, 509
MS_END, 511
MammographyCADSR, 510
MediaStorageDirectoryStorage, 509
ModalityPerformedProcedureStepSOPClass, 511
MultiframeGrayscaleByteSecondaryCaptureImage↩ Storage, 509
MultiframeGrayscaleWordSecondaryCapture↩ ImageStorage, 509
MultiframeSingleBitSecondaryCaptureImage↩ Storage, 509
MultiframeTrueColorSecondaryCaptureImage↩ Storage, 510
NoObject, 511
NuclearMedicineImageStorage, 510
NuclearMedicineImageStorageRetired, 509
ObjectEnd, 511
OphthalmicPhotography8BitImageStorage, 511
OphthalmicTomographyImageStorage, 511
PDF, 511
PETImageStorage, 510
Philips3D, 510
PhilipsPrivateMRSyntheticImageStorage, 511
RTDoseStorage, 510

RTImageStorage, 510
RTIonBeamsTreatmentRecordStorage, 511
RTIonPlanStorage, 511
RTPlanStorage, 510
RTStructureSetStorage, 510
RTTreatmentSummaryRecordStorage, 511
RawDataStorage, 510
SecondaryCaptureImageStorage, 509
Segmentation, 511
SegmentationStorage, 511
SpacialFiducialsStorage, 510
SpacialRegistrationStorage, 510
StandaloneCurveStorage, 510
StandaloneModalityLUTStorage, 510
StandaloneOverlayStorage, 510
StandaloneVOILUTStorage, 510
StudyComponentManagementSOPClass, 510
SurfaceSegmentationStorage, 511
ToshibaPrivateDataStorage, 510
URI, 511
UltrasoundImageStorage, 509
UltrasoundImageStorageRetired, 509
UltrasoundMultiFrameImageStorage, 509
UltrasoundMultiFrameImageStorageRetired, 509
VLEndoscopicImageStorage, 511
VLMicroscopicImageStorage, 511
VLPhotographicImageStorage, 511
VLWholeSlideMicroscopyImageStorage, 511
Video, 511
VideoEndoscopicImageStorage, 510
Waveform, 511
XRay3DAngiographicImageStorage, 511
XRayAngiographicBiPlaneImageStorageRetired, 510
XRayAngiographicImageStorage, 510
XRayRadiationDoseSR, 511
XRayRadiofluoroscopingImageStorage, 510
gdcm::MeshPrimitive
EDGE, 519
FACET, 519
LINE, 519
MPType_END, 519
TRIANGLE, 519
TRIANGLE_FAN, 519
TRIANGLE_STRIP, 519
VERTEX, 519
gdcm::Orientation
AXIAL, 546
CORONAL, 546
OBLIQUE, 546
SAGITTAL, 546
UNKNOWN, 546
gdcm::Overlay
Graphics, 550
Invalid, 550
ROI, 550
gdcm::Parser
DuplicateAttributeError, 556
JunkAfterDocElementError, 556
NoElementsError, 556
NoError, 556
NoMemoryError, 556
SyntaxError, 556
TagMismatchError, 556
UndefinedEntityError, 557
UnexpectedStateError, 557
gdcm::PhotometricInterpretation
ARGB, 572
CMYK, 572
HSV, 572
MONOCHROME1, 572
MONOCHROME2, 572
PALETTE_COLOR, 572
PI_END, 572
RGB, 572
UNKNOW, 572
YBR_FULL, 572
YBR_FULL_422, 572
YBR_ICT, 572
YBR_PARTIAL_420, 572
YBR_PARTIAL_422, 572
YBR_RCT, 572
gdcm::PixelFormat
FLOAT16, 575
FLOAT32, 575
FLOAT64, 575
INT12, 575
INT16, 575
INT32, 575
INT64, 575
INT8, 575
SINGLEBIT, 575
UINT12, 575
UINT16, 575
UINT32, 575
UINT64, 575
UINT8, 575
UNKNOWN, 575
gdcm::Printer
CONDENSED_STYLE, 604
VERBOSE_STYLE, 604
XML, 604
gdcm::STATIC_ASSERTION_FAILURE< true >
value, 700
gdcm::Segment
ALGOType_END, 651
AUTOMATIC, 651
MANUAL, 651
gdcm::Spacing

CALIBRATED, 696
DETECTOR, 696
MAGNIFIED, 696
UNKNOWN, 696
gdcm::Surface
NO, 720
POINTS, 720
STATES_END, 720
SURFACE, 720
UNKNOWN, 720
VIEWType_END, 720
WIREFRAME, 720
YES, 720
gdcm::SwapCode
BadBigEndian, 731
BadLittleEndian, 731
BigEndian, 731
LittleEndian, 731
Unknown, 731
gdcm::TransferSyntax
CT_private_ELE, 761
DeflatedExplicitVRLittleEndian, 760
Explicit, 760
ExplicitVRBigEndian, 760
ExplicitVRLittleEndian, 760
Implicit, 760
ImplicitVRBigEndianACRNEMA, 761
ImplicitVRBigEndianPrivateGE, 760
ImplicitVRLittleEndian, 760
JPEG2000, 761
JPEG2000Lossless, 761
JPEG2000Part2, 761
JPEG2000Part2Lossless, 761
JPEGBaselineProcess1, 760
JPEGExtendedProcess2_4, 760
JPEGExtendedProcess3_5, 760
JPEGFullProgressionProcess10_12, 760
JPEGLSLossless, 760
JPEGLSNearLossless, 760
JPEGLosslessProcess14, 760
JPEGLosslessProcess14_1, 760
JPEGSpectralSelectionProcess6_8, 760
JPIPReferenced, 761
MPEG2MainProfile, 761
RLELossless, 761
TS_END, 761
Unknown, 760
gdcm::Type
T1, 766
T1C, 766
T2, 766
T2C, 766
T3, 766
UNKNOWN, 766
gdcm::UIDs
AmbulatoryECGWaveformStorage, 776
AudioSRStorageTrialRetired, 777
BasicAnnotationBoxSOPClass, 775
BasicColorImageBoxSOPClass, 775
BasicColorPrintManagementMetaSOPClass, 775
BasicFilmBoxSOPClass, 775
BasicFilmSessionSOPClass, 775
BasicGrayscaleImageBoxSOPClass, 775
BasicGrayscalePrintManagementMetaSOPClass, 775
BasicPrintImageOverlayBoxSOPClassRetired, 776
BasicStudyContentNotificationSOPClassRetired, 775
BasicTextSRStorage, 777
BasicVoiceAudioWaveformStorage, 776
BlendingSoftcopyPresentationStateStorageSOP↩ Class, 777
BreastImagingRelevantPatientInformationQuery, 778
BreastTomosynthesisImageStorage, 780
CTImageStorage, 776
CardiacElectrophysiologyWaveformStorage, 776
CardiacRelevantPatientInformationQuery, 779
ChestCADSRStorage, 778
ColorSoftcopyPresentationStateStorageSOPClass, 777
ComprehensiveSRStorage, 777
ComprehensiveSRStorageTrialRetired, 777
ComputedRadiographyImageStorage, 776
DICOMApplicationContextName, 775
DICOMControlledTerminology, 775
DICOMUIDRegistry, 775
DeflatedExplicitVRLittleEndian, 773
DeformableSpatialRegistrationStorage, 777
DetachedInterpretationManagementSOPClass↩ Retired, 775
DetachedPatientManagementMetaSOPClass↩ Retired, 775
DetachedPatientManagementSOPClassRetired, 775
DetachedResultsManagementMetaSOPClass↩ Retired, 775
DetachedResultsManagementSOPClassRetired, 775
DetachedStudyManagementMetaSOPClassRetired, 775
DetachedStudyManagementSOPClassRetired, 775
DetachedVisitManagementSOPClassRetired, 775
DetailSRStorageTrialRetired, 777
dicomAETitle, 779
dicomApplicationCluster, 779
dicomAssociationAcceptor, 779
dicomAssociationInitiator, 779
dicomAuthorizedNodeCertificateReference, 779
dicomConfigurationRoot, 779

