

GDCM
2.2.4

Generated by Doxygen 1.8.5

Sat Dec 21 2013 01:40:25

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 Metalmage (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	35
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) <code>gdcmimg</code>	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	37
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	38
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 <code>gdcmData</code>	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 PDF to PDF/DICOM.	45
11.1 SYNOPSIS	45
11.2 DESCRIPTION	45
11.3 PARAMETERS	45
11.4 options	45
11.4.1 general options	45
11.5 Usage Example	46
11.6 PDF Info Mapping	46
11.7 SEE ALSO	47
11.8 COPYRIGHT	47
12 Extract Data Element Value Field.	49
12.1 SYNOPSIS	49
12.2 DESCRIPTION	49
12.3 PARAMETERS	49
12.4 options	49
12.4.1 PARAMETERS	49
12.4.2 options	49
12.4.3 general options	49
12.5 Typical usage	50
12.5.1 Copy Attribute Value to file	50
12.5.2 Extract Pixel Data	50
12.5.3 Encapsulated Syntax	50
12.5.4 Extract fragments as single file	51
12.6 Footnote about JPEG files	52
12.7 SEE ALSO	52
12.8 COPYRIGHT	52
13 Scan a directory containing DICOM files.	53
13.1 SYNOPSIS	53
13.2 DESCRIPTION	53
13.2.1 PARAMETERS	53
13.2.2 options	53
13.2.3 general options	53

13.3 Typical usage	54
13.4 Simple usage	54
13.5 Complex usage	54
13.6 SEE ALSO	54
13.7 COPYRIGHT	54
14 Tool to execute a DICOM Query/Retrieve operation	55
14.1 SYNOPSIS	55
14.2 DESCRIPTION	55
14.3 PARAMETERS	55
14.4 options	55
14.4.1 options	55
14.4.2 mode options	55
14.4.3 C-STORE options	56
14.4.4 C-FIND/C-MOVE options	56
14.4.5 C-MOVE options	56
14.4.6 general options	56
14.4.7 environment variable	56
14.5 C-ECHO usage	57
14.6 C-STORE usage	57
14.7 C-FIND usage	57
14.8 C-MOVE usage	58
14.9 patientroot notes	58
14.10Debugging	58
14.11Port Warning	58
14.12C-STORE Warnings	59
14.13C-MOVE Warnings	59
14.14C-FIND IMAGE level (Composite Object Instance)	59
14.15Storing the Query	59
14.16DICOM Public Servers	60
14.17SEE ALSO	60
14.18COPYRIGHT	60
15 Concatenate/Extract DICOM files.	61
15.1 SYNOPSIS	61
15.2 DESCRIPTION	61
15.3 PARAMETERS	61
15.4 options	61

15.4.1 options	61
15.4.2 general options	61
15.4.3 environment variable	62
15.5 Typical usage	62
15.5.1 SIEMENS Mosaic	62
15.6 SEE ALSO	63
15.7 COPYRIGHT	63
16 Simple DICOM viewer.	65
16.1 SYNOPSIS	65
16.2 DESCRIPTION	65
16.3 PARAMETERS	65
16.4 options	65
16.4.1 options	65
16.4.2 general options	65
16.5 Typical usage	66
16.6 Simple usage	66
16.7 Wiki Link	66
16.8 SEE ALSO	66
16.9 COPYRIGHT	66
17 Todo List	67
18 Deprecated List	69
19 Bug List	71
20 Namespace Index	73
20.1 Namespace List	73
21 Hierarchical Index	75
21.1 Class Hierarchy	75
22 Class Index	83
22.1 Class List	83
23 File Index	97
23.1 File List	97
24 Namespace Documentation	103
24.1 gdcm Namespace Reference	103

24.1.1	Detailed Description	117
24.1.2	Typedef Documentation	117
24.1.2.1	AComp	117
24.1.2.2	ASComp	117
24.1.2.3	BOOL_FUNCTION_PFILE_PFILE_POINTER	117
24.1.2.4	CComp	117
24.1.2.5	DComp	117
24.1.2.6	DTComp	117
24.1.2.7	FileList	117
24.1.2.8	IconImage	118
24.1.2.9	LOComp	118
24.1.2.10	LTComp	118
24.1.2.11	MacroEntry	118
24.1.2.12	NestedMacroEntries	118
24.1.2.13	PNComp	118
24.1.2.14	SHComp	118
24.1.2.15	STComp	118
24.1.2.16	TMComp	118
24.1.2.17	UIComp	118
24.1.2.18	UTComp	118
24.1.3	Enumeration Type Documentation	118
24.1.3.1	CompOperators	118
24.1.3.2	ECharSet	118
24.1.3.3	EQueryLevel	119
24.1.3.4	EQueryType	119
24.1.3.5	ERootType	119
24.1.3.6	LodModeType	119
24.1.4	Function Documentation	119
24.1.4.1	backslash	119
24.1.4.2	GetVRFromTag	120
24.1.4.3	operator!=	120
24.1.4.4	operator!=	120
24.1.4.5	operator<<	120
24.1.4.6	operator<<	120
24.1.4.7	operator<<	120
24.1.4.8	operator<<	120
24.1.4.9	operator<<	120

24.1.4.10 operator<<	120
24.1.4.11 operator<<	120
24.1.4.12 operator<<	120
24.1.4.13 operator<<	120
24.1.4.14 operator<<	120
24.1.4.15 operator<<	121
24.1.4.16 operator<<	121
24.1.4.17 operator<<	121
24.1.4.18 operator<<	121
24.1.4.19 operator<<	121
24.1.4.20 operator<<	121
24.1.4.21 operator<<	121
24.1.4.22 operator<<	121
24.1.4.23 operator<<	121
24.1.4.24 operator<<	121
24.1.4.25 operator<<	121
24.1.4.26 operator<<	121
24.1.4.27 operator<<	121
24.1.4.28 operator<<	121
24.1.4.29 operator<<	121
24.1.4.30 operator<<	121
24.1.4.31 operator<<	121
24.1.4.32 operator<<	122
24.1.4.33 operator<<	122
24.1.4.34 operator<<	122
24.1.4.35 operator<<	122
24.1.4.36 operator<<	122
24.1.4.37 operator<<	122
24.1.4.38 operator<<	122
24.1.4.39 operator<<	122
24.1.4.40 operator<<	122
24.1.4.41 operator<<	122
24.1.4.42 operator<<	122
24.1.4.43 operator<<	122
24.1.4.44 operator<<	122
24.1.4.45 operator<<	122
24.1.4.46 operator<<	123

24.1.4.47 operator<<	123
24.1.4.48 operator<<	123
24.1.4.49 operator<<	123
24.1.4.50 operator<<	123
24.1.4.51 operator<<	123
24.1.4.52 operator<<	123
24.1.4.53 operator<<	123
24.1.4.54 operator<<	123
24.1.4.55 operator<<	123
24.1.4.56 operator<<	123
24.1.4.57 operator<<	123
24.1.4.58 operator<<	123
24.1.4.59 operator==	124
24.1.4.60 operator>>	124
24.1.4.61 operator>>	124
24.1.4.62 operator>>	124
24.1.4.63 to_string	124
24.1.4.64 TYPETOENCODING	124
24.1.5 Variable Documentation	124
24.1.5.1 GlobalInstance	124
24.1.5.2 VRBINARY	124
24.2 gdcm::network Namespace Reference	124
24.2.1 Enumeration Type Documentation	128
24.2.1.1 EEventID	128
24.2.1.2 EStateID	129
24.2.2 Function Documentation	129
24.2.2.1 GetStateIndex	129
24.2.3 Variable Documentation	129
24.2.3.1 cMaxEventID	129
24.2.3.2 cMaxStateID	129
24.3 gdcm::SegmentHelper Namespace Reference	130
24.4 gdcm::terminal Namespace Reference	130
24.4.1 Detailed Description	130
24.4.2 Enumeration Type Documentation	131
24.4.2.1 Attribute	131
24.4.2.2 Color	131
24.4.2.3 Mode	131

24.4.3	Function Documentation	131
24.4.3.1	setattribute	131
24.4.3.2	setbgcolor	131
24.4.3.3	setfgcolor	131
24.4.3.4	setmode	131
25	Class Documentation	133
25.1	gdcmm::network::AAabortPDU Class Reference	133
25.1.1	Detailed Description	134
25.1.2	Constructor & Destructor Documentation	134
25.1.2.1	AAabortPDU	134
25.1.3	Member Function Documentation	134
25.1.3.1	IsLastFragment	134
25.1.3.2	Print	134
25.1.3.3	Read	134
25.1.3.4	SetReason	135
25.1.3.5	SetSource	135
25.1.3.6	Size	135
25.1.3.7	Write	135
25.2	gdcmm::network::AAssociateACPDU Class Reference	135
25.2.1	Detailed Description	136
25.2.2	Member Typedef Documentation	137
25.2.2.1	SizeType	137
25.2.3	Constructor & Destructor Documentation	137
25.2.3.1	AAssociateACPDU	137
25.2.4	Member Function Documentation	137
25.2.4.1	AddPresentationContextAC	137
25.2.4.2	GetNumberOfPresentationContextAC	137
25.2.4.3	GetPresentationContextAC	137
25.2.4.4	GetUserInformation	137
25.2.4.5	InitFromRQ	137
25.2.4.6	IsLastFragment	137
25.2.4.7	Print	137
25.2.4.8	Read	137
25.2.4.9	SetCalledAETitle	137
25.2.4.10	SetCallingAETitle	137
25.2.4.11	Size	137

25.2.4.12 Write	137
25.2.5 Friends And Related Function Documentation	137
25.2.5.1 AAssociateRQPDU	138
25.3 gdcmm::network::AAssociateRJPDU Class Reference	138
25.3.1 Detailed Description	139
25.3.2 Constructor & Destructor Documentation	139
25.3.2.1 AAssociateRJPDU	139
25.3.3 Member Function Documentation	139
25.3.3.1 IsLastFragment	139
25.3.3.2 Print	139
25.3.3.3 Read	139
25.3.3.4 Size	139
25.3.3.5 Write	139
25.4 gdcmm::network::AAssociateRQPDU Class Reference	139
25.4.1 Detailed Description	141
25.4.2 Member Typedef Documentation	141
25.4.2.1 PresentationContextArrayType	141
25.4.2.2 SizeType	141
25.4.3 Constructor & Destructor Documentation	141
25.4.3.1 AAssociateRQPDU	141
25.4.3.2 AAssociateRQPDU	141
25.4.4 Member Function Documentation	142
25.4.4.1 AddPresentationContext	142
25.4.4.2 GetCalledAETitle	142
25.4.4.3 GetCallingAETitle	142
25.4.4.4 GetNumberOfPresentationContext	142
25.4.4.5 GetPresentationContext	142
25.4.4.6 GetPresentationContextByAbstractSyntax	142
25.4.4.7 GetPresentationContextByID	142
25.4.4.8 GetPresentationContexts	142
25.4.4.9 GetReserved43_74	142
25.4.4.10 GetUserInfo	142
25.4.4.11 IsAETitleValid	142
25.4.4.12 IsLastFragment	142
25.4.4.13 Print	142
25.4.4.14 Read	142
25.4.4.15 SetCalledAETitle	142

25.4.4.16 SetCallingAETitle	142
25.4.4.17 SetUserInformation	143
25.4.4.18 Size	143
25.4.4.19 Write	143
25.4.5 Friends And Related Function Documentation	143
25.4.5.1 AAssociateACPDU	143
25.5 gdcm::AbortEvent Class Reference	143
25.6 gdcm::network::AbstractSyntax Class Reference	144
25.6.1 Detailed Description	144
25.6.2 Constructor & Destructor Documentation	145
25.6.2.1 AbstractSyntax	145
25.6.3 Member Function Documentation	145
25.6.3.1 GetAsDataElement	145
25.6.3.2 GetName	145
25.6.3.3 operator==	145
25.6.3.4 Print	145
25.6.3.5 Read	145
25.6.3.6 SetName	145
25.6.3.7 SetNameFromUID	145
25.6.3.8 Size	145
25.6.3.9 Write	145
25.7 gdcm::AnonymizeEvent Class Reference	145
25.7.1 Detailed Description	147
25.7.2 Member Typedef Documentation	147
25.7.2.1 Self	147
25.7.2.2 Superclass	147
25.7.3 Constructor & Destructor Documentation	147
25.7.3.1 AnonymizeEvent	147
25.7.3.2 ~AnonymizeEvent	147
25.7.3.3 AnonymizeEvent	147
25.7.4 Member Function Documentation	147
25.7.4.1 CheckEvent	147
25.7.4.2 GetEventName	147
25.7.4.3 GetTag	147
25.7.4.4 MakeObject	147
25.7.4.5 SetTag	147
25.8 gdcm::Anonymizer Class Reference	148

25.8.1 Detailed Description	149
25.8.2 Constructor & Destructor Documentation	150
25.8.2.1 Anonymizer	150
25.8.2.2 ~Anonymizer	150
25.8.3 Member Function Documentation	150
25.8.3.1 BALCPPProtect	150
25.8.3.2 BasicApplicationLevelConfidentialityProfile	151
25.8.3.3 CanEmptyTag	151
25.8.3.4 Empty	151
25.8.3.5 GetBasicApplicationLevelConfidentialityProfileAttributes	151
25.8.3.6 GetCryptographicMessageSyntax	151
25.8.3.7 GetFile	151
25.8.3.8 New	151
25.8.3.9 RecurseDataSet	151
25.8.3.10 Remove	151
25.8.3.11 RemoveGroupLength	151
25.8.3.12 RemovePrivateTags	152
25.8.3.13 RemoveRetired	152
25.8.3.14 Replace	152
25.8.3.15 Replace	152
25.8.3.16 SetCryptographicMessageSyntax	152
25.8.3.17 SetFile	152
25.9 gdcmm::AnyEvent Class Reference	152
25.10gdcmm::network::ApplicationContext Class Reference	154
25.10.1 Detailed Description	154
25.10.2 Constructor & Destructor Documentation	154
25.10.2.1 ApplicationContext	154
25.10.3 Member Function Documentation	155
25.10.3.1 GetName	155
25.10.3.2 Print	155
25.10.3.3 Read	155
25.10.3.4 SetName	155
25.10.3.5 Size	155
25.10.3.6 Write	155
25.11gdcmm::ApplicationEntity Class Reference	155
25.11.1 Detailed Description	156
25.11.2 Member Function Documentation	156