dicomDescription, 779
dicomDevice, 779
dicomDeviceName, 779
dicomDeviceSerialNumber, 779
dicomDevicesRoot, 779
dicomHostname, 779
dicomInstalled, 779
dicomInstitutionAddress, 779
dicomInstitutionDepartmentName, 779
dicomInstitutionName, 779
dicomIssuerOfPatientID, 779
dicomManufacturer, 779
dicomManufacturerModelName, 779
dicomNetworkAE, 779
dicomNetworkConnection, 780
dicomNetworkConnectionReference, 779
dicomPort, 779
dicomPreferredCalledAETitle, 779
dicomPreferredCallingAETitle, 779
dicomPrimaryDeviceType, 779
dicomRelatedDeviceReference, 779
dicomSOPClass, 779
dicomSoftwareVersion, 779
dicomStationName, 779
dicomSupportedCharacterSet, 779
dicomTLSCyphersuite, 779
dicomThisNodeCertificateReference, 779
dicomTransferCapability, 780
dicomTransferRole, 779
dicomTransferSyntax, 779
dicomUniqueAETitle, 780
dicomUniqueAETitlesRegistryRoot, 779
dicomVendorData, 779
DigitalIntraoralXRayImageStorageForPresentation, 776
DigitalIntraoralXRayImageStorageForProcessing, 776
DigitalMammographyXRayImageStorageFor↩ Presentation, 776
DigitalMammographyXRayImageStorageFor↩ Processing, 776
DigitalXRayImageStorageForPresentation, 776
DigitalXRayImageStorageForProcessing, 776
EncapsulatedCDAStorage, 778
EncapsulatedPDFStorage, 778
EnhancedCTImageStorage, 776
EnhancedMRImageStorage, 776
EnhancedSRStorage, 777
EnhancedUSVolumeStorage, 780
EnhancedXAImageStorage, 777
EnhancedXRFImageStorage, 777
ExplicitVRBigEndian, 773
ExplicitVRLittleEndian, 773
GeneralECGWaveformStorage, 776
GeneralPurposePerformedProcedureStepSOP↩ Class, 778
GeneralPurposeScheduledProcedureStepSOP↩ Class, 778
GeneralPurposeWorklistInformationModelFIND, 778
GeneralPurposeWorklistManagementMetaSOP↩ Class, 778
GeneralRelevantPatientInformationQuery, 778
GrayscaleSoftcopyPresentationStateStorageSOP↩ Class, 777
HangingProtocolInformationModelFIND, 779
HangingProtocolInformationModelMOVE, 779
HangingProtocolStorage, 779
HardcopyColorImageStorageSOPClassRetired, 776
HardcopyGrayscaleImageStorageSOPClassRetired, 776
HemodynamicWaveformStorage, 776
ICBM452T1FrameofReference, 775
ICBMSingleSubjectMRIFrameofReference, 775
ImageOverlayBoxSOPClassRetired, 776
ImplicitVRLittleEndianDefaultTransferSyntaxforDIC↩ OM, 773
InstanceAvailabilityNotificationSOPClass, 778
JPEG2000ImageCompression, 774
JPEG2000ImageCompressionLosslessOnly, 774
JPEG2000Part2MulticomponentImageCompression, 774
JPEG2000Part2MulticomponentImageCompression↩ LosslessOnly, 774
JPEGBaselineProcess1DefaultTransferSyntaxfor↩ LossyJPEG8BitImageCompression, 773
JPEGExtendedHierarchicalProcess1618Retired, 774
JPEGExtendedHierarchicalProcess1719Retired, 774
JPEGExtendedProcess24DefaultTransferSyntaxfor↩ LossyJPEG12BitImageCompressionProcess4only, 773
JPEGExtendedProcess35Retired, 773
JPEGFullProgressionHierarchicalProcess2426↩ Retired, 774
JPEGFullProgressionHierarchicalProcess2527↩ Retired, 774
JPEGFullProgressionNonHierarchicalProcess1012↩ Retired, 773
JPEGFullProgressionNonHierarchicalProcess1113↩ Retired, 773
JPEGLSLosslessImageCompression, 774
JPEGLSLossyNearLosslessImageCompression, 774
JPEGLosslessHierarchicalProcess28Retired, 774
JPEGLosslessHierarchicalProcess29Retired, 774
JPEGLosslessNonHierarchicalFirstOrderPrediction↩ Process14SelectionValue1DefaultTransfer↩ SyntaxforLosslessJPEGImageCompression, 774
JPEGLosslessNonHierarchicalProcess14, 773

JPEGLosslessNonHierarchicalProcess15Retired, 774
JPEGSpectralSelectionHierarchicalProcess2022↩ Retired, 774
JPEGSpectralSelectionHierarchicalProcess2123↩ Retired, 774
JPEGSpectralSelectionNonHierarchicalProcess68↩ Retired, 773
JPEGSpectralSelectionNonHierarchicalProcess79↩ Retired, 773
JPIPReferenced, 774
JPIPReferencedDeflate, 774
KeyObjectSelectionDocumentStorage, 778
MPEG2MainProfileMainLevel, 774
MRImageStorage, 776
MRSpectroscopyStorage, 776
MammographyCADSRStorage, 777
MediaCreationManagementSOPClassUID, 776
MediaStorageDirectoryStorage, 774
ModalityPerformedProcedureStepNotificationSOP↩ Class, 775
ModalityPerformedProcedureStepRetrieveSOP↩ Class, 775
ModalityPerformedProcedureStepSOPClass, 775
ModalityWorklistInformationModelFIND, 778
MultiframeGrayscaleByteSecondaryCaptureImage↩ Storage, 776
MultiframeGrayscaleWordSecondaryCapture↩ ImageStorage, 776
MultiframeSingleBitSecondaryCaptureImage↩ Storage, 776
MultiframeTrueColorSecondaryCaptureImage↩ Storage, 776
NuclearMedicineImageStorage, 777
NuclearMedicineImageStorageRetired, 776
OphthalmicPhotography16BitImageStorage, 777
OphthalmicPhotography8BitImageStorage, 777
OphthalmicTomographyImageStorage, 777
PatientRootQueryRetrieveInformationModelFIND, 778
PatientRootQueryRetrieveInformationModelGET, 778
PatientRootQueryRetrieveInformationModelMOVE, 778
PatientStudyOnlyQueryRetrieveInformationModelF↩ INDRetired, 778
PatientStudyOnlyQueryRetrieveInformationModel↩ GETRetired, 778
PatientStudyOnlyQueryRetrieveInformationModel↩ MOVERetired, 778
PositronEmissionTomographyImageStorage, 778
PresentationLUTSOPClass, 776
PrintJobSOPClass, 775
PrintQueueManagementSOPClassRetired, 776
PrintQueueSOPInstanceRetired, 776
PrinterConfigurationRetrievalSOPClass, 775
PrinterConfigurationRetrievalSOPInstance, 775
PrinterSOPClass, 775
PrinterSOPInstance, 775
ProceduralEventLoggingSOPClass, 775
ProceduralEventLoggingSOPInstance, 775
ProcedureLogStorage, 777
ProductCharacteristicsQuerySOPClass, 779
PseudoColorSoftcopyPresentationStateStorageSO↩ PClass, 777
PullPrintRequestSOPClassRetired, 776
PullStoredPrintManagementMetaSOPClassRetired, 776
RFC2557MIMEencapsulation, 774
RLELossless, 774
RTBeamsDeliveryInstructionStorageSupplement74↩ FrozenDraft, 778
RTBeamsTreatmentRecordStorage, 778
RTBrachyTreatmentRecordStorage, 778
RTConventionalMachineVerificationSupplement74↩ FrozenDraft, 778
RTDoseStorage, 778
RTImageStorage, 778
RTIonBeamsTreatmentRecordStorage, 778
RTIonMachineVerificationSupplement74FrozenDraft, 778
RTIonPlanStorage, 778
RTPlanStorage, 778
RTStructureSetStorage, 778
RTTreatmentSummaryRecordStorage, 778
RawDataStorage, 777
RealWorldValueMappingStorage, 777
ReferencedColorPrintManagementMetaSOPClass↩ Retired, 775
ReferencedGrayscalePrintManagementMetaSOP↩ ClassRetired, 775
ReferencedImageBoxSOPClassRetired, 775
SPM2AVG152PDFrameofReference, 774
SPM2AVG152T1FrameofReference, 774
SPM2AVG152T2FrameofReference, 774
SPM2AVG305T1FrameofReference, 774
SPM2BRAINMASKFrameofReference, 774
SPM2CSFFrameofReference, 774
SPM2EPIFrameofReference, 774
SPM2FILT1FrameofReference, 774
SPM2GRAYFrameofReference, 774
SPM2PDFrameofReference, 774
SPM2PETFrameofReference, 774
SPM2SINGLESUBJT1FrameofReference, 774
SPM2SPECTFrameofReference, 774
SPM2T1FrameofReference, 774
SPM2T2FrameofReference, 774
SPM2TRANSMFrameofReference, 774

SPM2WHITEFrameofReference, 774
SecondaryCaptureImageStorage, 776
SegmentationStorage, 777
SpatialFiducialsStorage, 777
SpatialRegistrationStorage, 777
StandaloneCurveStorageRetired, 776
StandaloneModalityLUTStorageRetired, 777
StandaloneOverlayStorageRetired, 776
StandalonePETCurveStorageRetired, 778
StandaloneVOILUTStorageRetired, 777
StereometricRelationshipStorage, 777
StorageCommitmentPullModelSOPClassRetired, 775
StorageCommitmentPullModelSOPInstanceRetired, 775
StorageCommitmentPushModelSOPClass, 775
StorageCommitmentPushModelSOPInstance, 775
StorageServiceClass, 775
StoredPrintStorageSOPClassRetired, 776
StudyComponentManagementSOPClassRetired, 775
StudyRootQueryRetrieveInformationModelFIND, 778
StudyRootQueryRetrieveInformationModelGET, 778
StudyRootQueryRetrieveInformationModelMOVE, 778
SubstanceAdministrationLoggingSOPClass, 775
SubstanceAdministrationLoggingSOPInstance, 775
SubstanceApprovalQuerySOPClass, 779
SurfaceSegmentationStorage, 780
TalairachBrainAtlasFrameofReference, 774
TextSRStorageTrialRetired, 777
uid_1_2_840_10008_15_0_3_1, 785
uid_1_2_840_10008_15_0_3_10, 785
uid_1_2_840_10008_15_0_3_11, 785
uid_1_2_840_10008_15_0_3_12, 786
uid_1_2_840_10008_15_0_3_13, 786
uid_1_2_840_10008_15_0_3_14, 786
uid_1_2_840_10008_15_0_3_15, 786
uid_1_2_840_10008_15_0_3_16, 786
uid_1_2_840_10008_15_0_3_17, 786
uid_1_2_840_10008_15_0_3_18, 786
uid_1_2_840_10008_15_0_3_19, 786
uid_1_2_840_10008_15_0_3_2, 785
uid_1_2_840_10008_15_0_3_20, 786
uid_1_2_840_10008_15_0_3_21, 786
uid_1_2_840_10008_15_0_3_22, 786
uid_1_2_840_10008_15_0_3_23, 786
uid_1_2_840_10008_15_0_3_24, 786
uid_1_2_840_10008_15_0_3_25, 786
uid_1_2_840_10008_15_0_3_26, 786
uid_1_2_840_10008_15_0_3_27, 786
uid_1_2_840_10008_15_0_3_28, 786
uid_1_2_840_10008_15_0_3_29, 786
uid_1_2_840_10008_15_0_3_3, 785
uid_1_2_840_10008_15_0_3_30, 786
uid_1_2_840_10008_15_0_3_31, 786
uid_1_2_840_10008_15_0_3_4, 785
uid_1_2_840_10008_15_0_3_5, 785
uid_1_2_840_10008_15_0_3_6, 785
uid_1_2_840_10008_15_0_3_7, 785
uid_1_2_840_10008_15_0_3_8, 785
uid_1_2_840_10008_15_0_3_9, 785
uid_1_2_840_10008_15_0_4_1, 786
uid_1_2_840_10008_15_0_4_2, 786
uid_1_2_840_10008_15_0_4_3, 786
uid_1_2_840_10008_15_0_4_4, 786
uid_1_2_840_10008_15_0_4_5, 786
uid_1_2_840_10008_15_0_4_6, 786
uid_1_2_840_10008_15_0_4_7, 786
uid_1_2_840_10008_15_0_4_8, 786
uid_1_2_840_10008_1_1, 780
uid_1_2_840_10008_1_2, 780
uid_1_2_840_10008_1_20_1, 781
uid_1_2_840_10008_1_20_1_1, 781
uid_1_2_840_10008_1_20_2, 781
uid_1_2_840_10008_1_20_2_1, 781
uid_1_2_840_10008_1_2_1, 780
uid_1_2_840_10008_1_2_1_99, 780
uid_1_2_840_10008_1_2_2, 780
uid_1_2_840_10008_1_2_4_100, 781
uid_1_2_840_10008_1_2_4_50, 780
uid_1_2_840_10008_1_2_4_51, 780
uid_1_2_840_10008_1_2_4_52, 780
uid_1_2_840_10008_1_2_4_53, 780
uid_1_2_840_10008_1_2_4_54, 780
uid_1_2_840_10008_1_2_4_55, 780
uid_1_2_840_10008_1_2_4_56, 780
uid_1_2_840_10008_1_2_4_57, 780
uid_1_2_840_10008_1_2_4_58, 780
uid_1_2_840_10008_1_2_4_59, 780
uid_1_2_840_10008_1_2_4_60, 780
uid_1_2_840_10008_1_2_4_61, 780
uid_1_2_840_10008_1_2_4_62, 780
uid_1_2_840_10008_1_2_4_63, 780
uid_1_2_840_10008_1_2_4_64, 780
uid_1_2_840_10008_1_2_4_65, 780
uid_1_2_840_10008_1_2_4_66, 780
uid_1_2_840_10008_1_2_4_70, 780
uid_1_2_840_10008_1_2_4_80, 780
uid_1_2_840_10008_1_2_4_81, 780
uid_1_2_840_10008_1_2_4_90, 780
uid_1_2_840_10008_1_2_4_91, 780
uid_1_2_840_10008_1_2_4_92, 780
uid_1_2_840_10008_1_2_4_93, 780
uid_1_2_840_10008_1_2_4_94, 780
uid_1_2_840_10008_1_2_4_95, 781
uid_1_2_840_10008_1_2_5, 781
uid_1_2_840_10008_1_2_6_1, 781

uid_1_2_840_10008_1_2_6_2, 781
uid_1_2_840_10008_1_3_10, 781
uid_1_2_840_10008_1_40, 781
uid_1_2_840_10008_1_40_1, 781
uid_1_2_840_10008_1_42, 781
uid_1_2_840_10008_1_42_1, 781
uid_1_2_840_10008_1_4_1_1, 781
uid_1_2_840_10008_1_4_1_10, 781
uid_1_2_840_10008_1_4_1_11, 781
uid_1_2_840_10008_1_4_1_12, 781
uid_1_2_840_10008_1_4_1_13, 781
uid_1_2_840_10008_1_4_1_14, 781
uid_1_2_840_10008_1_4_1_15, 781
uid_1_2_840_10008_1_4_1_16, 781
uid_1_2_840_10008_1_4_1_17, 781
uid_1_2_840_10008_1_4_1_18, 781
uid_1_2_840_10008_1_4_1_2, 781
uid_1_2_840_10008_1_4_1_3, 781
uid_1_2_840_10008_1_4_1_4, 781
uid_1_2_840_10008_1_4_1_5, 781
uid_1_2_840_10008_1_4_1_6, 781
uid_1_2_840_10008_1_4_1_7, 781
uid_1_2_840_10008_1_4_1_8, 781
uid_1_2_840_10008_1_4_1_9, 781
uid_1_2_840_10008_1_4_2_1, 781
uid_1_2_840_10008_1_4_2_2, 781
uid_1_2_840_10008_1_9, 781
uid_1_2_840_10008_2_16_4, 781
uid_1_2_840_10008_2_6_1, 781
uid_1_2_840_10008_3_1_1_1, 781
uid_1_2_840_10008_3_1_2_1_1, 781
uid_1_2_840_10008_3_1_2_1_4, 781
uid_1_2_840_10008_3_1_2_2_1, 781
uid_1_2_840_10008_3_1_2_3_1, 781
uid_1_2_840_10008_3_1_2_3_2, 782
uid_1_2_840_10008_3_1_2_3_3, 782
uid_1_2_840_10008_3_1_2_3_4, 782
uid_1_2_840_10008_3_1_2_3_5, 782
uid_1_2_840_10008_3_1_2_5_1, 782
uid_1_2_840_10008_3_1_2_5_4, 782
uid_1_2_840_10008_3_1_2_5_5, 782
uid_1_2_840_10008_3_1_2_6_1, 782
uid_1_2_840_10008_4_2, 782
uid_1_2_840_10008_5_1_1_1, 782
uid_1_2_840_10008_5_1_1_14, 782
uid_1_2_840_10008_5_1_1_15, 782
uid_1_2_840_10008_5_1_1_16, 782
uid_1_2_840_10008_5_1_1_16_376, 782
uid_1_2_840_10008_5_1_1_17, 782
uid_1_2_840_10008_5_1_1_17_376, 782
uid_1_2_840_10008_5_1_1_18, 782
uid_1_2_840_10008_5_1_1_18_1, 782
uid_1_2_840_10008_5_1_1_2, 782
uid_1_2_840_10008_5_1_1_22, 782
uid_1_2_840_10008_5_1_1_23, 782
uid_1_2_840_10008_5_1_1_24, 782
uid_1_2_840_10008_5_1_1_24_1, 782
uid_1_2_840_10008_5_1_1_25, 782
uid_1_2_840_10008_5_1_1_26, 782
uid_1_2_840_10008_5_1_1_27, 782
uid_1_2_840_10008_5_1_1_29, 782
uid_1_2_840_10008_5_1_1_30, 782
uid_1_2_840_10008_5_1_1_31, 782
uid_1_2_840_10008_5_1_1_32, 782
uid_1_2_840_10008_5_1_1_33, 782
uid_1_2_840_10008_5_1_1_4, 782
uid_1_2_840_10008_5_1_1_4_1, 782
uid_1_2_840_10008_5_1_1_4_2, 782
uid_1_2_840_10008_5_1_1_9, 782
uid_1_2_840_10008_5_1_1_9_1, 782
uid_1_2_840_10008_5_1_4_1_1_1, 782
uid_1_2_840_10008_5_1_4_1_1_10, 783
uid_1_2_840_10008_5_1_4_1_1_104_1, 784
uid_1_2_840_10008_5_1_4_1_1_104_2, 784
uid_1_2_840_10008_5_1_4_1_1_11, 783
uid_1_2_840_10008_5_1_4_1_1_11_1, 783
uid_1_2_840_10008_5_1_4_1_1_11_2, 783
uid_1_2_840_10008_5_1_4_1_1_11_3, 783
uid_1_2_840_10008_5_1_4_1_1_11_4, 783
uid_1_2_840_10008_5_1_4_1_1_128, 784
uid_1_2_840_10008_5_1_4_1_1_129, 784
uid_1_2_840_10008_5_1_4_1_1_12_1, 783
uid_1_2_840_10008_5_1_4_1_1_12_1_1, 783
uid_1_2_840_10008_5_1_4_1_1_12_2, 783
uid_1_2_840_10008_5_1_4_1_1_12_2_1, 783
uid_1_2_840_10008_5_1_4_1_1_12_3, 783
uid_1_2_840_10008_5_1_4_1_1_13_1_1, 783
uid_1_2_840_10008_5_1_4_1_1_13_1_2, 783
uid_1_2_840_10008_5_1_4_1_1_13_1_3, 786
uid_1_2_840_10008_5_1_4_1_1_1_1, 782
uid_1_2_840_10008_5_1_4_1_1_1_1_1, 782
uid_1_2_840_10008_5_1_4_1_1_1_2, 782
uid_1_2_840_10008_5_1_4_1_1_1_2_1, 782
uid_1_2_840_10008_5_1_4_1_1_1_3, 782
uid_1_2_840_10008_5_1_4_1_1_1_3_1, 783
uid_1_2_840_10008_5_1_4_1_1_2, 783
uid_1_2_840_10008_5_1_4_1_1_20, 783
uid_1_2_840_10008_5_1_4_1_1_2_1, 783
uid_1_2_840_10008_5_1_4_1_1_3, 783
uid_1_2_840_10008_5_1_4_1_1_3_1, 783
uid_1_2_840_10008_5_1_4_1_1_4, 783
uid_1_2_840_10008_5_1_4_1_1_481_1, 784
uid_1_2_840_10008_5_1_4_1_1_481_2, 784
uid_1_2_840_10008_5_1_4_1_1_481_3, 784
uid_1_2_840_10008_5_1_4_1_1_481_4, 784
uid_1_2_840_10008_5_1_4_1_1_481_5, 784
uid_1_2_840_10008_5_1_4_1_1_481_6, 784
uid_1_2_840_10008_5_1_4_1_1_481_7, 784