25.11.2.1 IsValid	156
25.11.2.2 Print	156
25.11.2.3 SetBlob	156
25.11.2.4 Squeeze	156
25.11.3 Member Data Documentation	156
25.11.3.1 Internal	156
25.11.3.2 MaxLength	156
25.11.3.3 MaxNumberOfComponents	156
25.11.3.4 Padding	156
25.11.3.5 Separator	156
25.12gdcmm::network::AReleaseRPPDU Class Reference	157
25.12.1 Detailed Description	158
25.12.2 Constructor & Destructor Documentation	158
25.12.2.1 AReleaseRPPDU	158
25.12.3 Member Function Documentation	158
25.12.3.1 IsLastFragment	158
25.12.3.2 Print	158
25.12.3.3 Read	158
25.12.3.4 Size	158
25.12.3.5 Write	158
25.13gdcmm::network::AReleaseRQPDU Class Reference	158
25.13.1 Detailed Description	159
25.13.2 Constructor & Destructor Documentation	159
25.13.2.1 AReleaseRQPDU	160
25.13.3 Member Function Documentation	160
25.13.3.1 IsLastFragment	160
25.13.3.2 Print	160
25.13.3.3 Read	160
25.13.3.4 Size	160
25.13.3.5 Write	160
25.14gdcmm::network::ARTIMTimer Class Reference	160
25.14.1 Detailed Description	161
25.14.2 Constructor & Destructor Documentation	161
25.14.2.1 ARTIMTimer	161
25.14.3 Member Function Documentation	161
25.14.3.1 GetElapsedTime	161
25.14.3.2 GetHasExpired	161

25.14.3.3 GetTimeout	161
25.14.3.4 SetTimeout	161
25.14.3.5 Start	161
25.14.3.6 Stop	161
25.15gdcmm::ASN1 Class Reference	161
25.15.1 Detailed Description	162
25.15.2 Constructor & Destructor Documentation	162
25.15.2.1 ASN1	162
25.15.2.2 ~ASN1	162
25.15.3 Member Function Documentation	162
25.15.3.1 ParseDump	162
25.15.3.2 ParseDumpFile	162
25.15.3.3 TestPBKDF2	162
25.16gdcmm::network::AsynchronousOperationsWindowSub Class Reference	162
25.16.1 Detailed Description	162
25.16.2 Constructor & Destructor Documentation	163
25.16.2.1 AsynchronousOperationsWindowSub	163
25.16.3 Member Function Documentation	163
25.16.3.1 Print	163
25.16.3.2 Read	163
25.16.3.3 Size	163
25.16.3.4 Write	163
25.17gdcmm::Attribute< Group, Element, TVR, TVM > Class Template Reference	163
25.17.1 Detailed Description	165
25.17.2 Member Typedef Documentation	165
25.17.2.1 ArrayType	165
25.17.3 Member Enumeration Documentation	165
25.17.3.1 anonymous enum	165
25.17.4 Member Function Documentation	165
25.17.4.1 GDCM_STATIC_ASSERT	165
25.17.4.2 GDCM_STATIC_ASSERT	165
25.17.4.3 GDCM_STATIC_ASSERT	165
25.17.4.4 GetAsDataElement	166
25.17.4.5 GetDictVM	166
25.17.4.6 GetDictVR	166
25.17.4.7 GetNumberOfValues	166
25.17.4.8 GetTag	166

25.17.4.9 GetValue	167
25.17.4.10GetValue	167
25.17.4.11GetValues	167
25.17.4.12GetVM	167
25.17.4.13GetVR	167
25.17.4.14operator!=	167
25.17.4.15operator<	168
25.17.4.16operator==	168
25.17.4.17operator[]	168
25.17.4.18operator[]	168
25.17.4.19Print	168
25.17.4.20Set	168
25.17.4.21SetByteValue	168
25.17.4.22SetByteValueNoSwap	169
25.17.4.23SetFromDataElement	169
25.17.4.24SetFromDataSet	169
25.17.4.25SetValue	169
25.17.4.26SetValues	169
25.17.5 Member Data Documentation	170
25.17.5.1 Internal	170
25.18gdcmm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference	170
25.18.1 Member Typedef Documentation	172
25.18.1.1 ArrayType	172
25.18.2 Member Enumeration Documentation	172
25.18.2.1 anonymous enum	172
25.18.3 Member Function Documentation	172
25.18.3.1 GDCM_STATIC_ASSERT	172
25.18.3.2 GDCM_STATIC_ASSERT	172
25.18.3.3 GDCM_STATIC_ASSERT	172
25.18.3.4 GDCM_STATIC_ASSERT	172
25.18.3.5 GetAsDataElement	172
25.18.3.6 GetDictVM	173
25.18.3.7 GetDictVR	173
25.18.3.8 GetNumberOfValues	173
25.18.3.9 GetTag	173
25.18.3.10GetValue	173
25.18.3.11GetValue	173

25.18.3.12	GetValues	173
25.18.3.13	GetVM	173
25.18.3.14	GetVR	173
25.18.3.15	operator!=	173
25.18.3.16	operator<	173
25.18.3.17	operator==	174
25.18.3.18	Print	174
25.18.3.19	Set	174
25.18.3.20	SetByteValue	174
25.18.3.21	SetByteValueNoSwap	174
25.18.3.22	SetFromDataElement	174
25.18.3.23	SetFromDataSet	174
25.18.3.24	SetValue	174
25.18.4	Member Data Documentation	175
25.18.4.1	Internal	175
25.19	gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference	175
25.19.1	Member Function Documentation	176
25.19.1.1	GetVM	176
25.20	gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference	176
25.20.1	Member Function Documentation	177
25.20.1.1	GetVM	177
25.21	gdcmm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference	177
25.21.1	Member Typedef Documentation	179
25.21.1.1	ArrayType	179
25.21.2	Constructor & Destructor Documentation	179
25.21.2.1	Attribute	179
25.21.2.2	~Attribute	179
25.21.3	Member Function Documentation	179
25.21.3.1	GDCM_STATIC_ASSERT	179
25.21.3.2	GDCM_STATIC_ASSERT	179
25.21.3.3	GDCM_STATIC_ASSERT	179
25.21.3.4	GetAsDataElement	179
25.21.3.5	GetDictVM	179
25.21.3.6	GetDictVR	179
25.21.3.7	GetNumberOfValues	180
25.21.3.8	GetTag	180
25.21.3.9	GetValue	180

25.21.3.10GetValue	180
25.21.3.11GetValues	180
25.21.3.12GetVM	180
25.21.3.13GetVR	180
25.21.3.14operator[]	180
25.21.3.15operator[]	180
25.21.3.16Print	180
25.21.3.17Set	181
25.21.3.18SetByteValue	181
25.21.3.19SetFromDataElement	181
25.21.3.20SetFromDataSet	181
25.21.3.21SetNumberOfValues	181
25.21.3.22SetValue	181
25.21.3.23SetValue	181
25.21.3.24SetValues	182
25.22gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	182
25.22.1 Member Function Documentation	183
25.22.1.1 GetVM	183
25.23gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference	183
25.23.1 Member Function Documentation	184
25.23.1.1 GetVM	185
25.24gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference	185
25.24.1 Member Function Documentation	186
25.24.1.1 GetVM	186
25.25gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference	186
25.25.1 Member Function Documentation	187
25.25.1.1 GetVM	188
25.26gdcmm::AudioCodec Class Reference	188
25.26.1 Detailed Description	189
25.26.2 Constructor & Destructor Documentation	189
25.26.2.1 AudioCodec	189
25.26.2.2 ~AudioCodec	189
25.26.3 Member Function Documentation	189
25.26.3.1 CanCode	189
25.26.3.2 CanDecode	190
25.26.3.3 Decode	190
25.27gdcmm::Base64 Class Reference	190

25.27.1 Detailed Description	190
25.27.2 Constructor & Destructor Documentation	190
25.27.2.1 Base64	190
25.27.2.2 ~Base64	190
25.27.3 Member Function Documentation	191
25.27.3.1 Decode	191
25.27.3.2 Encode	192
25.27.3.3 GetDecodeLength	192
25.27.3.4 GetEncodeLength	192
25.28gdcmm::network::BaseCompositeMessage Class Reference	192
25.28.1 Detailed Description	193
25.28.2 Member Function Documentation	194
25.28.2.1 ConstructPDV	194
25.29gdcmm::network::BasePDU Class Reference	194
25.29.1 Detailed Description	195
25.29.2 Constructor & Destructor Documentation	196
25.29.2.1 ~BasePDU	196
25.29.3 Member Function Documentation	196
25.29.3.1 IsLastFragment	196
25.29.3.2 Print	196
25.29.3.3 Read	196
25.29.3.4 Size	196
25.29.3.5 Write	196
25.30gdcmm::BaseRootQuery Class Reference	196
25.30.1 Detailed Description	198
25.30.2 Constructor & Destructor Documentation	198
25.30.2.1 BaseRootQuery	198
25.30.2.2 ~BaseRootQuery	198
25.30.3 Member Function Documentation	199
25.30.3.1 AddQueryDataSet	199
25.30.3.2 Construct	199
25.30.3.3 GetAbstractSyntaxUID	199
25.30.3.4 GetQueryDataSet	199
25.30.3.5 GetQueryDataSet	199
25.30.3.6 GetQueryLevelFromQueryRoot	199
25.30.3.7 GetQueryLevelFromString	199
25.30.3.8 GetQueryLevelString	199

25.30.3.9 GetTagListByLevel	199
25.30.3.10 InitializeDataSet	199
25.30.3.11 Print	199
25.30.3.12 SetSearchParameter	199
25.30.3.13 SetSearchParameter	199
25.30.3.14 SetSearchParameter	200
25.30.3.15 ValidateQuery	200
25.30.3.16 WriteHelpFile	200
25.30.3.17 WriteQuery	200
25.30.4 Friends And Related Function Documentation	200
25.30.4.1 QueryFactory	200
25.30.5 Member Data Documentation	200
25.30.5.1 mDataSet	200
25.30.5.2 mHelpDescription	200
25.30.5.3 mImage	200
25.30.5.4 mPatient	200
25.30.5.5 mRootType	200
25.30.5.6 mSeries	200
25.30.5.7 mStudy	200
25.31 gdcmm::SegmentHelper::BasicCodedEntry Struct Reference	200
25.31.1 Detailed Description	202
25.31.2 Constructor & Destructor Documentation	202
25.31.2.1 BasicCodedEntry	202
25.31.2.2 BasicCodedEntry	202
25.31.2.3 BasicCodedEntry	202
25.31.3 Member Function Documentation	202
25.31.3.1 IsEmpty	202
25.31.4 Member Data Documentation	202
25.31.4.1 CM	202
25.31.4.2 CSD	202
25.31.4.3 CSV	202
25.31.4.4 CV	203
25.32 gdcmm::BasicOffsetTable Class Reference	203
25.32.1 Detailed Description	204
25.32.2 Constructor & Destructor Documentation	204
25.32.2.1 BasicOffsetTable	204
25.32.3 Member Function Documentation	205

25.32.3.1 Read	205
25.32.4 Friends And Related Function Documentation	205
25.32.4.1 operator<<	205
25.33gdcmm::Bitmap Class Reference	205
25.33.1 Detailed Description	208
25.33.2 Member Typedef Documentation	208
25.33.2.1 LUTPtr	208
25.33.3 Constructor & Destructor Documentation	208
25.33.3.1 Bitmap	208
25.33.3.2 ~Bitmap	208
25.33.4 Member Function Documentation	208
25.33.4.1 AreOverlaysInPixelData	208
25.33.4.2 Clear	208
25.33.4.3 ComputeLossyFlag	208
25.33.4.4 GetBuffer	208
25.33.4.5 GetBuffer2	208
25.33.4.6 GetBufferLength	208
25.33.4.7 GetColumns	209
25.33.4.8 GetDataElement	209
25.33.4.9 GetDataElement	209
25.33.4.10GetDimension	209
25.33.4.11GetDimensions	209
25.33.4.12GetLUT	209
25.33.4.13GetLUT	209
25.33.4.14GetNeedByteSwap	209
25.33.4.15GetNumberOfDimensions	209
25.33.4.16GetPhotometricInterpretation	209
25.33.4.17GetPixelFormat	210
25.33.4.18GetPixelFormat	210
25.33.4.19GetPlanarConfiguration	210
25.33.4.20GetRows	210
25.33.4.21GetTransferSyntax	210
25.33.4.22IsEmpty	210
25.33.4.23IsLossy	210
25.33.4.24IsTransferSyntaxCompatible	210
25.33.4.25Print	210
25.33.4.26SetColumns	210