uid_1_2_840_10008_5_1_4_1_1_481_8, 784
uid_1_2_840_10008_5_1_4_1_1_481_9, 784
uid_1_2_840_10008_5_1_4_1_1_4_1, 783
uid_1_2_840_10008_5_1_4_1_1_4_2, 783
uid_1_2_840_10008_5_1_4_1_1_5, 783
uid_1_2_840_10008_5_1_4_1_1_6, 783
uid_1_2_840_10008_5_1_4_1_1_66, 783
uid_1_2_840_10008_5_1_4_1_1_66_1, 783
uid_1_2_840_10008_5_1_4_1_1_66_2, 783
uid_1_2_840_10008_5_1_4_1_1_66_3, 784
uid_1_2_840_10008_5_1_4_1_1_66_4, 784
uid_1_2_840_10008_5_1_4_1_1_66_5, 786
uid_1_2_840_10008_5_1_4_1_1_67, 784
uid_1_2_840_10008_5_1_4_1_1_6_1, 783
uid_1_2_840_10008_5_1_4_1_1_6_2, 786
uid_1_2_840_10008_5_1_4_1_1_7, 783
uid_1_2_840_10008_5_1_4_1_1_77_1, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_1, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_2, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_3, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_4, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_6, 786
uid_1_2_840_10008_5_1_4_1_1_77_2, 784
uid_1_2_840_10008_5_1_4_1_1_7_1, 783
uid_1_2_840_10008_5_1_4_1_1_7_2, 783
uid_1_2_840_10008_5_1_4_1_1_7_3, 783
uid_1_2_840_10008_5_1_4_1_1_7_4, 783
uid_1_2_840_10008_5_1_4_1_1_8, 783
uid_1_2_840_10008_5_1_4_1_1_88_1, 784
uid_1_2_840_10008_5_1_4_1_1_88_11, 784
uid_1_2_840_10008_5_1_4_1_1_88_2, 784
uid_1_2_840_10008_5_1_4_1_1_88_22, 784
uid_1_2_840_10008_5_1_4_1_1_88_3, 784
uid_1_2_840_10008_5_1_4_1_1_88_33, 784
uid_1_2_840_10008_5_1_4_1_1_88_4, 784
uid_1_2_840_10008_5_1_4_1_1_88_40, 784
uid_1_2_840_10008_5_1_4_1_1_88_50, 784
uid_1_2_840_10008_5_1_4_1_1_88_59, 784
uid_1_2_840_10008_5_1_4_1_1_88_65, 784
uid_1_2_840_10008_5_1_4_1_1_88_67, 784
uid_1_2_840_10008_5_1_4_1_1_9, 783
uid_1_2_840_10008_5_1_4_1_1_9_1, 783
uid_1_2_840_10008_5_1_4_1_1_9_1_1, 783
uid_1_2_840_10008_5_1_4_1_1_9_1_2, 783
uid_1_2_840_10008_5_1_4_1_1_9_1_3, 783
uid_1_2_840_10008_5_1_4_1_1_9_2_1, 783
uid_1_2_840_10008_5_1_4_1_1_9_3_1, 783
uid_1_2_840_10008_5_1_4_1_1_9_4_1, 783
uid_1_2_840_10008_5_1_4_1_2_1_1, 784
uid_1_2_840_10008_5_1_4_1_2_1_2, 785
uid_1_2_840_10008_5_1_4_1_2_1_3, 785
uid_1_2_840_10008_5_1_4_1_2_2_1, 785
uid_1_2_840_10008_5_1_4_1_2_2_2, 785
uid_1_2_840_10008_5_1_4_1_2_2_3, 785
uid_1_2_840_10008_5_1_4_1_2_3_1, 785
uid_1_2_840_10008_5_1_4_1_2_3_2, 785
uid_1_2_840_10008_5_1_4_1_2_3_3, 785
uid_1_2_840_10008_5_1_4_31, 785
uid_1_2_840_10008_5_1_4_32, 785
uid_1_2_840_10008_5_1_4_32_1, 785
uid_1_2_840_10008_5_1_4_32_2, 785
uid_1_2_840_10008_5_1_4_32_3, 785
uid_1_2_840_10008_5_1_4_33, 785
uid_1_2_840_10008_5_1_4_34_1, 785
uid_1_2_840_10008_5_1_4_34_2, 785
uid_1_2_840_10008_5_1_4_34_3, 785
uid_1_2_840_10008_5_1_4_34_4, 785
uid_1_2_840_10008_5_1_4_34_4_1, 785
uid_1_2_840_10008_5_1_4_34_4_2, 785
uid_1_2_840_10008_5_1_4_34_4_3, 785
uid_1_2_840_10008_5_1_4_34_4_4, 785
uid_1_2_840_10008_5_1_4_34_5, 785
uid_1_2_840_10008_5_1_4_37_1, 785
uid_1_2_840_10008_5_1_4_37_2, 785
uid_1_2_840_10008_5_1_4_37_3, 785
uid_1_2_840_10008_5_1_4_38_1, 785
uid_1_2_840_10008_5_1_4_38_2, 785
uid_1_2_840_10008_5_1_4_38_3, 785
uid_1_2_840_10008_5_1_4_41, 785
uid_1_2_840_10008_5_1_4_42, 785
UltrasoundImageStorage, 776
UltrasoundImageStorageRetired, 776
UltrasoundMultiframeImageStorage, 776
UltrasoundMultiframeImageStorageRetired, 776
UnifiedProcedureStepEventSOPClass, 778
UnifiedProcedureStepPullSOPClass, 778
UnifiedProcedureStepPushSOPClass, 778
UnifiedProcedureStepWatchSOPClass, 778
UnifiedWorklistandProcedureStepSOPInstance, 778
UnifiedWorklistandProcedureStepServiceClass, 778
VLEndoscopicImageStorage, 777
VLImageStorageTrialRetired, 777
VLMicroscopicImageStorage, 777
VLMultiframeImageStorageTrialRetired, 777
VLPhotographicImageStorage, 777
VLSlideCoordinatesMicroscopicImageStorage, 777
VLWholeSlideMicroscopyImageStorage, 780
VOILUTBoxSOPClass, 776
VerificationSOPClass, 773
VideoEndoscopicImageStorage, 777
VideoMicroscopicImageStorage, 777