25.33.4.27SetDataElement	210
25.33.4.28SetDimension	211
25.33.4.29SetDimensions	211
25.33.4.30SetLossyFlag	211
25.33.4.31SetLUT	211
25.33.4.32SetNeedByteSwap	211
25.33.4.33SetNumberOfDimensions	211
25.33.4.34SetPhotometricInterpretation	211
25.33.4.35SetPixelFormat	211
25.33.4.36SetPlanarConfiguration	211
25.33.4.37SetRows	212
25.33.4.38SetTransferSyntax	212
25.33.4.39TryJPEG2000Codec	212
25.33.4.40TryJPEG2000Codec2	212
25.33.4.41TryJPEGCodec	212
25.33.4.42TryJPEGCodec2	212
25.33.4.43TryJPEGLSCodec	212
25.33.4.44TryKAKADUCodec	212
25.33.4.45TryPVRGCodec	212
25.33.4.46TryRAWCodec	212
25.33.4.47TryRLECodec	212
25.33.5 Friends And Related Function Documentation	212
25.33.5.1 ImageChangeTransferSyntax	212
25.33.5.2 PixmapReader	212
25.33.6 Member Data Documentation	212
25.33.6.1 Dimensions	212
25.33.6.2 LossyFlag	212
25.33.6.3 LUT	212
25.33.6.4 NeedByteSwap	212
25.33.6.5 NumberOfDimensions	212
25.33.6.6 PF	212
25.33.6.7 PI	212
25.33.6.8 PixelData	212
25.33.6.9 PlanarConfiguration	213
25.33.6.10TS	213
25.34gdcm::BitmapToBitmapFilter Class Reference	213
25.34.1 Detailed Description	214

25.34.2 Constructor & Destructor Documentation	214
25.34.2.1 BitmapToBitmapFilter	214
25.34.2.2 ~BitmapToBitmapFilter	215
25.34.3 Member Function Documentation	215
25.34.3.1 GetOutput	215
25.34.3.2 GetOutputAsBitmap	215
25.34.3.3 SetInput	215
25.34.4 Member Data Documentation	215
25.34.4.1 Input	215
25.34.4.2 Output	215
25.35gdcmm::BoxRegion Class Reference	215
25.35.1 Detailed Description	217
25.35.2 Constructor & Destructor Documentation	217
25.35.2.1 BoxRegion	217
25.35.2.2 ~BoxRegion	217
25.35.2.3 BoxRegion	217
25.35.3 Member Function Documentation	217
25.35.3.1 Area	217
25.35.3.2 BoundingBox	217
25.35.3.3 Clone	217
25.35.3.4 ComputeBoundingBox	217
25.35.3.5 Empty	217
25.35.3.6 GetXMax	217
25.35.3.7 GetXMin	217
25.35.3.8 GetYMax	218
25.35.3.9 GetYMin	218
25.35.3.10GetZMax	218
25.35.3.11GetZMin	218
25.35.3.12IsValid	218
25.35.3.13operator=	218
25.35.3.14Print	218
25.35.3.15SetDomain	218
25.36gdcmm::ByteBuffer Class Reference	218
25.36.1 Detailed Description	218
25.36.2 Constructor & Destructor Documentation	219
25.36.2.1 ByteBuffer	219
25.36.3 Member Function Documentation	219

25.36.3.1 Get	219
25.36.3.2 GetStart	219
25.36.3.3 ShiftEnd	219
25.36.3.4 UpdatePosition	219
25.37gdcmm::ByteSwap< T > Class Template Reference	219
25.37.1 Detailed Description	219
25.37.2 Member Function Documentation	220
25.37.2.1 Swap	220
25.37.2.2 SwapFromSwapCodeIntoSystem	220
25.37.2.3 SwapRange	220
25.37.2.4 SwapRangeFromSwapCodeIntoSystem	220
25.37.2.5 SystemIsBigEndian	220
25.37.2.6 SystemIsLittleEndian	220
25.38gdcmm::ByteSwapFilter Class Reference	220
25.38.1 Detailed Description	220
25.38.2 Constructor & Destructor Documentation	221
25.38.2.1 ByteSwapFilter	221
25.38.2.2 ~ByteSwapFilter	221
25.38.3 Member Function Documentation	221
25.38.3.1 ByteSwap	221
25.38.3.2 SetByteSwapTag	221
25.39gdcmm::ByteValue Class Reference	221
25.39.1 Detailed Description	223
25.39.2 Constructor & Destructor Documentation	223
25.39.2.1 ByteValue	223
25.39.2.2 ByteValue	223
25.39.2.3 ~ByteValue	223
25.39.3 Member Function Documentation	223
25.39.3.1 Clear	223
25.39.3.2 Fill	223
25.39.3.3 GetBuffer	223
25.39.3.4 GetLength	224
25.39.3.5 GetPointer	224
25.39.3.6 IsEmpty	224
25.39.3.7 IsPrintable	224
25.39.3.8 operator const std::vector< char > &	224
25.39.3.9 operator=	224

25.39.3.10operator==	224
25.39.3.11operator==	224
25.39.3.12Print	224
25.39.3.13PrintASCII	225
25.39.3.14PrintGroupLength	225
25.39.3.15PrintHex	225
25.39.3.16Read	225
25.39.3.17Read	225
25.39.3.18SetLength	225
25.39.3.19Write	225
25.39.3.20Write	225
25.39.3.21WriteBuffer	225
25.40gdcm::network::CEchoRQ Class Reference	225
25.40.1 Detailed Description	226
25.40.2 Member Function Documentation	227
25.40.2.1 ConstructPDV	227
25.40.3 Member Data Documentation	227
25.40.3.1 AffectedSOPClassUID	227
25.40.3.2 MessageID	227
25.41gdcm::network::CEchoRSP Class Reference	227
25.41.1 Detailed Description	228
25.41.2 Member Function Documentation	228
25.41.2.1 ConstructPDVByDataSet	228
25.42gdcm::network::CFind Class Reference	228
25.42.1 Detailed Description	228
25.43gdcm::network::CFindCancelRQ Class Reference	229
25.43.1 Detailed Description	229
25.43.2 Member Function Documentation	230
25.43.2.1 ConstructPDVByDataSet	230
25.44gdcm::network::CFindRQ Class Reference	230
25.44.1 Detailed Description	231
25.44.2 Member Function Documentation	231
25.44.2.1 ConstructPDV	231
25.45gdcm::network::CFindRSP Class Reference	231
25.45.1 Detailed Description	232
25.45.2 Member Function Documentation	232
25.45.2.1 ConstructPDVByDataSet	232

25.46gdcmm::network::CMoveCancelRq Class Reference	232
25.46.1 Member Function Documentation	233
25.46.1.1 ConstructPDVByDataSet	233
25.47gdcmm::network::CMoveRQ Class Reference	234
25.47.1 Detailed Description	234
25.47.2 Member Function Documentation	235
25.47.2.1 ConstructPDV	235
25.48gdcmm::network::CMoveRSP Class Reference	235
25.48.1 Detailed Description	236
25.48.2 Member Function Documentation	236
25.48.2.1 ConstructPDVByDataSet	236
25.49gdcmm::Codec Class Reference	236
25.49.1 Detailed Description	237
25.50gdcmm::Coder Class Reference	237
25.50.1 Detailed Description	238
25.50.2 Constructor & Destructor Documentation	238
25.50.2.1 ~Coder	238
25.50.3 Member Function Documentation	238
25.50.3.1 CanCode	238
25.50.3.2 Code	239
25.50.3.3 InternalCode	239
25.51gdcmm::CodeString Class Reference	239
25.51.1 Detailed Description	240
25.51.2 Member Typedef Documentation	240
25.51.2.1 const_iterator	240
25.51.2.2 const_reference	240
25.51.2.3 const_reverse_iterator	240
25.51.2.4 difference_type	240
25.51.2.5 iterator	240
25.51.2.6 pointer	240
25.51.2.7 reference	240
25.51.2.8 reverse_iterator	240
25.51.2.9 size_type	241
25.51.2.10value_type	241
25.51.3 Constructor & Destructor Documentation	241
25.51.3.1 CodeString	241
25.51.3.2 CodeString	241

25.51.3.3 CodeString	241
25.51.3.4 CodeString	241
25.51.4 Member Function Documentation	241
25.51.4.1 GetAsString	241
25.51.4.2 IsValid	241
25.51.4.3 Size	241
25.51.4.4 TrimInternal	241
25.51.5 Friends And Related Function Documentation	241
25.51.5.1 operator!=	241
25.51.5.2 operator<<	241
25.51.5.3 operator==	241
25.52gdcmm::Command Class Reference	241
25.52.1 Detailed Description	243
25.52.2 Constructor & Destructor Documentation	243
25.52.2.1 Command	243
25.52.2.2 ~Command	243
25.52.3 Member Function Documentation	243
25.52.3.1 Execute	243
25.52.3.2 Execute	243
25.53gdcmm::CommandDataSet Class Reference	243
25.53.1 Detailed Description	245
25.53.2 Constructor & Destructor Documentation	245
25.53.2.1 CommandDataSet	245
25.53.2.2 ~CommandDataSet	245
25.53.3 Member Function Documentation	245
25.53.3.1 Insert	245
25.53.3.2 Read	245
25.53.3.3 Replace	245
25.53.3.4 Write	245
25.53.4 Friends And Related Function Documentation	245
25.53.4.1 operator<<	245
25.54gdcmm::network::CompositeMessageFactory Class Reference	245
25.54.1 Detailed Description	246
25.54.2 Member Function Documentation	246
25.54.2.1 ConstructCEchoRQ	246
25.54.2.2 ConstructCFindRQ	246
25.54.2.3 ConstructCMoveRQ	246

25.54.2.4 ConstructCStoreRQ	246
25.54.2.5 ConstructCStoreRSP	246
25.55gdcmm::CompositeNetworkFunctions Class Reference	246
25.55.1 Detailed Description	247
25.55.2 Member Typedef Documentation	247
25.55.2.1 KeyValuePairArrayType	247
25.55.2.2 KeyValuePairType	248
25.55.3 Member Function Documentation	248
25.55.3.1 CEcho	248
25.55.3.2 CFind	248
25.55.3.3 CMove	248
25.55.3.4 ConstructQuery	249
25.55.3.5 ConstructQuery	249
25.55.3.6 CStore	249
25.56gdcmm::ConstCharWrapper Class Reference	249
25.56.1 Detailed Description	250
25.56.2 Constructor & Destructor Documentation	250
25.56.2.1 ConstCharWrapper	250
25.56.3 Member Function Documentation	250
25.56.3.1 operator const char *	250
25.57gdcmm::CP246ExplicitDataElement Class Reference	250
25.57.1 Detailed Description	251
25.57.2 Member Function Documentation	251
25.57.2.1 GetLength	251
25.57.2.2 Read	252
25.57.2.3 ReadPreValue	252
25.57.2.4 ReadValue	252
25.57.2.5 ReadWithLength	252
25.58gdcmm::CryptographicMessageSyntax Class Reference	252
25.58.1 Detailed Description	252
25.58.2 Member Enumeration Documentation	253
25.58.2.1 CipherTypes	253
25.58.3 Constructor & Destructor Documentation	253
25.58.3.1 CryptographicMessageSyntax	253
25.58.3.2 ~CryptographicMessageSyntax	253
25.58.4 Member Function Documentation	253
25.58.4.1 Decrypt	253

25.58.4.2 Encrypt	253
25.58.4.3 GetCipherType	253
25.58.4.4 ParseCertificateFile	253
25.58.4.5 ParseKeyFile	253
25.58.4.6 SetCipherType	253
25.59gdcmm::CSAElement Class Reference	253
25.59.1 Detailed Description	255
25.59.2 Member Typedef Documentation	255
25.59.2.1 DataPtr	255
25.59.3 Constructor & Destructor Documentation	255
25.59.3.1 CSAElement	255
25.59.3.2 CSAElement	255
25.59.4 Member Function Documentation	255
25.59.4.1 GetByteValue	256
25.59.4.2 GetKey	256
25.59.4.3 GetName	256
25.59.4.4 GetNoOfItems	256
25.59.4.5 GetSyngoDT	256
25.59.4.6 GetValue	256
25.59.4.7 GetValue	256
25.59.4.8 GetVM	256
25.59.4.9 GetVR	256
25.59.4.10IsEmpty	257
25.59.4.11operator<	257
25.59.4.12operator=	257
25.59.4.13operator==	257
25.59.4.14SetByteValue	257
25.59.4.15SetKey	257
25.59.4.16SetName	257
25.59.4.17SetNoOfItems	257
25.59.4.18SetSyngoDT	257
25.59.4.19SetValue	257
25.59.4.20SetVM	257
25.59.4.21SetVR	257
25.59.5 Friends And Related Function Documentation	257
25.59.5.1 operator<<	257
25.59.6 Member Data Documentation	257

25.59.6.1 DataField	257
25.59.6.2 KeyField	258
25.59.6.3 NameField	258
25.59.6.4 NoOfItemsField	258
25.59.6.5 SyngoDTField	258
25.59.6.6 ValueMultiplicityField	258
25.59.6.7 VRField	258
25.60gdcm::CSAHeader Class Reference	258
25.60.1 Detailed Description	259
25.60.2 Member Enumeration Documentation	260
25.60.2.1 CSAHeaderType	260
25.60.3 Constructor & Destructor Documentation	260
25.60.3.1 CSAHeader	260
25.60.3.2 ~CSAHeader	260
25.60.4 Member Function Documentation	260
25.60.4.1 FindCSAElementByName	260
25.60.4.2 GetCSADataInfo	261
25.60.4.3 GetCSAEEnd	261
25.60.4.4 GetCSAElementByName	261
25.60.4.5 GetCSAImageHeaderInfoTag	261
25.60.4.6 GetCSASeriesHeaderInfoTag	261
25.60.4.7 GetDataSet	261
25.60.4.8 GetFormat	261
25.60.4.9 GetInterfile	261
25.60.4.10 LoadFromDataElement	262
25.60.4.11 Print	262
25.60.4.12 Read	262
25.60.4.13 Write	262
25.60.5 Friends And Related Function Documentation	262
25.60.5.1 operator<<	262
25.61gdcm::CSAHeaderDict Class Reference	262
25.61.1 Detailed Description	263
25.61.2 Member Typedef Documentation	263
25.61.2.1 ConstIterator	263
25.61.2.2 Iterator	263
25.61.2.3 MapCSAHeaderDictEntry	263
25.61.3 Constructor & Destructor Documentation	263

25.61.3.1 CSAHeaderDict	263
25.61.4 Member Function Documentation	263
25.61.4.1 AddCSAHeaderDictEntry	263
25.61.4.2 Begin	263
25.61.4.3 End	263
25.61.4.4 GetCSAHeaderDictEntry	263
25.61.4.5 IsEmpty	264
25.61.4.6 LoadDefault	264
25.61.5 Friends And Related Function Documentation	264
25.61.5.1 Dicts	264
25.61.5.2 operator<<	264
25.62gdcmm::CSAHeaderDictEntry Class Reference	264
25.62.1 Detailed Description	264
25.62.2 Constructor & Destructor Documentation	265
25.62.2.1 CSAHeaderDictEntry	265
25.62.3 Member Function Documentation	265
25.62.3.1 GetDescription	265
25.62.3.2 GetName	265
25.62.3.3 GetVM	265
25.62.3.4 GetVR	265
25.62.3.5 operator<	265
25.62.3.6 SetDescription	265
25.62.3.7 SetName	265
25.62.3.8 SetVM	265
25.62.3.9 SetVR	265
25.62.4 Friends And Related Function Documentation	266
25.62.4.1 operator<<	266
25.63gdcmm::CSAHeaderDictException Class Reference	266
25.64gdcmm::network::CStoreRQ Class Reference	267
25.64.1 Detailed Description	267
25.64.2 Member Function Documentation	268
25.64.2.1 ConstructPDV	268
25.65gdcmm::network::CStoreRSP Class Reference	268
25.65.1 Detailed Description	269
25.65.2 Member Function Documentation	269
25.65.2.1 ConstructPDV	269
25.66gdcmm::Curve Class Reference	269

25.66.1 Detailed Description	271
25.66.2 Constructor & Destructor Documentation	271
25.66.2.1 Curve	271
25.66.2.2 ~Curve	271
25.66.2.3 Curve	271
25.66.3 Member Function Documentation	271
25.66.3.1 Decode	271
25.66.3.2 GetAsPoints	271
25.66.3.3 GetCurveDataDescriptor	271
25.66.3.4 GetDataValueRepresentation	271
25.66.3.5 GetDimensions	271
25.66.3.6 GetGroup	271
25.66.3.7 GetNumberOfCurves	271
25.66.3.8 GetNumberOfPoints	271
25.66.3.9 GetTypeOfData	271
25.66.3.10GetTypeOfDataDescription	271
25.66.3.11IsEmpty	271
25.66.3.12Print	271
25.66.3.13SetCoordinateStartValue	272
25.66.3.14SetCoordinateStepValue	272
25.66.3.15SetCurve	272
25.66.3.16SetCurveDataDescriptor	272
25.66.3.17SetCurveDescription	272
25.66.3.18SetDataValueRepresentation	272
25.66.3.19SetDimensions	272
25.66.3.20SetGroup	272
25.66.3.21SetNumberOfPoints	272
25.66.3.22SetTypeOfData	272
25.66.3.23Update	272
25.67gdcmm::DataElement Class Reference	272
25.67.1 Detailed Description	275
25.67.2 Member Typedef Documentation	275
25.67.2.1 ValuePtr	275
25.67.3 Constructor & Destructor Documentation	275
25.67.3.1 DataElement	275
25.67.3.2 DataElement	276
25.67.4 Member Function Documentation	276

25.67.4.1 Clear	276
25.67.4.2 Empty	276
25.67.4.3 GetByteValue	276
25.67.4.4 GetLength	276
25.67.4.5 GetSequenceOfFragments	276
25.67.4.6 GetSequenceOfItems	276
25.67.4.7 GetSequenceOfItems	277
25.67.4.8 GetTag	277
25.67.4.9 GetTag	277
25.67.4.10 GetValue	277
25.67.4.11 GetValue	277
25.67.4.12 GetValueAsSQ	277
25.67.4.13 GetVL	278
25.67.4.14 GetVL	278
25.67.4.15 GetVR	278
25.67.4.16 IsEmpty	278
25.67.4.17 IsUndefinedLength	278
25.67.4.18 operator<	278
25.67.4.19 operator=	279
25.67.4.20 operator==	279
25.67.4.21 Read	279
25.67.4.22 ReadOrSkip	279
25.67.4.23 ReadPreValue	279
25.67.4.24 ReadValue	279
25.67.4.25 ReadWithLength	279
25.67.4.26 SetByteValue	279
25.67.4.27 SetTag	279
25.67.4.28 SetValue	280
25.67.4.29 SetVL	280
25.67.4.30 SetVLToUndefined	280
25.67.4.31 SetVR	280
25.67.4.32 Write	281
25.67.5 Friends And Related Function Documentation	281
25.67.5.1 operator<<	281
25.67.6 Member Data Documentation	281
25.67.6.1 TagField	281
25.67.6.2 ValueField	281

25.67.6.3 ValueLengthField	281
25.67.6.4 VRField	281
25.68gdcM::DataElementException Class Reference	281
25.69gdcM::DataEvent Class Reference	282
25.69.1 Detailed Description	283
25.69.2 Member Typedef Documentation	283
25.69.2.1 Self	283
25.69.2.2 Superclass	283
25.69.3 Constructor & Destructor Documentation	284
25.69.3.1 DataEvent	284
25.69.3.2 ~DataEvent	284
25.69.3.3 DataEvent	284
25.69.4 Member Function Documentation	284
25.69.4.1 CheckEvent	284
25.69.4.2 GetData	284
25.69.4.3 GetDataLength	284
25.69.4.4 GetEventName	284
25.69.4.5 MakeObject	284
25.69.4.6 SetData	284
25.70gdcM::DataSet Class Reference	284
25.70.1 Detailed Description	286
25.70.2 Member Typedef Documentation	287
25.70.2.1 ConstIterator	287
25.70.2.2 DataElementSet	287
25.70.2.3 Iterator	287
25.70.2.4 SizeType	287
25.70.3 Member Function Documentation	287
25.70.3.1 Begin	287
25.70.3.2 Begin	287
25.70.3.3 Clear	287
25.70.3.4 ComputeDataElement	287
25.70.3.5 ComputeGroupLength	288
25.70.3.6 End	288
25.70.3.7 End	288
25.70.3.8 FindDataElement	288
25.70.3.9 FindDataElement	288
25.70.3.10FindNextDataElement	288

25.70.3.11GetDataElement	288
25.70.3.12GetDataElement	289
25.70.3.13GetDEEnd	289
25.70.3.14GetDES	289
25.70.3.15GetDES	289
25.70.3.16GetLength	289
25.70.3.17GetMediaStorage	289
25.70.3.18GetPrivateCreator	289
25.70.3.19Insert	289
25.70.3.20InsertDataElement	289
25.70.3.21IsEmpty	290
25.70.3.22operator()	290
25.70.3.23operator=	290
25.70.3.24operator[]	290
25.70.3.25Print	290
25.70.3.26Read	290
25.70.3.27ReadNested	290
25.70.3.28ReadSelectedTags	290
25.70.3.29ReadSelectedTagsWithLength	290
25.70.3.30ReadUpToTag	290
25.70.3.31ReadUpToTagWithLength	290
25.70.3.32ReadWithLength	290
25.70.3.33Remove	290
25.70.3.34Replace	290
25.70.3.35ReplaceEmpty	291
25.70.3.36Size	291
25.70.3.37Write	291
25.70.4 Friends And Related Function Documentation	291
25.70.4.1 CSAHeader	291
25.70.4.2 operator<<	291
25.71gdcm::DataSetEvent Class Reference	291
25.71.1 Detailed Description	292
25.71.2 Member Typedef Documentation	292
25.71.2.1 Self	292
25.71.2.2 Superclass	292
25.71.3 Constructor & Destructor Documentation	293
25.71.3.1 DataSetEvent	293

25.71.3.2 ~DataSetEvent	293
25.71.3.3 DataSetEvent	293
25.71.4 Member Function Documentation	293
25.71.4.1 CheckEvent	293
25.71.4.2 GetDataSet	293
25.71.4.3 GetEventName	293
25.71.4.4 MakeObject	293
25.72gdcmm::DataSetHelper Class Reference	293
25.72.1 Detailed Description	293
25.72.2 Member Function Documentation	293
25.72.2.1 ComputeVR	293
25.73gdcmm::Decoder Class Reference	294
25.73.1 Detailed Description	294
25.73.2 Constructor & Destructor Documentation	294
25.73.2.1 ~Decoder	294
25.73.3 Member Function Documentation	295
25.73.3.1 CanDecode	295
25.73.3.2 Decode	295
25.73.3.3 DecodeByStreams	295
25.74gdcmm::DefinedTerms Class Reference	295
25.74.1 Detailed Description	295
25.74.2 Constructor & Destructor Documentation	296
25.74.2.1 DefinedTerms	296
25.75gdcmm::Defs Class Reference	296
25.75.1 Detailed Description	297
25.75.2 Constructor & Destructor Documentation	297
25.75.2.1 Defs	297
25.75.2.2 ~Defs	297
25.75.3 Member Function Documentation	297
25.75.3.1 GetIODFromFile	297
25.75.3.2 GetIODNameFromMediaStorage	297
25.75.3.3 GetIODs	297
25.75.3.4 GetIODs	297
25.75.3.5 GetMacros	297
25.75.3.6 GetMacros	297
25.75.3.7 GetModules	297
25.75.3.8 GetModules	297

25.75.3.9 GetTypeFromTag	297
25.75.3.10IsEmpty	297
25.75.3.11LoadDefaults	297
25.75.3.12LoadFromFile	297
25.75.3.13Verify	298
25.75.3.14Verify	298
25.75.4 Friends And Related Function Documentation	298
25.75.4.1 Global	298
25.76gdcm::DeltaEncodingCodec Class Reference	298
25.76.1 Detailed Description	299
25.76.2 Constructor & Destructor Documentation	299
25.76.2.1 DeltaEncodingCodec	299
25.76.2.2 ~DeltaEncodingCodec	299
25.76.3 Member Function Documentation	299
25.76.3.1 CanDecode	299
25.76.3.2 Decode	299
25.76.3.3 Decode	300
25.77gdcm::DICOMDIR Class Reference	300
25.77.1 Detailed Description	300
25.77.2 Constructor & Destructor Documentation	300
25.77.2.1 DICOMDIR	300
25.77.2.2 DICOMDIR	300
25.78gdcm::DICOMDIRGenerator Class Reference	300
25.78.1 Detailed Description	301
25.78.2 Member Typedef Documentation	302
25.78.2.1 FilenamesType	302
25.78.2.2 FilenameType	302
25.78.3 Constructor & Destructor Documentation	302
25.78.3.1 DICOMDIRGenerator	302
25.78.3.2 ~DICOMDIRGenerator	302
25.78.4 Member Function Documentation	302
25.78.4.1 AddImageDirectoryRecord	302
25.78.4.2 AddPatientDirectoryRecord	302
25.78.4.3 AddSeriesDirectoryRecord	302
25.78.4.4 AddStudyDirectoryRecord	302
25.78.4.5 Generate	302
25.78.4.6 GetFile	302

25.78.4.7 GetScanner	302
25.78.4.8 SetDescriptor	302
25.78.4.9 SetFile	302
25.78.4.10SetFilenames	302
25.78.4.11SetRootDirectory	302
25.79gdcmm::Dict Class Reference	303
25.79.1 Detailed Description	303
25.79.2 Member Typedef Documentation	304
25.79.2.1 ConstIterator	304
25.79.2.2 Iterator	304
25.79.2.3 MapDictEntry	304
25.79.3 Constructor & Destructor Documentation	304
25.79.3.1 Dict	304
25.79.4 Member Function Documentation	304
25.79.4.1 AddDictEntry	304
25.79.4.2 Begin	304
25.79.4.3 End	304
25.79.4.4 GetDictEntry	304
25.79.4.5 GetDictEntryByKeyword	304
25.79.4.6 GetDictEntryByName	304
25.79.4.7 GetKeywordFromTag	305
25.79.4.8 IsEmpty	305
25.79.4.9 LoadDefault	305
25.79.5 Friends And Related Function Documentation	305
25.79.5.1 Dicts	305
25.79.5.2 operator<<	305
25.80gdcmm::DictConverter Class Reference	305
25.80.1 Detailed Description	306
25.80.2 Member Enumeration Documentation	306
25.80.2.1 OutputTypes	306
25.80.3 Constructor & Destructor Documentation	306
25.80.3.1 DictConverter	306
25.80.3.2 ~DictConverter	306
25.80.4 Member Function Documentation	306
25.80.4.1 AddGroupLength	306
25.80.4.2 Convert	306
25.80.4.3 ConvertToCXX	306

25.80.4.4 ConvertToXML	307
25.80.4.5 GetDictName	307
25.80.4.6 GetInputFilename	307
25.80.4.7 GetOutputFilename	307
25.80.4.8 GetOutputType	307
25.80.4.9 Readuint16	307
25.80.4.10ReadVM	307
25.80.4.11ReadVR	307
25.80.4.12SetDictName	307
25.80.4.13SetInputFileName	307
25.80.4.14SetOutputFileName	307
25.80.4.15SetOutputType	307
25.80.4.16WriteFooter	307
25.80.4.17WriteHeader	307
25.81gdcmm::DictEntry Class Reference	307
25.81.1 Detailed Description	308
25.81.2 Constructor & Destructor Documentation	308
25.81.2.1 DictEntry	308
25.81.3 Member Function Documentation	308
25.81.3.1 GetKeyword	308
25.81.3.2 GetName	308
25.81.3.3 GetRetired	309
25.81.3.4 GetVM	309
25.81.3.5 GetVR	309
25.81.3.6 IsUnique	309
25.81.3.7 SetElementXX	309
25.81.3.8 SetGroupXX	309
25.81.3.9 SetKeyword	309
25.81.3.10SetName	309
25.81.3.11SetRetired	309
25.81.3.12SetVM	309
25.81.3.13SetVR	309
25.81.4 Friends And Related Function Documentation	310
25.81.4.1 operator<<	310
25.82gdcmm::DictPrinter Class Reference	310
25.82.1 Detailed Description	311
25.82.2 Constructor & Destructor Documentation	311