VideoPhotographicImageStorage, 777
WaveformStorageTrialRetired, 776
XMLEncoding, 774
XRay3DAngiographicImageStorage, 777
XRay3DCraniofacialImageStorage, 777
XRayAngiographicBiPlaneImageStorageRetired, 777
XRayAngiographicImageStorage, 777
XRayRadiationDoseSRStorage, 778
XRayRadiofluoroscopicImageStorage, 777
gdcm::Usage
Conditional, 837
Invalid, 837
Mandatory, 837
UserOption, 837
gdcm::VM
VM0, 850
VM1, 850
VM10, 850
VM12, 850
VM16, 850
VM18, 850
VM1_2, 851
VM1_3, 851
VM1_32, 851
VM1_4, 851
VM1_5, 851
VM1_8, 851
VM1_99, 851
VM1_n, 851
VM2, 850
VM24, 850
VM256, 851
VM28, 850
VM2_2n, 851
VM2_n, 851
VM3, 850
VM30_30n, 851
VM32, 850
VM35, 850
VM3_3n, 851
VM3_4, 851
VM3_n, 851
VM4, 850
VM47_47n, 851
VM4_4n, 851
VM5, 850
VM6, 850
VM6_6n, 851
VM7_7n, 851
VM8, 850
VM9, 850
VM99, 851
VM_END, 851
gdcm::VR
AE, 854
AS, 854
AT, 854
CS, 854
DA, 854
DS, 855
DT, 855
FD, 855
FL, 855
INVALID, 854
IS, 855
LO, 855
LT, 855
OB, 855
OB_OW, 855
OD, 855
OF, 855
OW, 855
PN, 855
SH, 855
SL, 855
SQ, 855
SS, 855
ST, 855
TM, 855
UI, 855
UL, 855
UN, 855
US, 855
US_SS, 855
US_SS_OW, 855
UT, 855
VL16, 855
VL32, 855
VR_END, 855
VR_VM1, 855
VRALL, 855
VRASCII, 855
VRBINARY, 855
gdcm::XMLPrinter
LOADBULKDATA, 927
OnlyUUID, 927
gdcm::network
eAABORTPDUReceivedOpen, 135
eAABORTRequest, 135
eAASSOCIATE_RQPDUreceived, 135
eAASSOCIATERequestLocalUser, 135
eAASSOCIATEresponseAccept, 135
eAASSOCIATEresponseReject, 135
eARELEASE_RPPDUReceived, 135
eARELEASE_RQPDUReceivedOpen, 135
eARELEASERequest, 135
eARELEASEResponse, 135
eARTIMTimerExpired, 135

eASSOCIATE_ACPDUreceived, 135
eASSOCIATE_RJPDUreceived, 135
eEventDoesNotExist, 135
ePDATATFPDU, 135
ePDATArequest, 135
eSta10ReleaseCollisionAc, 136
eSta11ReleaseCollisionRq, 136
eSta12ReleaseCollisionAcLocal, 136
eSta13AwaitingClose, 136
eSta1Idle, 135
eSta2Open, 135
eSta3WaitLocalAssoc, 135
eSta4LocalAssocDone, 135
eSta5WaitRemoteAssoc, 135
eSta6TransferReady, 135
eSta7WaitRelease, 135
eSta8WaitLocalRelease, 136
eSta9ReleaseCollisionRqLocal, 136
eStaDoesNotExist, 135
eTransportConnConfirmLocal, 135
eTransportConnIndicLocal, 135
eTransportConnectionClosed, 135
eUnrecognizedPDUReceived, 135
gdcm::network::DIMSE
C_CANCEL_RQ, 325
C_ECHO_RQ, 325
C_ECHO_RSP, 325
C_FIND_RQ, 324
C_FIND_RSP, 324
C_GET_RQ, 324
C_GET_RSP, 324
C_MOVE_RQ, 324
C_MOVE_RSP, 325
C_STORE_RQ, 324
C_STORE_RSP, 324
N_ACTION_RQ, 325
N_ACTION_RSP, 325
N_CREATE_RQ, 325
N_CREATE_RSP, 325
N_DELETE_RQ, 325
N_DELETE_RSP, 325
N_EVENT_REPORT_RQ, 325
N_EVENT_REPORT_RSP, 325
N_GET_RQ, 325
N_GET_RSP, 325
N_SET_RQ, 325
N_SET_RSP, 325
gdcm::terminal
black, 137
blink, 137
blue, 138
bright, 137
CONSOLE, 138
cyan, 138
dim, 137
green, 137
hidden, 137
magenta, 138
red, 137
reset, 137
reverse, 137
underline, 137
VT100, 138
white, 138
yellow, 137
GeneralECGWaveformStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 776
GeneralElectricMagneticResonanceImageStorage
gdcm::MediaStorage, 510
GeneralPurposePerformedProcedureStepSOPClass
gdcm::UIDs, 778
GeneralPurposeScheduledProcedureStepSOPClass
gdcm::UIDs, 778
GeneralPurposeWorklistInformationModelFIND
gdcm::UIDs, 778
GeneralPurposeWorklistManagementMetaSOPClass
gdcm::UIDs, 778
GeneralRelevantPatientInformationQuery
gdcm::UIDs, 778
Graphics
gdcm::Overlay, 550
GrayscaleSoftcopyPresentationStateStorageSOPClass
gdcm::MediaStorage, 510
gdcm::UIDs, 777
green
gdcm::terminal, 137

HSV
gdcm::PhotometricInterpretation, 572
HangingProtocolInformationModelFIND
gdcm::UIDs, 779
HangingProtocolInformationModelMOVE
gdcm::UIDs, 779
HangingProtocolStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 779
HardcopyColorImageStorageSOPClassRetired
gdcm::UIDs, 776
HardcopyGrayscaleImageStorage
gdcm::MediaStorage, 510
HardcopyGrayscaleImageStorageSOPClassRetired
gdcm::UIDs, 776
HemodynamicWaveformStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 776
hidden
gdcm::terminal, 137

ICBM452T1FrameofReference
gdcm::UIDs, 775
ICBMSingleSubjectMRIFrameofReference
gdcm::UIDs, 775
INT12
gdcm::PixelFormat, 575
INT16
gdcm::PixelFormat, 575
INT32
gdcm::PixelFormat, 575
INT64
gdcm::PixelFormat, 575
INT8
gdcm::PixelFormat, 575
INTERFILE
gdcm::CSAHeader, 269
INVALID
gdcm::VR, 854
IS
gdcm::VR, 855
ImageOverlayBoxSOPClassRetired
gdcm::UIDs, 776
Implicit
gdcm::TransferSyntax, 760
ImplicitVRBigEndianACRNEMA
gdcm::TransferSyntax, 761
ImplicitVRBigEndianPrivateGE
gdcm::TransferSyntax, 760
ImplicitVRLittleEndian
gdcm::TransferSyntax, 760
ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
gdcm::UIDs, 773
InstanceAvailabilityNotificationSOPClass
gdcm::UIDs, 778
Invalid
gdcm::Overlay, 550
gdcm::Usage, 837

JPEG2000
gdcm::TransferSyntax, 761
JPEG2000_COMPRESSION
vtkGDCMImageWriter, 875
JPEG2000ImageCompression
gdcm::UIDs, 774
JPEG2000ImageCompressionLosslessOnly
gdcm::UIDs, 774
JPEG2000Lossless
gdcm::TransferSyntax, 761
JPEG2000Part2
gdcm::TransferSyntax, 761
JPEG2000Part2Lossless
gdcm::TransferSyntax, 761
JPEG2000Part2MulticomponentImageCompression
gdcm::UIDs, 774
JPEG2000Part2MulticomponentImageCompression↩
LosslessOnly
gdcm::UIDs, 774
JPEG_COMPRESSION
vtkGDCMImageWriter, 875
JPEGBaselineProcess1
gdcm::TransferSyntax, 760
JPEGBaselineProcess1DefaultTransferSyntaxforLossyJ↩
PEG8BitImageCompression
gdcm::UIDs, 773
JPEGExtendedHierarchicalProcess1618Retired
gdcm::UIDs, 774
JPEGExtendedHierarchicalProcess1719Retired
gdcm::UIDs, 774
JPEGExtendedProcess24DefaultTransferSyntaxfor↩
LossyJPEG12BitImageCompressionProcess4only
gdcm::UIDs, 773
JPEGExtendedProcess2_4
gdcm::TransferSyntax, 760
JPEGExtendedProcess35Retired
gdcm::UIDs, 773
JPEGExtendedProcess3_5
gdcm::TransferSyntax, 760
JPEGFullProgressionHierarchicalProcess2426Retired
gdcm::UIDs, 774
JPEGFullProgressionHierarchicalProcess2527Retired
gdcm::UIDs, 774
JPEGFullProgressionNonHierarchicalProcess1012↩
Retired
gdcm::UIDs, 773
JPEGFullProgressionNonHierarchicalProcess1113↩
Retired
gdcm::UIDs, 773
JPEGFullProgressionProcess10_12
gdcm::TransferSyntax, 760
JPEGLS_COMPRESSION
vtkGDCMImageWriter, 875
JPEGLSLossless
gdcm::TransferSyntax, 760
JPEGLSLosslessImageCompression
gdcm::UIDs, 774
JPEGLSLossyNearLosslessImageCompression
gdcm::UIDs, 774
JPEGLSNearLossless
gdcm::TransferSyntax, 760
JPEGLosslessHierarchicalProcess28Retired
gdcm::UIDs, 774
JPEGLosslessHierarchicalProcess29Retired
gdcm::UIDs, 774
JPEGLosslessNonHierarchicalFirstOrderPrediction↩
Process14SelectionValue1DefaultTransfer↩
SyntaxforLosslessJPEGImageCompression
gdcm::UIDs, 774
JPEGLosslessNonHierarchicalProcess14

gdcm::UIDs, 773
JPEGLosslessNonHierarchicalProcess15Retired
gdcm::UIDs, 774
JPEGLosslessProcess14
gdcm::TransferSyntax, 760
JPEGLosslessProcess14_1
gdcm::TransferSyntax, 760
JPEGSpectralSelectionHierarchicalProcess2022Retired
gdcm::UIDs, 774
JPEGSpectralSelectionHierarchicalProcess2123Retired
gdcm::UIDs, 774
JPEGSpectralSelectionNonHierarchicalProcess68Retired
gdcm::UIDs, 773
JPEGSpectralSelectionNonHierarchicalProcess79Retired
gdcm::UIDs, 773
JPEGSpectralSelectionProcess6_8
gdcm::TransferSyntax, 760
JPIPReferenced
gdcm::TransferSyntax, 761
gdcm::UIDs, 774
JPIPReferencedDeflate
gdcm::UIDs, 774
JunkAfterDocElementError
gdcm::Parser, 556