25.82.2.1 DictPrinter	311
25.82.2.2 ~DictPrinter	312
25.82.3 Member Function Documentation	312
25.82.3.1 Print	312
25.82.3.2 PrintDataElement2	312
25.82.3.3 PrintDataSet2	312
25.83gdcmm::Dicts Class Reference	312
25.83.1 Detailed Description	313
25.83.2 Member Enumeration Documentation	313
25.83.2.1 ConstructorType	313
25.83.3 Constructor & Destructor Documentation	313
25.83.3.1 Dicts	313
25.83.3.2 ~Dicts	313
25.83.4 Member Function Documentation	313
25.83.4.1 GetConstructorString	313
25.83.4.2 GetCSAHeaderDict	313
25.83.4.3 GetDictEntry	313
25.83.4.4 GetDictEntry	314
25.83.4.5 GetPrivateDict	314
25.83.4.6 GetPrivateDict	314
25.83.4.7 GetPublicDict	314
25.83.4.8 IsEmpty	314
25.83.4.9 LoadDefaults	314
25.83.5 Friends And Related Function Documentation	314
25.83.5.1 Global	314
25.83.5.2 operator<<	314
25.84gdcmm::network::DIMSE Class Reference	314
25.84.1 Detailed Description	315
25.84.2 Member Enumeration Documentation	315
25.84.2.1 CommandTypes	315
25.85gdcmm::DirectionCosines Class Reference	316
25.85.1 Detailed Description	317
25.85.2 Constructor & Destructor Documentation	317
25.85.2.1 DirectionCosines	317
25.85.2.2 DirectionCosines	317
25.85.2.3 ~DirectionCosines	317
25.85.3 Member Function Documentation	317

25.85.3.1 ComputeDistAlongNormal	317
25.85.3.2 Cross	317
25.85.3.3 CrossDot	317
25.85.3.4 Dot	317
25.85.3.5 IsValid	317
25.85.3.6 Normalize	317
25.85.3.7 operator const double *	317
25.85.3.8 Print	318
25.85.3.9 SetFromString	318
25.86gdcmm::Directory Class Reference	318
25.86.1 Detailed Description	319
25.86.2 Member Typedef Documentation	319
25.86.2.1 FilenamesType	319
25.86.2.2 FilenameType	319
25.86.3 Constructor & Destructor Documentation	319
25.86.3.1 Directory	319
25.86.3.2 ~Directory	319
25.86.4 Member Function Documentation	319
25.86.4.1 Explore	319
25.86.4.2 GetDirectories	319
25.86.4.3 GetFilenames	320
25.86.4.4 GetToplevel	320
25.86.4.5 Load	320
25.86.4.6 Print	320
25.86.5 Friends And Related Function Documentation	320
25.86.5.1 operator<<	320
25.87gdcmm::DirectoryHelper Class Reference	320
25.87.1 Detailed Description	321
25.87.2 Member Function Documentation	321
25.87.2.1 GetCTImageSeriesUIDs	321
25.87.2.2 GetFilenamesFromSeriesUIDs	321
25.87.2.3 GetFrameOfReference	321
25.87.2.4 GetMRImageSeriesUIDs	321
25.87.2.5 GetRTStructSeriesUIDs	322
25.87.2.6 GetSeriesUIDsBySOPClassUID	322
25.87.2.7 GetSOPClassUID	322
25.87.2.8 GetStringValueFromTag	322

25.87.2.9 LoadImageFromFiles	322
25.87.2.10 RetrieveSOPInstanceUIDFromIndex	322
25.87.2.11 RetrieveSOPInstanceUIDFromZPosition	322
25.88gdcmm::DummyValueGenerator Class Reference	322
25.88.1 Detailed Description	322
25.88.2 Member Function Documentation	322
25.88.2.1 Generate	323
25.89gdcmm::Dumper Class Reference	323
25.89.1 Detailed Description	324
25.89.2 Constructor & Destructor Documentation	324
25.89.2.1 Dumper	324
25.89.2.2 ~Dumper	324
25.90gdcmm::Element< TVR, TVM > Class Template Reference	325
25.90.1 Detailed Description	327
25.90.2 Member Typedef Documentation	327
25.90.2.1 Type	327
25.90.3 Member Function Documentation	327
25.90.3.1 GetAsDataElement	327
25.90.3.2 GetLength	327
25.90.3.3 GetValue	327
25.90.3.4 GetValue	327
25.90.3.5 GetValues	327
25.90.3.6 GetVM	327
25.90.3.7 GetVR	327
25.90.3.8 operator[]	327
25.90.3.9 Print	327
25.90.3.10 Read	328
25.90.3.11 Set	328
25.90.3.12 SetFromDataElement	328
25.90.3.13 SetNoSwap	328
25.90.3.14 SetValue	328
25.90.3.15 Write	328
25.90.4 Member Data Documentation	328
25.90.4.1 Internal	328
25.91gdcmm::Element< TVR, VM::VM1_2 > Class Template Reference	328
25.91.1 Member Typedef Documentation	329
25.91.1.1 Parent	329

25.91.2 Member Function Documentation	329
25.91.2.1 SetLength	329
25.92gdcmm::Element< TVR, VM::VM1_n > Class Template Reference	329
25.92.1 Member Typedef Documentation	331
25.92.1.1 Type	331
25.92.2 Constructor & Destructor Documentation	331
25.92.2.1 Element	331
25.92.2.2 ~Element	331
25.92.2.3 Element	331
25.92.3 Member Function Documentation	331
25.92.3.1 GetAsDataElement	331
25.92.3.2 GetLength	331
25.92.3.3 GetValue	331
25.92.3.4 GetValue	331
25.92.3.5 GetVM	331
25.92.3.6 GetVR	331
25.92.3.7 operator=	331
25.92.3.8 operator[]	331
25.92.3.9 Print	331
25.92.3.10Read	331
25.92.3.11Set	331
25.92.3.12SetArray	332
25.92.3.13SetFromDataElement	332
25.92.3.14SetLength	332
25.92.3.15SetNoSwap	332
25.92.3.16SetValue	332
25.92.3.17Write	332
25.92.3.18WriteASCII	332
25.93gdcmm::Element< TVR, VM::VM2_n > Class Template Reference	332
25.93.1 Member Typedef Documentation	334
25.93.1.1 Parent	334
25.93.2 Member Function Documentation	334
25.93.2.1 SetLength	334
25.94gdcmm::Element< TVR, VM::VM2_n > Class Template Reference	334
25.94.1 Member Typedef Documentation	335
25.94.1.1 Parent	335
25.94.2 Member Function Documentation	335

25.94.2.1 SetLength	335
25.95gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference	335
25.95.1 Member Typedef Documentation	337
25.95.1.1 Parent	337
25.95.2 Member Function Documentation	337
25.95.2.1 SetLength	337
25.96gdcmm::Element< TVR, VM::VM3_n > Class Template Reference	337
25.96.1 Member Typedef Documentation	338
25.96.1.1 Parent	338
25.96.2 Member Function Documentation	338
25.96.2.1 SetLength	338
25.97gdcmm::Element< VR::AS, VM::VM5 > Class Template Reference	338
25.97.1 Member Function Documentation	339
25.97.1.1 GetLength	339
25.97.1.2 Print	339
25.97.2 Member Data Documentation	339
25.97.2.1 Internal	339
25.98gdcmm::Element< VR::OB, VM::VM1 > Class Template Reference	339
25.99gdcmm::Element< VR::OW, VM::VM1 > Class Template Reference	340
25.100gdcmm::ElementDisableCombinations< TVR, TVM > Class Template Reference	342
25.100. Detailed Description	342
25.101gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference	343
25.102gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference	343
25.103gdcmm::EncapsulatedDocument Class Reference	343
25.103. Detailed Description	343
25.103.2 Constructor & Destructor Documentation	343
25.103.2.1 EncapsulatedDocument	343
25.104gdcmm::EncodingImplementation< T > Class Template Reference	344
25.104. Detailed Description	344
25.105gdcmm::EncodingImplementation< VR::VRASCII > Class Template Reference	344
25.105. Member Function Documentation	344
25.105.1.1 Read	344
25.105.1.2 ReadComputeLength	345
25.105.1.3 ReadNoSwap	345
25.105.1.4 Write	345
25.105.1.5 Write	345
25.105.1.6 Write	345

25.106	dcm::EncodingImplementation< VR::VRBINARY > Class Template Reference	345
25.106.1	Member Function Documentation	345
25.106.1.1	Read	345
25.106.1.2	ReadComputeLength	346
25.106.1.3	ReadNoSwap	346
25.106.1.4	Write	346
25.107	dcm::EndEvent Class Reference	346
25.108	dcm::EnumeratedValues Class Reference	347
25.108.1	Detailed Description	347
25.108.2	Constructor & Destructor Documentation	348
25.108.2.1	EnumeratedValues	348
25.109	dcm::Event Class Reference	348
25.109.1	Detailed Description	349
25.109.2	Constructor & Destructor Documentation	349
25.109.2.1	Event	349
25.109.2.2	~Event	349
25.109.2.3	~Event	349
25.109.3	Member Function Documentation	349
25.109.3.1	CheckEvent	349
25.109.3.2	GetEventName	349
25.109.3.3	MakeObject	349
25.109.3.4	Print	349
25.110	dcm::Exception Class Reference	350
25.110.1	Detailed Description	351
25.110.2	Constructor & Destructor Documentation	351
25.110.2.1	Exception	351
25.110.2.2	~Exception	351
25.110.3	Member Function Documentation	351
25.110.3.1	GetDescription	351
25.110.3.2	what	351
25.111	dcm::ExitEvent Class Reference	351
25.112	dcm::ExplicitDataElement Class Reference	353
25.112.1	Detailed Description	354
25.112.2	Member Function Documentation	354
25.112.2.1	GetLength	354
25.112.2.2	Read	354
25.112.2.3	ReadPreValue	354

25.112.2.4	ReadValue	354
25.112.2.5	ReadWithLength	354
25.112.2.6	Write	354
25.113	gdcm::ExplicitImplicitDataElement Class Reference	354
25.113.1	Detailed Description	356
25.113.2	Member Function Documentation	356
25.113.2.1	GetLength	356
25.113.2.2	Read	356
25.113.2.3	ReadPreValue	356
25.113.2.4	ReadValue	356
25.113.2.5	ReadWithLength	356
25.114	gdcm::Fiducials Class Reference	356
25.114.1	Detailed Description	356
25.114.2	Constructor & Destructor Documentation	357
25.114.2.1	Fiducials	357
25.115	gdcm::File Class Reference	357
25.115.1	Detailed Description	358
25.115.2	Constructor & Destructor Documentation	359
25.115.2.1	File	359
25.115.2.2	~File	359
25.115.3	Member Function Documentation	359
25.115.3.1	GetDataSet	359
25.115.3.2	GetDataSet	359
25.115.3.3	GetHeader	359
25.115.3.4	GetHeader	360
25.115.3.5	Read	360
25.115.3.6	SetDataSet	360
25.115.3.7	SetHeader	360
25.115.3.8	Write	360
25.115.4	Friends And Related Function Documentation	360
25.115.4.1	operator<<	360
25.116	gdcm::FileAnonymizer Class Reference	360
25.116.1	Detailed Description	362
25.116.2	Constructor & Destructor Documentation	362
25.116.2.1	FileAnonymizer	362
25.116.2.2	~FileAnonymizer	362
25.116.3	Member Function Documentation	362

25.116.3.1Empty	362
25.116.3.2Remove	362
25.116.3.3Replace	362
25.116.3.4Replace	363
25.116.3.5SetInputFileName	363
25.116.3.6SetOutputFileName	363
25.116.3.7Write	363
25.117dcm::FileDerivation Class Reference	363
25.117.1Detailed Description	364
25.117.2Constructor & Destructor Documentation	364
25.117.2.1FileDerivation	364
25.117.2.2~FileDerivation	364
25.117.3Member Function Documentation	364
25.117.3.1AddDerivationDescription	364
25.117.3.2AddPurposeOfReferenceCodeSequence	364
25.117.3.3AddReference	364
25.117.3.4AddSourceImageSequence	364
25.117.3.5Derive	364
25.117.3.6GetFile	365
25.117.3.7GetFile	365
25.117.3.8SetDerivationCodeSequenceCodeValue	365
25.117.3.9SetDerivationDescription	365
25.117.3.10SetFile	365
25.117.3.11SetPurposeOfReferenceCodeSequenceCodeValue	365
25.118dcm::FileExplicitFilter Class Reference	365
25.118.1Detailed Description	366
25.118.2Constructor & Destructor Documentation	366
25.118.2.1FileExplicitFilter	366
25.118.2.2~FileExplicitFilter	366
25.118.3Member Function Documentation	366
25.118.3.1Change	367
25.118.3.2ChangeFMI	367
25.118.3.3GetFile	367
25.118.3.4ProcessDataSet	367
25.118.3.5SetChangePrivateTags	367
25.118.3.6SetFile	367
25.118.3.7SetRecomputeItemLength	367