KeyObjectSelectionDocument
gdcm::MediaStorage, 510
KeyObjectSelectionDocumentStorage
gdcm::UIDs, 778

LD_ALL
gdcm, 126
LD_NOSEQ
gdcm, 126
LD_NOSHADOW
gdcm, 126
LD_NOSHADOWSEQ
gdcm, 126
LINE
gdcm::MeshPrimitive, 519
LO
gdcm::VR, 855
LOADBULKDATA
gdcm::XMLPrinter, 927
LT
gdcm::VR, 855
LeadECGWaveformStorage
gdcm::MediaStorage, 510
LittleEndian
gdcm::SwapCode, 731

MAGNIFIED
gdcm::Spacing, 696
MANUAL
gdcm::Segment, 651
MONOCHROME1
gdcm::PhotometricInterpretation, 572
MONOCHROME2
gdcm::PhotometricInterpretation, 572
MPEG2MainProfile
gdcm::TransferSyntax, 761
MPEG2MainProfileMainLevel
gdcm::UIDs, 774
MPType_END
gdcm::MeshPrimitive, 519
MRImageStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
MRSpectroscopyStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
MS_END
gdcm::MediaStorage, 511
magenta
gdcm::terminal, 138
MammographyCADSR
gdcm::MediaStorage, 510
MammographyCADSRStorage
gdcm::UIDs, 777
Mandatory
gdcm::Usage, 837
MediaCreationManagementSOPClassUID
gdcm::UIDs, 776
MediaStorageDirectoryStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 774
ModalityPerformedProcedureStepNotificationSOPClass
gdcm::UIDs, 775
ModalityPerformedProcedureStepRetrieveSOPClass
gdcm::UIDs, 775
ModalityPerformedProcedureStepSOPClass
gdcm::MediaStorage, 511
gdcm::UIDs, 775
ModalityWorklistInformationModelFIND
gdcm::UIDs, 778
MultiframeGrayscaleByteSecondaryCaptureImage↩
Storage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
MultiframeGrayscaleWordSecondaryCaptureImage↩
Storage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
MultiframeSingleBitSecondaryCaptureImageStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
MultiframeTrueColorSecondaryCaptureImageStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 776

N_ACTION_RQ
gdcm::network::DIMSE, 325
N_ACTION_RSP
gdcm::network::DIMSE, 325
N_CREATE_RQ
gdcm::network::DIMSE, 325
N_CREATE_RSP
gdcm::network::DIMSE, 325
N_DELETE_RQ
gdcm::network::DIMSE, 325
N_DELETE_RSP
gdcm::network::DIMSE, 325
N_EVENT_REPORT_RQ
gdcm::network::DIMSE, 325
N_EVENT_REPORT_RSP
gdcm::network::DIMSE, 325
N_GET_RQ
gdcm::network::DIMSE, 325
N_GET_RSP
gdcm::network::DIMSE, 325
N_SET_RQ
gdcm::network::DIMSE, 325
N_SET_RSP
gdcm::network::DIMSE, 325
NO
gdcm::Surface, 720
NO_COMPRESSION
vtkGDCMImageWriter, 875
NOMAGIC
gdcm::CSAHeader, 269
NoElementsError
gdcm::Parser, 556
NoError
gdcm::Parser, 556
NoMemoryError
gdcm::Parser, 556
NoObject
gdcm::MediaStorage, 511
NuclearMedicineImageStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 777
NuclearMedicineImageStorageRetired
gdcm::MediaStorage, 509
gdcm::UIDs, 776

OB
gdcm::VR, 855
OB_OW
gdcm::VR, 855
OBLIQUE
gdcm::Orientation, 546
OD
gdcm::VR, 855
OF
gdcm::VR, 855
OPENSSL
gdcm::CryptoFactory, 260
OPENSSLP7
gdcm::CryptoFactory, 260
OW
gdcm::VR, 855
ObjectEnd
gdcm::MediaStorage, 511
OnlyUUID
gdcm::XMLPrinter, 927
operator!=
gdcm, 126
operator<<
gdcm, 126–130
operator>>
gdcm, 130
operator==
gdcm, 130
OphthalmicPhotography16BitImageStorage
gdcm::UIDs, 777
OphthalmicPhotography8BitImageStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 777
OphthalmicTomographyImageStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 777

PALETTE_COLOR
gdcm::PhotometricInterpretation, 572
PDF
gdcm::MediaStorage, 511
PETImageStorage
gdcm::MediaStorage, 510
PHILIPS
gdcm::Dicts, 322
PI_END
gdcm::PhotometricInterpretation, 572
PN
gdcm::VR, 855
POINTS
gdcm::Surface, 720
PatientRootQueryRetrieveInformationModelFIND
gdcm::UIDs, 778
PatientRootQueryRetrieveInformationModelGET
gdcm::UIDs, 778
PatientRootQueryRetrieveInformationModelMOVE
gdcm::UIDs, 778
PatientStudyOnlyQueryRetrieveInformationModelFIND↩
Retired
gdcm::UIDs, 778
PatientStudyOnlyQueryRetrieveInformationModelGET↩
Retired
gdcm::UIDs, 778

PatientStudyOnlyQueryRetrieveInformationModelMOVE↩ Retired
gdcm::UIDs, 778
Philips3D
gdcm::MediaStorage, 510
PhilipsPrivateMRSyntheticImageStorage
gdcm::MediaStorage, 511
PositronEmissionTomographyImageStorage
gdcm::UIDs, 778
PresentationLUTSOPClass
gdcm::UIDs, 776
PrintJobSOPClass
gdcm::UIDs, 775
PrintQueueManagementSOPClassRetired
gdcm::UIDs, 776
PrintQueueSOPInstanceRetired
gdcm::UIDs, 776
PrinterConfigurationRetrievalSOPClass
gdcm::UIDs, 775
PrinterConfigurationRetrievalSOPInstance
gdcm::UIDs, 775
PrinterSOPClass
gdcm::UIDs, 775
PrinterSOPInstance
gdcm::UIDs, 775
ProceduralEventLoggingSOPClass
gdcm::UIDs, 775
ProceduralEventLoggingSOPInstance
gdcm::UIDs, 775
ProcedureLogStorage
gdcm::UIDs, 777
ProductCharacteristicsQuerySOPClass
gdcm::UIDs, 779
PseudoColorSoftcopyPresentationStateStorageSOP↩ Class
gdcm::UIDs, 777
PullPrintRequestSOPClassRetired
gdcm::UIDs, 776
PullStoredPrintManagementMetaSOPClassRetired
gdcm::UIDs, 776

RED
gdcm::LookupTable, 498
RFC2557MIMEencapsulation
gdcm::UIDs, 774
RGB
gdcm::PhotometricInterpretation, 572
RLE_COMPRESSION
vtkGDCMImageWriter, 875
RLELossless
gdcm::TransferSyntax, 761
gdcm::UIDs, 774
ROI
gdcm::Overlay, 550
RTBeamsDeliveryInstructionStorageSupplement74↩ FrozenDraft
gdcm::UIDs, 778
RTBeamsTreatmentRecordStorage
gdcm::UIDs, 778
RTBrachyTreatmentRecordStorage
gdcm::UIDs, 778
RTConventionalMachineVerificationSupplement74↩ FrozenDraft
gdcm::UIDs, 778
RTDoseStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 778
RTImageStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 778
RTIonBeamsTreatmentRecordStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 778
RTIonMachineVerificationSupplement74FrozenDraft
gdcm::UIDs, 778
RTIonPlanStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 778
RTPlanStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 778
RTStructureSetStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 778
RTTreatmentSummaryRecordStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 778
RawDataStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 777
RealWorldValueMappingStorage
gdcm::UIDs, 777
red
gdcm::terminal, 137
ReferencedColorPrintManagementMetaSOPClassRetired
gdcm::UIDs, 775
ReferencedGrayscalePrintManagementMetaSOPClass↩ Retired
gdcm::UIDs, 775
ReferencedImageBoxSOPClassRetired
gdcm::UIDs, 775
reset
gdcm::terminal, 137
reverse
gdcm::terminal, 137