25.118.3.8SetRecomputeSequenceLength	367
25.118.3.9SetUseVRUN	367
25.119.0dcm::FileMetaInformation Class Reference	367
25.119.1Detailed Description	370
25.119.2Constructor & Destructor Documentation	370
25.119.2.1FileMetaInformation	370
25.119.2.2~FileMetaInformation	370
25.119.2.3FileMetaInformation	370
25.119.3Member Function Documentation	370
25.119.3.1AppendImplementationClassUID	370
25.119.3.2ComputeDataSetMediaStorageSOPClass	370
25.119.3.3ComputeDataSetTransferSyntax	370
25.119.3.4Default	370
25.119.3.5FillFromDataSet	370
25.119.3.6GetDataSetTransferSyntax	370
25.119.3.7GetFileMetaInformationVersion	371
25.119.3.8GetFullLength	371
25.119.3.9GetGDCMImplementationClassUID	371
25.119.3.10GetGDCMImplementationVersionName	371
25.119.3.11GetGDCMSourceApplicationEntityTitle	371
25.119.3.12GetImplementationClassUID	371
25.119.3.13GetImplementationVersionName	371
25.119.3.14GetMediaStorage	371
25.119.3.15GetMetaInformationTS	371
25.119.3.16GetPreamble	371
25.119.3.17GetPreamble	371
25.119.3.18GetSourceApplicationEntityTitle	371
25.119.3.19Insert	371
25.119.3.20Valid	371
25.119.3.21Read	371
25.119.3.22ReadCompat	371
25.119.3.23ReadCompatInternal	371
25.119.3.24Replace	372
25.119.3.25SetDataSetTransferSyntax	372
25.119.3.26SetImplementationClassUID	372
25.119.3.27SetImplementationVersionName	372
25.119.3.28SetPreamble	372

25.119.3.2	SetSourceApplicationEntityTitle	372
25.119.3.3	Write	372
25.119.4	Friends And Related Function Documentation	372
25.119.4.1	operator<<	372
25.119.5	Member Data Documentation	372
25.119.5.1	DataSetMS	372
25.119.5.2	DataSetTS	372
25.119.5.3	MetaInformationTS	373
25.120	dcm::Filename Class Reference	373
25.120.1	Detailed Description	373
25.120.2	Constructor & Destructor Documentation	374
25.120.2.1	Filename	374
25.120.3	Member Function Documentation	374
25.120.3.1	EndWith	374
25.120.3.2	GetExtension	374
25.120.3.3	GetFileName	374
25.120.3.4	GetName	374
25.120.3.5	GetPath	374
25.120.3.6	IsEmpty	374
25.120.3.7	IsIdentical	374
25.120.3.8	Join	374
25.120.3.9	operator const char *	374
25.120.3.10	ToUnixSlashes	374
25.120.3.11	ToWindowsSlashes	375
25.121	dcm::FilenameGenerator Class Reference	375
25.121.1	Detailed Description	375
25.121.2	Member Typedef Documentation	376
25.121.2.1	FileNamesType	376
25.121.2.2	FilenameType	376
25.121.2.3	SizeType	376
25.121.3	Constructor & Destructor Documentation	376
25.121.3.1	FilenameGenerator	376
25.121.3.2	~FilenameGenerator	376
25.121.4	Member Function Documentation	376
25.121.4.1	Generate	376
25.121.4.2	GetFilename	376
25.121.4.3	GetFileNames	376

25.121.4.4	GetNumberOfFileNames	376
25.121.4.5	GetPattern	377
25.121.4.6	GetPrefix	377
25.121.4.7	SetNumberOfFileNames	377
25.121.4.8	SetPattern	377
25.121.4.9	SetPrefix	377
25.122	dcm::FileSet Class Reference	377
25.122.1	Detailed Description	378
25.122.2	Member Typedef Documentation	378
25.122.2.1	FileType	378
25.122.2.2	FileType	378
25.122.3	Constructor & Destructor Documentation	378
25.122.3.1	FileSet	378
25.122.4	Member Function Documentation	378
25.122.4.1	AddFile	378
25.122.4.2	AddFile	378
25.122.4.3	GetFiles	378
25.122.4.4	SetFiles	378
25.122.5	Friends And Related Function Documentation	378
25.122.5.1	operator<<	378
25.123	dcm::FileWithName Class Reference	378
25.123.1	Detailed Description	380
25.123.2	Constructor & Destructor Documentation	380
25.123.2.1	FileWithName	380
25.123.3	Member Data Documentation	380
25.123.3.1	filename	380
25.124	dcm::FindPatientRootQuery Class Reference	380
25.124.1	Detailed Description	381
25.124.2	Constructor & Destructor Documentation	381
25.124.2.1	FindPatientRootQuery	381
25.124.3	Member Function Documentation	381
25.124.3.1	GetAbstractSyntaxUID	381
25.124.3.2	GetTagListByLevel	382
25.124.3.3	InitializeDataSet	382
25.124.3.4	ValidateQuery	382
25.124.4	Friends And Related Function Documentation	382
25.124.4.1	QueryFactory	382

25.125.5dcm::FindStudyRootQuery Class Reference	382
25.125.1Detailed Description	384
25.125.2Constructor & Destructor Documentation	384
25.125.2.1FindStudyRootQuery	384
25.125.3Member Function Documentation	384
25.125.3.1GetAbstractSyntaxUID	384
25.125.3.2GetTagListByLevel	384
25.125.3.3InitializeDataSet	384
25.125.3.4ValidateQuery	384
25.125.4Friends And Related Function Documentation	384
25.125.4.1QueryFactory	384
25.126.6dcm::Fragment Class Reference	384
25.126.1Detailed Description	386
25.126.2Constructor & Destructor Documentation	386
25.126.2.1Fragment	386
25.126.3Member Function Documentation	386
25.126.3.1GetLength	386
25.126.3.2Read	386
25.126.3.3ReadBacktrack	386
25.126.3.4ReadPreValue	386
25.126.3.5ReadValue	386
25.126.3.6Write	387
25.126.4Friends And Related Function Documentation	387
25.126.4.1operator<<	387
25.127.7dcm::Global Class Reference	387
25.127.1Detailed Description	388
25.127.2Constructor & Destructor Documentation	388
25.127.2.1Global	388
25.127.2.2~Global	388
25.127.3Member Function Documentation	388
25.127.3.1Append	388
25.127.3.2GetDefs	388
25.127.3.3GetDicts	388
25.127.3.4GetDicts	388
25.127.3.5GetInstance	388
25.127.3.6LoadResourcesFiles	389
25.127.3.7Locate	389

25.127.3.8Prepend	389
25.127.4Friends And Related Function Documentation	389
25.127.4.1operator<<	389
25.128gdcmm::GroupDict Class Reference	389
25.128.1Detailed Description	390
25.128.2Member Typedef Documentation	390
25.128.2.1GroupStringVector	390
25.128.3Constructor & Destructor Documentation	390
25.128.3.1GroupDict	390
25.128.3.2~GroupDict	390
25.128.4Member Function Documentation	390
25.128.4.1Add	390
25.128.4.2GetAbbreviation	390
25.128.4.3GetName	390
25.128.4.4Insert	391
25.128.4.5Size	391
25.128.5Friends And Related Function Documentation	391
25.128.5.1operator<<	391
25.129gdcmm::IconImageFilter Class Reference	391
25.129.1Detailed Description	391
25.129.2Constructor & Destructor Documentation	392
25.129.2.1IconImageFilter	392
25.129.2.2~IconImageFilter	392
25.129.3Member Function Documentation	392
25.129.3.1Extract	392
25.129.3.2ExtractIconImages	392
25.129.3.3ExtractVeprolIconImages	392
25.129.3.4GetFile	392
25.129.3.5GetFile	392
25.129.3.6GetIconImage	393
25.129.3.7GetNumberOfIconImages	393
25.129.3.8SetFile	393
25.130gdcmm::IconImageGenerator Class Reference	393
25.130.1Detailed Description	394
25.130.2Constructor & Destructor Documentation	394
25.130.2.1IconImageGenerator	394
25.130.2.2~IconImageGenerator	394

25.130.3	Member Function Documentation	394
25.130.3.1	AutoPixelMinMax	394
25.130.3.2	ConvertRGBToPaletteColor	394
25.130.3.3	Generate	394
25.130.3.4	GetIconImage	395
25.130.3.5	GetPixmap	395
25.130.3.6	GetPixmap	395
25.130.3.7	SetOutputDimensions	395
25.130.3.8	SetOutsideValuePixel	395
25.130.3.9	SetPixelMinMax	395
25.130.3.10	SetPixmap	395
25.130	gdcm::ignore_char Struct Reference	395
25.131.1	Constructor & Destructor Documentation	396
25.131.1.1	ignore_char	396
25.131.2	Member Data Documentation	396
25.131.2.1	m_char	396
25.132	gdcm::Image Class Reference	396
25.132.1	Detailed Description	397
25.132.2	Constructor & Destructor Documentation	398
25.132.2.1	Image	398
25.132.2.2	~Image	398
25.132.3	Member Function Documentation	398
25.132.3.1	GetDirectionCosines	398
25.132.3.2	GetDirectionCosines	398
25.132.3.3	GetIntercept	398
25.132.3.4	GetOrigin	398
25.132.3.5	GetOrigin	399
25.132.3.6	GetSlope	399
25.132.3.7	GetSpacing	399
25.132.3.8	GetSpacing	399
25.132.3.9	Print	399
25.132.3.10	SetDirectionCosines	399
25.132.3.11	SetDirectionCosines	399
25.132.3.12	SetDirectionCosines	399
25.132.3.13	SetIntercept	399
25.132.3.14	SetOrigin	399
25.132.3.15	SetOrigin	399

25.132.3.1	SetOrigin	399
25.132.3.1	SetSlope	399
25.132.3.1	SetSpacing	399
25.132.3.1	SetSpacing	400
25.133	dcm::ImageApplyLookupTable Class Reference	400
25.133.1	Detailed Description	402
25.133.2	Constructor & Destructor Documentation	402
25.133.2.1	ImageApplyLookupTable	402
25.133.2.2	~ImageApplyLookupTable	402
25.133.3	Member Function Documentation	402
25.133.3.1	Apply	402
25.134	dcm::ImageChangePhotometricInterpretation Class Reference	402
25.134.1	Detailed Description	405
25.134.2	Constructor & Destructor Documentation	405
25.134.2.1	ImageChangePhotometricInterpretation	405
25.134.2.2	~ImageChangePhotometricInterpretation	405
25.134.3	Member Function Documentation	405
25.134.3.1	Change	405
25.134.3.2	ChangeMonochrome	405
25.134.3.3	GetPhotometricInterpretation	405
25.134.3.4	RGB2YBR	405
25.134.3.5	SetPhotometricInterpretation	405
25.134.3.6	YBR2RGB	406
25.135	dcm::ImageChangePlanarConfiguration Class Reference	406
25.135.1	Detailed Description	408
25.135.2	Constructor & Destructor Documentation	408
25.135.2.1	ImageChangePlanarConfiguration	408
25.135.2.2	~ImageChangePlanarConfiguration	408
25.135.3	Member Function Documentation	408
25.135.3.1	Change	408
25.135.3.2	GetPlanarConfiguration	408
25.135.3.3	RGBPixelsToRGBPlanes	408
25.135.3.4	RGBPlanesToRGBPixels	408
25.135.3.5	SetPlanarConfiguration	408
25.136	dcm::ImageChangeTransferSyntax Class Reference	409
25.136.1	Detailed Description	411
25.136.2	Constructor & Destructor Documentation	411

25.136.2.1ImageChangeTransferSyntax	411
25.136.2.2~ImageChangeTransferSyntax	411
25.136.3Member Function Documentation	411
25.136.3.1Change	411
25.136.3.2GetTransferSyntax	411
25.136.3.3SetCompressIconImage	412
25.136.3.4SetForce	412
25.136.3.5SetTransferSyntax	412
25.136.3.6SetUserCodec	412
25.136.3.7TryJPEG2000Codec	412
25.136.3.8TryJPEGCodec	412
25.136.3.9TryJPEGLSCodec	412
25.136.3.10TryRAWCodec	412
25.136.3.11TryRLECodec	412
25.137gdcmm::ImageCodec Class Reference	413
25.137.1Detailed Description	415
25.137.2Member Typedef Documentation	415
25.137.2.1LUTPtr	415
25.137.3Constructor & Destructor Documentation	415
25.137.3.1ImageCodec	415
25.137.3.2~ImageCodec	415
25.137.4Member Function Documentation	415
25.137.4.1CanCode	415
25.137.4.2CanDecode	415
25.137.4.3Decode	415
25.137.4.4DecodeByStreams	416
25.137.4.5DoByteSwap	416
25.137.4.6DoInvertMonochrome	416
25.137.4.7DoOverlayCleanup	416
25.137.4.8DoPaddedCompositePixelCode	416
25.137.4.9DoPlanarConfiguration	416
25.137.4.10DoSimpleCopy	416
25.137.4.11DoYBR	416
25.137.4.12GetDimensions	416
25.137.4.13GetHeaderInfo	416
25.137.4.14GetLossyFlag	416
25.137.4.15GetLUT	416

25.137.4.16	GetNeedByteSwap	416
25.137.4.16	GetNumberOfDimensions	416
25.137.4.16	GetPhotometricInterpretation	416
25.137.4.16	GetPixelFormat	416
25.137.4.20	GetPixelFormat	416
25.137.4.20	GetPlanarConfiguration	416
25.137.4.23	Lossy	416
25.137.4.23	Valid	417
25.137.4.23	SetDimensions	417
25.137.4.25	SetDimensions	417
25.137.4.26	SetLossyFlag	417
25.137.4.27	SetLUT	417
25.137.4.28	SetNeedByteSwap	417
25.137.4.29	SetNeedOverlayCleanup	417
25.137.4.30	SetNumberOfDimensions	417
25.137.4.33	SetPhotometricInterpretation	417
25.137.4.33	SetPixelFormat	417
25.137.4.33	SetPlanarConfiguration	417
25.137.5	Friends And Related Function Documentation	417
25.137.5.1	ImageChangePhotometricInterpretation	417
25.137.6	Member Data Documentation	417
25.137.6.1	Dimensions	418
25.137.6.2	LossyFlag	418
25.137.6.3	LUT	418
25.137.6.4	NeedByteSwap	418
25.137.6.5	NeedOverlayCleanup	418
25.137.6.6	NumberOfDimensions	418
25.137.6.7	PF	418
25.137.6.8	PI	418
25.137.6.9	PlanarConfiguration	418
25.137.6.10	RequestPaddedCompositePixelCode	418
25.137.6.11	RequestPlanarConfiguration	418
25.138	dcm::ImageConverter Class Reference	418
25.138.1	Detailed Description	418
25.138.2	Constructor & Destructor Documentation	419
25.138.2.1	ImageConverter	419
25.138.2.2	~ImageConverter	419

25.138.3	Member Function Documentation	419
25.138.3.1	Convert	419
25.138.3.2	GetOutput	419
25.138.3.3	SetInput	419
25.139	dcm::ImageFragmentSplitter Class Reference	419
25.139.1	Detailed Description	421
25.139.2	Constructor & Destructor Documentation	421
25.139.2.1	ImageFragmentSplitter	421
25.139.2.2	~ImageFragmentSplitter	421
25.139.3	Member Function Documentation	421
25.139.3.1	GetFragmentSizeMax	421
25.139.3.2	SetForce	421
25.139.3.3	SetFragmentSizeMax	421
25.139.3.4	Split	421
25.140	dcm::ImageHelper Class Reference	421
25.140.1	Detailed Description	422
25.140.2	Member Function Documentation	422
25.140.2.1	ComputeSpacingFromImagePositionPatient	422
25.140.2.2	GetDimensionsValue	423
25.140.2.3	GetDirectionCosinesFromDataSet	423
25.140.2.4	GetDirectionCosinesValue	423
25.140.2.5	GetForcePixelSpacing	423
25.140.2.6	GetForceRescaleInterceptSlope	423
25.140.2.7	GetLUT	423
25.140.2.8	GetOriginValue	423
25.140.2.9	GetPhotometricInterpretationValue	423
25.140.2.10	GetPixelFormatValue	423
25.140.2.11	GetPlanarConfigurationValue	423
25.140.2.12	GetPointerFromElement	423
25.140.2.13	GetRescaleInterceptSlopeValue	423
25.140.2.14	GetSpacingTagFromMediaStorage	424
25.140.2.15	GetSpacingValue	424
25.140.2.16	GetZSpacingTagFromMediaStorage	424
25.140.2.17	SetDimensionsValue	424
25.140.2.18	SetDirectionCosinesValue	424
25.140.2.19	SetForcePixelSpacing	424
25.140.2.20	SetForceRescaleInterceptSlope	424

25.140.2.2	SetOriginValue	424
25.140.2.2	SetRescaleInterceptSlopeValue	424
25.140.2.2	SetSpacingValue	424
25.141	gdcm::ImageReader Class Reference	424
25.141.1	Detailed Description	427
25.141.2	Constructor & Destructor Documentation	427
25.141.2.1	ImageReader	427
25.141.2.2	~ImageReader	427
25.141.3	Member Function Documentation	427
25.141.3.1	GetImage	427
25.141.3.2	GetImage	427
25.141.3.3	Read	427
25.141.3.4	ReadACRNEMAIImage	428
25.141.3.5	ReadImage	428
25.142	gdcm::ImageRegionReader Class Reference	428
25.142.1	Detailed Description	430
25.142.2	Constructor & Destructor Documentation	430
25.142.2.1	ImageRegionReader	430
25.142.2.2	~ImageRegionReader	430
25.142.3	Member Function Documentation	430
25.142.3.1	ComputeBufferLength	430
25.142.3.2	GetRegion	430
25.142.3.3	Read	430
25.142.3.4	ReadInformation	430
25.142.3.5	ReadIntoBuffer	431
25.142.3.6	SetRegion	431
25.143	gdcm::ImageToImageFilter Class Reference	431
25.143.1	Detailed Description	432
25.143.2	Constructor & Destructor Documentation	433
25.143.2.1	ImageToImageFilter	433
25.143.2.2	~ImageToImageFilter	433
25.143.3	Member Function Documentation	433
25.143.3.1	GetInput	433
25.143.3.2	GetOutput	433
25.144	gdcm::ImageWriter Class Reference	433
25.144.1	Detailed Description	435
25.144.2	Constructor & Destructor Documentation	435

25.144.2.1ImageWriter	435
25.144.2.2~ImageWriter	435
25.144.3Member Function Documentation	435
25.144.3.1GetImage	435
25.144.3.2GetImage	435
25.144.3.3Write	435
25.145dcm::network::ImplementationClassUIDSub Class Reference	436
25.145.1Detailed Description	436
25.145.2Constructor & Destructor Documentation	436
25.145.2.1ImplementationClassUIDSub	436
25.145.3Member Function Documentation	436
25.145.3.1Print	436
25.145.3.2Read	436
25.145.3.3Size	436
25.145.3.4Write	436
25.146dcm::network::ImplementationUIDSub Class Reference	436
25.146.1Detailed Description	437
25.146.2Constructor & Destructor Documentation	437
25.146.2.1ImplementationUIDSub	437
25.146.3Member Function Documentation	437
25.146.3.1Write	437
25.147dcm::network::ImplementationVersionNameSub Class Reference	437
25.147.1Detailed Description	437
25.147.2Constructor & Destructor Documentation	437
25.147.2.1ImplementationVersionNameSub	437
25.147.3Member Function Documentation	437
25.147.3.1Print	437
25.147.3.2Read	437
25.147.3.3Size	437
25.147.3.4Write	438
25.148dcm::ImplicitDataElement Class Reference	438
25.148.1Detailed Description	439
25.148.2Member Function Documentation	440
25.148.2.1GetLength	440
25.148.2.2Read	440
25.148.2.3ReadPreValue	440
25.148.2.4ReadValue	440

25.148.2.5ReadWithLength	440
25.148.2.6Write	440
25.149dcm::InitializeEvent Class Reference	440
25.150dcm::IOD Class Reference	441
25.150.1Detailed Description	442
25.150.2Member Typedef Documentation	442
25.150.2.1MapIODEntry	442
25.150.2.2SizeType	442
25.150.3Constructor & Destructor Documentation	442
25.150.3.1IOD	442
25.150.4Member Function Documentation	442
25.150.4.1AddIODEntry	442
25.150.4.2Clear	442
25.150.4.3GetIODEntry	442
25.150.4.4GetNumberOfIODs	442
25.150.4.5GetTypeFromTag	443
25.150.5Friends And Related Function Documentation	443
25.150.5.1operator<<	443
25.151dcm::IODEntry Class Reference	443
25.151.1Detailed Description	443
25.151.2Constructor & Destructor Documentation	444
25.151.2.1IODEntry	444
25.151.3Member Function Documentation	444
25.151.3.1GetIE	444
25.151.3.2GetName	444
25.151.3.3GetRef	444
25.151.3.4GetUsage	444
25.151.3.5GetUsageType	444
25.151.3.6SetIE	444
25.151.3.7SetName	444
25.151.3.8SetRef	444
25.151.3.9SetUsage	444
25.151.4Friends And Related Function Documentation	444
25.151.4.1operator<<	444
25.152dcm::IODs Class Reference	445
25.152.1Detailed Description	445
25.152.2Member Typedef Documentation	445

25.152.2.1	IODMapType	445
25.152.2.2	IODMapTypeConstIterator	445
25.152.2.3	IODName	445
25.152.3	Constructor & Destructor Documentation	446
25.152.3.1	IODs	446
25.152.4	Member Function Documentation	446
25.152.4.1	AddIOD	446
25.152.4.2	Begin	446
25.152.4.3	Clear	446
25.152.4.4	End	446
25.152.4.5	GetIOD	446
25.152.5	Friends And Related Function Documentation	446
25.152.5.1	operator<<	446
25.153	dcm::IPPSorter Class Reference	446
25.153.1	Detailed Description	448
25.153.2	Constructor & Destructor Documentation	448
25.153.2.1	IPPSorter	448
25.153.2.2	~IPPSorter	448
25.153.3	Member Function Documentation	448
25.153.3.1	GetDirectionCosinesTolerance	448
25.153.3.2	GetZSpacing	448
25.153.3.3	GetZSpacingTolerance	448
25.153.3.4	SetComputeZSpacing	448
25.153.3.5	SetDirectionCosinesTolerance	449
25.153.3.6	SetDropDuplicatePositions	449
25.153.3.7	SetZSpacingTolerance	449
25.153.3.8	Sort	449
25.153.4	Member Data Documentation	449
25.153.4.1	ComputeZSpacing	449
25.153.4.2	DirCosTolerance	449
25.153.4.3	DropDuplicatePositions	450
25.153.4.4	ZSpacing	450
25.153.4.5	ZTolerance	450
25.154	dcm::Item Class Reference	450
25.154.1	Detailed Description	452
25.154.2	Constructor & Destructor Documentation	452
25.154.2.1	Item	452

25.154.2.2Item	452
25.154.3Member Function Documentation	452
25.154.3.1Clear	452
25.154.3.2FindDataElement	452
25.154.3.3GetDataElement	452
25.154.3.4GetLength	452
25.154.3.5GetNestedDataSet	452
25.154.3.6GetNestedDataSet	453
25.154.3.7InsertDataElement	453
25.154.3.8Read	453
25.154.3.9SetNestedDataSet	453
25.154.3.10Write	453
25.154.4Friends And Related Function Documentation	453
25.154.4.1operator<<	453
25.155dcm::IterationEvent Class Reference	453
25.156dcm::JPEG12Codec Class Reference	455
25.156.1Detailed Description	456
25.156.2Constructor & Destructor Documentation	456
25.156.2.1JPEG12Codec	456
25.156.2.2~JPEG12Codec	456
25.156.3Member Function Documentation	456
25.156.3.1DecodeByStreams	456
25.156.3.2GetHeaderInfo	456
25.156.3.3InternalCode	456
25.156.3.4sStateSuspension	456
25.157dcm::JPEG16Codec Class Reference	457
25.157.1Detailed Description	458
25.157.2Constructor & Destructor Documentation	458
25.157.2.1JPEG16Codec	458
25.157.2.2~JPEG16Codec	458
25.157.3Member Function Documentation	458
25.157.3.1DecodeByStreams	458
25.157.3.2GetHeaderInfo	458
25.157.3.3InternalCode	458
25.157.3.4sStateSuspension	458
25.158dcm::JPEG2000Codec Class Reference	459
25.158.1Detailed Description	460

25.158.2	Constructor & Destructor Documentation	460
25.158.2.1	JPEG2000Codec	460
25.158.2.2	~Jpeg2000Codec	460
25.158.3	Member Function Documentation	460
25.158.3.1	CanCode	460
25.158.3.2	CanDecode	461
25.158.3.3	Code	461
25.158.3.4	Decode	461
25.158.3.5	DecodeByStreams	461
25.158.3.6	DecodeExtent	461
25.158.3.7	GetHeaderInfo	461
25.158.3.8	GetQuality	461
25.158.3.9	GetRate	461
25.158.3.10	SetNumberOfResolutions	461
25.158.3.11	SetQuality	461
25.158.3.12	SetRate	461
25.158.3.13	SetReversible	461
25.158.3.14	SetTileSize	461
25.158.4	Friends And Related Function Documentation	461
25.158.4.1	Bitmap	461
25.158.4.2	ImageRegionReader	461
25.159	dcm::JPEG8Codec Class Reference	462
25.159.1	Detailed Description	463
25.159.2	Constructor & Destructor Documentation	463
25.159.2.1	JPEG8Codec	463
25.159.2.2	~Jpeg8Codec	463
25.159.3	Member Function Documentation	463
25.159.3.1	DecodeByStreams	463
25.159.3.2	GetHeaderInfo	463
25.159.3.3	InternalCode	463
25.159.3.4	IsStateSuspension	463
25.160	dcm::JPEGCodec Class Reference	464
25.160.1	Detailed Description	465
25.160.2	Constructor & Destructor Documentation	466
25.160.2.1	JPEGCodec	466
25.160.2.2	~JpegCodec	466
25.160.3	Member Function Documentation	466

25.160.3.1CanCode	466
25.160.3.2CanDecode	466
25.160.3.3Code	466
25.160.3.4ComputeOffsetTable	466
25.160.3.5Decode	466
25.160.3.6DecodeByStreams	467
25.160.3.7DecodeExtent	467
25.160.3.8GetHeaderInfo	467
25.160.3.9GetLossless	467
25.160.3.10GetQuality	467
25.160.3.11StateSuspension	467
25.160.3.12Valid	467
25.160.3.13SetBitSample	467
25.160.3.14SetLossless	467
25.160.3.15SetPixelFormat	467
25.160.3.16SetQuality	467
25.160.4Friends And Related Function Documentation	467
25.160.4.1ImageRegionReader	467
25.160.5Member Data Documentation	467
25.160.5.1BitSample	468
25.160.5.2Lossless	468
25.160.5.3Quality	468
25.160.6gdcm::JPEGLSCodec Class Reference	468
25.161.1Detailed Description	470
25.161.2Constructor & Destructor Documentation	470
25.161.2.1JPEGLSCodec	470
25.161.2.2~JPEGLSCodec	470
25.161.3Member Function Documentation	470
25.161.3.1CanCode	470
25.161.3.2CanDecode	470
25.161.3.3Code	470
25.161.3.4Decode	470
25.161.3.5Decode	470
25.161.3.6DecodeExtent	470
25.161.3.7GetBufferLength	470
25.161.3.8GetHeaderInfo	471
25.161.3.9GetLossless	471