SAGITTAL
gdcm::Orientation, 546

SH
gdcm::VR, 855
SIEMENS
gdcm::Dicts, 322
SINGLEBIT
gdcm::PixelFormat, 575
SL
gdcm::VR, 855
SLICE_ORIENTATION_XY
vtkImageColorViewer, 897
SLICE_ORIENTATION_XZ
vtkImageColorViewer, 897
SLICE_ORIENTATION_YZ
vtkImageColorViewer, 897
SPM2AVG152PDFrameofReference
gdcm::UIDs, 774
SPM2AVG152T1FrameofReference
gdcm::UIDs, 774
SPM2AVG152T2FrameofReference
gdcm::UIDs, 774
SPM2AVG305T1FrameofReference
gdcm::UIDs, 774
SPM2BRAINMASKFrameofReference
gdcm::UIDs, 774
SPM2CSFFrameofReference
gdcm::UIDs, 774
SPM2EPIFrameofReference
gdcm::UIDs, 774
SPM2FILT1FrameofReference
gdcm::UIDs, 774
SPM2GRAYFrameofReference
gdcm::UIDs, 774
SPM2PDFrameofReference
gdcm::UIDs, 774
SPM2PETFrameofReference
gdcm::UIDs, 774
SPM2SINGLESUBJT1FrameofReference
gdcm::UIDs, 774
SPM2SPECTFrameofReference
gdcm::UIDs, 774
SPM2T1FrameofReference
gdcm::UIDs, 774
SPM2T2FrameofReference
gdcm::UIDs, 774
SPM2TRANSMFrameofReference
gdcm::UIDs, 774
SPM2WHITEFrameofReference
gdcm::UIDs, 774
SQ
gdcm::VR, 855
SS
gdcm::VR, 855
ST
gdcm::VR, 855
STATES_END
gdcm::Surface, 720
SURFACE
gdcm::Surface, 720
SV10
gdcm::CSAHeader, 269
SecondaryCaptureImageStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
Segmentation
gdcm::MediaStorage, 511
SegmentationStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 777
SpacialFiducialsStorage
gdcm::MediaStorage, 510
SpacialRegistrationStorage
gdcm::MediaStorage, 510
SpatialFiducialsStorage
gdcm::UIDs, 777
SpatialRegistrationStorage
gdcm::UIDs, 777
StandaloneCurveStorage
gdcm::MediaStorage, 510
StandaloneCurveStorageRetired
gdcm::UIDs, 776
StandaloneModalityLUTStorage
gdcm::MediaStorage, 510
StandaloneModalityLUTStorageRetired
gdcm::UIDs, 777
StandaloneOverlayStorage
gdcm::MediaStorage, 510
StandaloneOverlayStorageRetired
gdcm::UIDs, 776
StandalonePETCurveStorageRetired
gdcm::UIDs, 778
StandaloneVOILUTStorage
gdcm::MediaStorage, 510
StandaloneVOILUTStorageRetired
gdcm::UIDs, 777
StereometricRelationshipStorage
gdcm::UIDs, 777
StorageCommitmentPullModelSOPClassRetired
gdcm::UIDs, 775
StorageCommitmentPullModelSOPInstanceRetired
gdcm::UIDs, 775
StorageCommitmentPushModelSOPClass
gdcm::UIDs, 775
StorageCommitmentPushModelSOPInstance
gdcm::UIDs, 775
StorageServiceClass
gdcm::UIDs, 775
StoredPrintStorageSOPClassRetired
gdcm::UIDs, 776

StudyComponentManagementSOPClass
gdcm::MediaStorage, 510
StudyComponentManagementSOPClassRetired
gdcm::UIDs, 775
StudyRootQueryRetrieveInformationModelFIND
gdcm::UIDs, 778
StudyRootQueryRetrieveInformationModelGET
gdcm::UIDs, 778
StudyRootQueryRetrieveInformationModelMOVE
gdcm::UIDs, 778
SubstanceAdministrationLoggingSOPClass
gdcm::UIDs, 775
SubstanceAdministrationLoggingSOPInstance
gdcm::UIDs, 775
SubstanceApprovalQuerySOPClass
gdcm::UIDs, 779
SurfaceSegmentationStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 780
SyntaxError
gdcm::Parser, 556

T1
gdcm::Type, 766
T1C
gdcm::Type, 766
T2
gdcm::Type, 766
T2C
gdcm::Type, 766
T3
gdcm::Type, 766
TM
gdcm::VR, 855
TRIANGLE
gdcm::MeshPrimitive, 519
TRIANGLE_FAN
gdcm::MeshPrimitive, 519
TRIANGLE_STRIP
gdcm::MeshPrimitive, 519
TS_END
gdcm::TransferSyntax, 761
TagMismatchError
gdcm::Parser, 556
TalairachBrainAtlasFrameofReference
gdcm::UIDs, 774
TextSRStorageTrialRetired
gdcm::UIDs, 777
ToshibaPrivateDataStorage
gdcm::MediaStorage, 510

UI
gdcm::VR, 855
UINT12
gdcm::PixelFormat, 575
UINT16
gdcm::PixelFormat, 575
UINT32
gdcm::PixelFormat, 575
UINT64
gdcm::PixelFormat, 575
UINT8
gdcm::PixelFormat, 575
UL
gdcm::VR, 855
UN
gdcm::VR, 855
UNKNOW
gdcm::PhotometricInterpretation, 572
UNKNOWN
gdcm::CSAHeader, 269
gdcm::LookupTable, 498
gdcm::Orientation, 546
gdcm::PixelFormat, 575
gdcm::Spacing, 696
gdcm::Surface, 720
gdcm::Type, 766
URI
gdcm::MediaStorage, 511
US
gdcm::VR, 855
US_SS
gdcm::VR, 855
US_SS_OW
gdcm::VR, 855
UT
gdcm::VR, 855
uid_1_2_840_10008_15_0_3_1
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_10
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_11
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_12
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_13
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_14
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_15
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_16
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_17
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_18
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_19
gdcm::UIDs, 786

uid_1_2_840_10008_15_0_3_2
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_20
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_21
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_22
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_23
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_24
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_25
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_26
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_27
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_28
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_29
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_3
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_30
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_31
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_3_4
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_5
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_6
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_7
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_8
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_3_9
gdcm::UIDs, 785
uid_1_2_840_10008_15_0_4_1
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_4_2
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_4_3
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_4_4
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_4_5
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_4_6
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_4_7
gdcm::UIDs, 786
uid_1_2_840_10008_15_0_4_8
gdcm::UIDs, 786
uid_1_2_840_10008_1_1
gdcm::UIDs, 780
uid_1_2_840_10008_1_2
gdcm::UIDs, 780
uid_1_2_840_10008_1_20_1
gdcm::UIDs, 781
uid_1_2_840_10008_1_20_1_1
gdcm::UIDs, 781
uid_1_2_840_10008_1_20_2
gdcm::UIDs, 781
uid_1_2_840_10008_1_20_2_1
gdcm::UIDs, 781
uid_1_2_840_10008_1_2_1
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_1_99
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_2
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_100
gdcm::UIDs, 781
uid_1_2_840_10008_1_2_4_50
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_51
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_52
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_53
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_54
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_55
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_56
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_57
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_58
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_59
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_60
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_61
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_62
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_63
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_64
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_65
gdcm::UIDs, 780

uid_1_2_840_10008_1_2_4_66
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_70
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_80
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_81
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_90
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_91
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_92
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_93
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_94
gdcm::UIDs, 780
uid_1_2_840_10008_1_2_4_95
gdcm::UIDs, 781
uid_1_2_840_10008_1_2_5
gdcm::UIDs, 781
uid_1_2_840_10008_1_2_6_1
gdcm::UIDs, 781
uid_1_2_840_10008_1_2_6_2
gdcm::UIDs, 781
uid_1_2_840_10008_1_3_10
gdcm::UIDs, 781
uid_1_2_840_10008_1_40
gdcm::UIDs, 781
uid_1_2_840_10008_1_40_1
gdcm::UIDs, 781
uid_1_2_840_10008_1_42
gdcm::UIDs, 781
uid_1_2_840_10008_1_42_1
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_1
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_10
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_11
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_12
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_13
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_14
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_15
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_16
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_17
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_18
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_2
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_3
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_4
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_5
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_6
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_7
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_8
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_1_9
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_2_1
gdcm::UIDs, 781
uid_1_2_840_10008_1_4_2_2
gdcm::UIDs, 781
uid_1_2_840_10008_1_9
gdcm::UIDs, 781
uid_1_2_840_10008_2_16_4
gdcm::UIDs, 781
uid_1_2_840_10008_2_6_1
gdcm::UIDs, 781
uid_1_2_840_10008_3_1_1_1
gdcm::UIDs, 781
uid_1_2_840_10008_3_1_2_1_1
gdcm::UIDs, 781
uid_1_2_840_10008_3_1_2_1_4
gdcm::UIDs, 781
uid_1_2_840_10008_3_1_2_2_1
gdcm::UIDs, 781
uid_1_2_840_10008_3_1_2_3_1
gdcm::UIDs, 781
uid_1_2_840_10008_3_1_2_3_2
gdcm::UIDs, 782
uid_1_2_840_10008_3_1_2_3_3
gdcm::UIDs, 782
uid_1_2_840_10008_3_1_2_3_4
gdcm::UIDs, 782
uid_1_2_840_10008_3_1_2_3_5
gdcm::UIDs, 782
uid_1_2_840_10008_3_1_2_5_1
gdcm::UIDs, 782
uid_1_2_840_10008_3_1_2_5_4
gdcm::UIDs, 782
uid_1_2_840_10008_3_1_2_5_5
gdcm::UIDs, 782
uid_1_2_840_10008_3_1_2_6_1
gdcm::UIDs, 782

uid_1_2_840_10008_4_2
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_14
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_15
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_16
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_16_376
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_17
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_17_376
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_18
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_18_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_2
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_22
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_23
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_24
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_24_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_25
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_26
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_27
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_29
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_30
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_31
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_32
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_33
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_4
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_4_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_4_2
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_9
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_1_9_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_4_1_1_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_4_1_1_10
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_104_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_104_2
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_11
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_11_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_11_2
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_11_3
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_11_4
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_128
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_129
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_12_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_12_1_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_12_2
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_12_2_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_12_3
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_13_1_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_13_1_2
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_13_1_3
gdcm::UIDs, 786
uid_1_2_840_10008_5_1_4_1_1_1_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_4_1_1_1_1_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_4_1_1_1_2
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_4_1_1_1_2_1
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_4_1_1_1_3
gdcm::UIDs, 782
uid_1_2_840_10008_5_1_4_1_1_1_3_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_2
gdcm::UIDs, 783

uid_1_2_840_10008_5_1_4_1_1_20
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_2_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_3
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_3_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_4
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_481_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_481_2
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_481_3
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_481_4
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_481_5
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_481_6
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_481_7
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_481_8
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_481_9
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_4_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_4_2
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_5
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_6
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_66
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_66_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_66_2
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_66_3
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_66_4
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_66_5
gdcm::UIDs, 786
uid_1_2_840_10008_5_1_4_1_1_67
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_6_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_6_2
gdcm::UIDs, 786
uid_1_2_840_10008_5_1_4_1_1_7
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_77_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_2
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_3
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_4
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_77_1_6
gdcm::UIDs, 786
uid_1_2_840_10008_5_1_4_1_1_77_2
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_7_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_7_2
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_7_3
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_7_4
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_8
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_88_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_11
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_2
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_22
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_3
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_33
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_4
gdcm::UIDs, 784

uid_1_2_840_10008_5_1_4_1_1_88_40
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_50
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_59
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_65
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_88_67
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_1_9
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_9_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_9_1_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_9_1_2
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_9_1_3
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_9_2_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_9_3_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_1_9_4_1
gdcm::UIDs, 783
uid_1_2_840_10008_5_1_4_1_2_1_1
gdcm::UIDs, 784
uid_1_2_840_10008_5_1_4_1_2_1_2
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_1_2_1_3
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_1_2_2_1
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_1_2_2_2
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_1_2_2_3
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_1_2_3_1
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_1_2_3_2
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_1_2_3_3
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_31
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_32
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_32_1
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_32_2
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_32_3
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_33
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_1
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_2
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_3
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_4
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_4_1
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_4_2
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_4_3
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_4_4
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_34_5
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_37_1
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_37_2
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_37_3
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_38_1
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_38_2
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_38_3
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_41
gdcm::UIDs, 785
uid_1_2_840_10008_5_1_4_42
gdcm::UIDs, 785
UltrasoundImageStorage
gdcm::MediaStorage, 509
gdcm::UIDs, 776
UltrasoundImageStorageRetired
gdcm::MediaStorage, 509
gdcm::UIDs, 776
UltrasoundMultiFrameImageStorage
gdcm::MediaStorage, 509
UltrasoundMultiFrameImageStorageRetired
gdcm::MediaStorage, 509
UltrasoundMultiframeImageStorage
gdcm::UIDs, 776
UltrasoundMultiframeImageStorageRetired
gdcm::UIDs, 776
UndefinedEntityError
gdcm::Parser, 557
underline
gdcm::terminal, 137

UnexpectedStateError
gdcm::Parser, 557
UnifiedProcedureStepEventSOPClass
gdcm::UIDs, 778
UnifiedProcedureStepPullSOPClass
gdcm::UIDs, 778
UnifiedProcedureStepPushSOPClass
gdcm::UIDs, 778
UnifiedProcedureStepWatchSOPClass
gdcm::UIDs, 778
UnifiedWorklistandProcedureStepSOPInstance
gdcm::UIDs, 778
UnifiedWorklistandProcedureStepServiceClass
gdcm::UIDs, 778
Unknown
gdcm::SwapCode, 731
gdcm::TransferSyntax, 760
UserOption
gdcm::Usage, 837

VERBOSE_STYLE
gdcm::Printer, 604
VERTEX
gdcm::MeshPrimitive, 519
VIEWType_END
gdcm::Surface, 720
VL16
gdcm::VR, 855
VL32
gdcm::VR, 855
VLEndoscopicImageStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 777
VLImageStorageTrialRetired
gdcm::UIDs, 777
VLMicroscopicImageStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 777
VLMultiframeImageStorageTrialRetired
gdcm::UIDs, 777
VLPhotographicImageStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 777
VLSlideCoordinatesMicroscopicImageStorage
gdcm::UIDs, 777
VLWholeSlideMicroscopyImageStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 780
VM0
gdcm::VM, 850
VM1
gdcm::VM, 850
VM10
gdcm::VM, 850
VM12
gdcm::VM, 850
VM16
gdcm::VM, 850
VM18
gdcm::VM, 850
VM1_2
gdcm::VM, 851
VM1_3
gdcm::VM, 851
VM1_32
gdcm::VM, 851
VM1_4
gdcm::VM, 851
VM1_5
gdcm::VM, 851
VM1_8
gdcm::VM, 851
VM1_99
gdcm::VM, 851
VM1_n
gdcm::VM, 851
VM2
gdcm::VM, 850
VM24
gdcm::VM, 850
VM256
gdcm::VM, 851
VM28
gdcm::VM, 850
VM2_2n
gdcm::VM, 851
VM2_n
gdcm::VM, 851
VM3
gdcm::VM, 850
VM30_30n
gdcm::VM, 851
VM32
gdcm::VM, 850
VM35
gdcm::VM, 850
VM3_3n
gdcm::VM, 851
VM3_4
gdcm::VM, 851
VM3_n
gdcm::VM, 851
VM4
gdcm::VM, 850
VM47_47n
gdcm::VM, 851
VM4_4n
gdcm::VM, 851

VM5
gdcm::VM, 850
VM6
gdcm::VM, 850
VM6_6n
gdcm::VM, 851
VM7_7n
gdcm::VM, 851
VM8
gdcm::VM, 850
VM9
gdcm::VM, 850
VM99
gdcm::VM, 851
VM_END
gdcm::VM, 851
VMType
gdcm::Attribute, 171
gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 177
VOILUTBoxSOPClass
gdcm::UIDs, 776
VR_END
gdcm::VR, 855
VR_VM1
gdcm::VR, 855
VRALL
gdcm::VR, 855
VRASCII
gdcm::VR, 855
VRBINARY
gdcm::VR, 855
VT100
gdcm::terminal, 138
value
gdcm::STATIC_ASSERTION_FAILURE< true >, 700
VerificationSOPClass
gdcm::UIDs, 773
Video
gdcm::MediaStorage, 511
VideoEndoscopicImageStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 777
VideoMicroscopicImageStorage
gdcm::UIDs, 777
VideoPhotographicImageStorage
gdcm::UIDs, 777
vtkGDCMImageWriter
JPEG2000_COMPRESSION, 875
JPEG_COMPRESSION, 875
JPEGLS_COMPRESSION, 875
NO_COMPRESSION, 875
RLE_COMPRESSION, 875
vtkImageColorViewer
SLICE_ORIENTATION_XY, 897
SLICE_ORIENTATION_XZ, 897
SLICE_ORIENTATION_YZ, 897

WIREFRAME
gdcm::Surface, 720
Waveform
gdcm::MediaStorage, 511
WaveformStorageTrialRetired
gdcm::UIDs, 776
white
gdcm::terminal, 138

XML
gdcm::Printer, 604
XMLEncoding
gdcm::UIDs, 774
XRay3DAngiographicImageStorage
gdcm::MediaStorage, 511
gdcm::UIDs, 777
XRay3DCraniofacialImageStorage
gdcm::UIDs, 777
XRayAngiographicBiPlaneImageStorageRetired
gdcm::MediaStorage, 510
gdcm::UIDs, 777
XRayAngiographicImageStorage
gdcm::MediaStorage, 510
gdcm::UIDs, 777
XRayRadiationDoseSR
gdcm::MediaStorage, 511
XRayRadiationDoseSRStorage
gdcm::UIDs, 778
XRayRadiofluoroscopicImageStorage
gdcm::UIDs, 777
XRayRadiofluoroscopingImageStorage
gdcm::MediaStorage, 510

YBR_FULL
gdcm::PhotometricInterpretation, 572
YBR_FULL_422
gdcm::PhotometricInterpretation, 572
YBR_ICT
gdcm::PhotometricInterpretation, 572
YBR_PARTIAL_420
gdcm::PhotometricInterpretation, 572
YBR_PARTIAL_422
gdcm::PhotometricInterpretation, 572
YBR_RCT
gdcm::PhotometricInterpretation, 572
YES
gdcm::Surface, 720
yellow
gdcm::terminal, 137

ZEROED_OUT
gdcm::CSAHeader, 269