25.161.3.1	SetBufferLength	. 471
25.161.3.1	SetLossless	. 471
25.161.3.1	SetLossyError	. 471
25.161.4	Friends And Related Function Documentation	. 471
25.161.4.1	ImageRegionReader	. 471
25.162	dcm::KAKADUCodec Class Reference	. 471
25.162.1	Detailed Description	. 473
25.162.2	Constructor & Destructor Documentation	. 473
25.162.2.1	KAKADUCodec	. 473
25.162.2.2	~KAKADUCodec	. 473
25.162.3	Member Function Documentation	. 473
25.162.3.1	CanCode	. 473
25.162.3.2	CanDecode	. 473
25.162.3.3	Code	. 473
25.162.3.4	Decode	. 473
25.163	dcm::LO Class Reference	. 473
25.163.1	Detailed Description	. 475
25.163.2	Member Typedef Documentation	. 475
25.163.2.1	const_iterator	. 475
25.163.2.2	const_reference	. 475
25.163.2.3	const_reverse_iterator	. 475
25.163.2.4	difference_type	. 475
25.163.2.5	iterator	. 475
25.163.2.6	pointer	. 475
25.163.2.7	reference	. 475
25.163.2.8	reverse_iterator	. 475
25.163.2.9	size_type	. 475
25.163.2.10	Superclass	. 475
25.163.2.11	Value_type	. 475
25.163.3	Constructor & Destructor Documentation	. 475
25.163.3.1	LO	. 475
25.163.3.2	LO	. 475
25.163.3.3	LO	. 475
25.163.3.4	LO	. 475
25.163.4	Member Function Documentation	. 475
25.163.4.1	IsValid	. 476
25.164	dcm::LookupTable Class Reference	. 476

25.164.1Detailed Description	478
25.164.2Member Enumeration Documentation	478
25.164.2.1LookupTableType	478
25.164.3Constructor & Destructor Documentation	478
25.164.3.1LookupTable	478
25.164.3.2~LookupTable	478
25.164.3.3LookupTable	478
25.164.4Member Function Documentation	478
25.164.4.1Allocate	478
25.164.4.2Clear	478
25.164.4.3Decode	478
25.164.4.4Decode	479
25.164.4.5GetBitSample	479
25.164.4.6GetBufferAsRGBA	479
25.164.4.7GetLUT	479
25.164.4.8GetLUTDescriptor	479
25.164.4.9GetLUTLength	479
25.164.4.10GetPointer	479
25.164.4.11InitializeBlueLUT	479
25.164.4.12Initialized	479
25.164.4.13InitializeGreenLUT	479
25.164.4.14InitializeLUT	479
25.164.4.15InitializeRedLUT	479
25.164.4.16Print	479
25.164.4.17SetBlueLUT	480
25.164.4.18SetGreenLUT	480
25.164.4.19SetLUT	480
25.164.4.20SetRedLUT	480
25.164.4.21WriteBufferAsRGBA	480
25.164.5Member Data Documentation	480
25.164.5.1BitSample	480
25.164.5.2IncompleteLUT	480
25.164.5.3Internal	480
25.165dcm::Scanner::ltstr Struct Reference	480
25.165.1Member Function Documentation	480
25.165.1.1operator()	480
25.166dcm::Macro Class Reference	480

25.166.1Detailed Description	481
25.166.2Member Typedef Documentation	481
25.166.2.1ArrayIncludeMacrosType	481
25.166.2.2MapModuleEntry	481
25.166.3Constructor & Destructor Documentation	481
25.166.3.1Macro	481
25.166.4Member Function Documentation	481
25.166.4.1AddMacroEntry	481
25.166.4.2Clear	482
25.166.4.3FindMacroEntry	482
25.166.4.4GetMacroEntry	482
25.166.4.5GetName	482
25.166.4.6SetName	482
25.166.4.7Verify	482
25.166.5Friends And Related Function Documentation	482
25.166.5.1operator<<	482
25.167gdcmmacros::Macros Class Reference	482
25.167.1Detailed Description	483
25.167.2Member Typedef Documentation	483
25.167.2.1ModuleMapType	483
25.167.3Constructor & Destructor Documentation	483
25.167.3.1Macros	483
25.167.4Member Function Documentation	483
25.167.4.1AddMacro	483
25.167.4.2Clear	483
25.167.4.3GetMacro	483
25.167.4.4IsEmpty	483
25.167.5Friends And Related Function Documentation	483
25.167.5.1operator<<	483
25.168gdcmmacros::network::MaximumLengthSub Class Reference	483
25.168.1Detailed Description	484
25.168.2Constructor & Destructor Documentation	484
25.168.2.1MaximumLengthSub	484
25.168.3Member Function Documentation	484
25.168.3.1GetMaximumLength	484
25.168.3.2Print	484
25.168.3.3Read	484

25.168.3.4	SetMaximumLength	484
25.168.3.5	Size	484
25.168.3.6	Write	484
25.169	gdcm::MD5 Class Reference	484
25.169.1	Detailed Description	485
25.169.2	Constructor & Destructor Documentation	485
25.169.2.1	MD5	485
25.169.2.2	~MD5	485
25.169.3	Member Function Documentation	485
25.169.3.1	Compute	485
25.169.3.2	ComputeFile	485
25.170	gdcm::MediaStorage Class Reference	485
25.170.1	Detailed Description	488
25.170.2	Member Enumeration Documentation	488
25.170.2.1	MSType	488
25.170.2.2	ObjectType	490
25.170.3	Constructor & Destructor Documentation	491
25.170.3.1	MediaStorage	491
25.170.4	Member Function Documentation	491
25.170.4.1	GetModality	491
25.170.4.2	GetModalityDimension	491
25.170.4.3	GetMSString	491
25.170.4.4	GetMSType	491
25.170.4.5	GetNumberOfModality	491
25.170.4.6	GetNumberOfMSString	491
25.170.4.7	GetNumberOfMSType	491
25.170.4.8	GetString	491
25.170.4.9	GuessFromModality	491
25.170.4.10	Image	491
25.170.4.11	Undefined	492
25.170.4.12	operator MSType	492
25.170.4.13	SetFromDataSet	492
25.170.4.14	SetFromFile	492
25.170.4.15	SetFromHeader	492
25.170.4.16	SetFromModality	492
25.170.4.17	SetFromSourceImageSequence	492
25.170.5	Friends And Related Function Documentation	492

25.170.5.1operator<<	492
25.171dcm::MemberCommand< T > Class Template Reference	492
25.171.1Detailed Description	494
25.171.2Member Typedef Documentation	494
25.171.2.1Self	494
25.171.2.2TConstMemberFunctionPointer	495
25.171.2.3TMemberFunctionPointer	495
25.171.3Constructor & Destructor Documentation	495
25.171.3.1MemberCommand	495
25.171.3.2~MemberCommand	495
25.171.4Member Function Documentation	495
25.171.4.1Execute	495
25.171.4.2Execute	495
25.171.4.3New	495
25.171.4.4SetCallbackFunction	495
25.171.4.5SetCallbackFunction	496
25.171.5Member Data Documentation	496
25.171.5.1m_ConstMemberFunction	496
25.171.5.2m_MemberFunction	496
25.171.5.3m_This	496
25.172dcm::MeshPrimitive Class Reference	496
25.172.1Detailed Description	498
25.172.2Member Typedef Documentation	498
25.172.2.1PrimitivesData	498
25.172.3Member Enumeration Documentation	498
25.172.3.1MPType	498
25.172.4Constructor & Destructor Documentation	499
25.172.4.1MeshPrimitive	499
25.172.4.2~MeshPrimitive	499
25.172.5Member Function Documentation	499
25.172.5.1AddPrimitiveData	499
25.172.5.2GetMPType	499
25.172.5.3GetMPTypeString	499
25.172.5.4GetNumberOfPrimitivesData	499
25.172.5.5GetPrimitiveData	499
25.172.5.6GetPrimitiveData	499
25.172.5.7GetPrimitiveData	499

25.172.5.8GetPrimitiveData	499
25.172.5.9GetPrimitivesData	499
25.172.5.10GetPrimitivesData	499
25.172.5.11GetPrimitiveType	499
25.172.5.12SetPrimitiveData	499
25.172.5.13SetPrimitiveData	499
25.172.5.14SetPrimitivesData	499
25.172.5.15SetPrimitiveType	499
25.172.6Member Data Documentation	499
25.172.6.1PrimitiveData	499
25.172.6.2PrimitiveType	499
25.173dcm::ModifiedEvent Class Reference	499
25.174dcm::Module Class Reference	501
25.174.1Detailed Description	501
25.174.2Member Typedef Documentation	502
25.174.2.1ArrayIncludeMacrosType	502
25.174.2.2MapModuleEntry	502
25.174.3Constructor & Destructor Documentation	502
25.174.3.1Module	502
25.174.4Member Function Documentation	502
25.174.4.1AddMacro	502
25.174.4.2AddModuleEntry	502
25.174.4.3Clear	502
25.174.4.4FindModuleEntryInMacros	502
25.174.4.5GetModuleEntryInMacros	502
25.174.4.6GetName	502
25.174.4.7SetName	502
25.174.4.8Verify	502
25.174.5Friends And Related Function Documentation	502
25.174.5.1operator<<	502
25.175dcm::ModuleEntry Class Reference	503
25.175.1Detailed Description	504
25.175.2Member Typedef Documentation	504
25.175.2.1Description	504
25.175.3Constructor & Destructor Documentation	504
25.175.3.1ModuleEntry	504
25.175.3.2~ModuleEntry	505

25.175.4	Member Function Documentation	505
25.175.4.1	GetDescription	505
25.175.4.2	GetName	505
25.175.4.3	GetType	505
25.175.4.4	SetDescription	505
25.175.4.5	SetName	505
25.175.4.6	SetType	505
25.175.5	Friends And Related Function Documentation	505
25.175.5.1	operator<<	505
25.175.6	Member Data Documentation	505
25.175.6.1	DataElementType	505
25.175.6.2	DescriptionField	505
25.175.6.3	Name	505
25.176	dcm::Modules Class Reference	505
25.176.1	Detailed Description	506
25.176.2	Member Typedef Documentation	506
25.176.2.1	ModuleMapType	506
25.176.3	Constructor & Destructor Documentation	506
25.176.3.1	Modules	506
25.176.4	Member Function Documentation	506
25.176.4.1	AddModule	506
25.176.4.2	Clear	506
25.176.4.3	GetModule	506
25.176.4.4	IsEmpty	506
25.176.5	Friends And Related Function Documentation	507
25.176.5.1	operator<<	507
25.177	dcm::MovePatientRootQuery Class Reference	507
25.177.1	Detailed Description	508
25.177.2	Constructor & Destructor Documentation	508
25.177.2.1	MovePatientRootQuery	508
25.177.3	Member Function Documentation	508
25.177.3.1	GetAbstractSyntaxUID	508
25.177.3.2	GetTagListByLevel	508
25.177.3.3	InitializeDataSet	508
25.177.3.4	ValidateQuery	508
25.177.4	Friends And Related Function Documentation	509
25.177.4.1	QueryFactory	509

25.178	gdcmm::MoveStudyRootQuery Class Reference	509
25.178.1	Detailed Description	510
25.178.2	Constructor & Destructor Documentation	510
25.178.2.1	MoveStudyRootQuery	510
25.178.3	Member Function Documentation	510
25.178.3.1	GetAbstractSyntaxUID	510
25.178.3.2	GetTagListByLevel	511
25.178.3.3	InitializeDataSet	511
25.178.3.4	ValidateQuery	511
25.178.4	Friends And Related Function Documentation	511
25.178.4.1	QueryFactory	511
25.179	gdcmm::NestedModuleEntries Class Reference	511
25.179.1	Detailed Description	513
25.179.2	Member Typedef Documentation	513
25.179.2.1	SizeType	513
25.179.3	Constructor & Destructor Documentation	513
25.179.3.1	NestedModuleEntries	513
25.179.4	Member Function Documentation	513
25.179.4.1	AddModuleEntry	513
25.179.4.2	GetModuleEntry	513
25.179.4.3	GetModuleEntry	513
25.179.4.4	GetNumberOfModuleEntries	513
25.179.5	Friends And Related Function Documentation	513
25.179.5.1	operator<<	513
25.180	gdcmm::NoEvent Class Reference	514
25.180.1	Detailed Description	514
25.181	gdcmm::Object Class Reference	514
25.181.1	Detailed Description	516
25.181.2	Constructor & Destructor Documentation	516
25.181.2.1	Object	516
25.181.2.2	~Object	516
25.181.2.3	Object	516
25.181.3	Member Function Documentation	516
25.181.3.1	operator=	516
25.181.3.2	Print	516
25.181.3.3	Register	516
25.181.3.4	UnRegister	516

25.181.4Friends And Related Function Documentation	516
25.181.4.1operator<<	516
25.181.4.2SmartPointer	517
25.182dcm::Orientation Class Reference	517
25.182.1Detailed Description	517
25.182.2Member Enumeration Documentation	518
25.182.2.1OrientationType	518
25.182.3Constructor & Destructor Documentation	518
25.182.3.1Orientation	518
25.182.3.2~Orientation	518
25.182.4Member Function Documentation	518
25.182.4.1GetLabel	518
25.182.4.2GetMajorAxisFromPatientRelativeDirectionCosine	518
25.182.4.3GetObliquityThresholdCosineValue	518
25.182.4.4GetType	518
25.182.4.5Print	518
25.182.4.6SetObliquityThresholdCosineValue	518
25.182.5Friends And Related Function Documentation	518
25.182.5.1operator<<	518
25.183dcm::Overlay Class Reference	519
25.183.1Detailed Description	521
25.183.2Member Enumeration Documentation	521
25.183.2.1OverlayType	521
25.183.3Constructor & Destructor Documentation	522
25.183.3.1Overlay	522
25.183.3.2~Overlay	522
25.183.3.3Overlay	522
25.183.4Member Function Documentation	522
25.183.4.1Decode	522
25.183.4.2Decompress	522
25.183.4.3GetBitPosition	522
25.183.4.4GetBitsAllocated	522
25.183.4.5GetBuffer	522
25.183.4.6GetColumns	522
25.183.4.7GetDescription	522
25.183.4.8GetGroup	522
25.183.4.9GetOrigin	522

25.183.4.10	GetOverlayData	523
25.183.4.10	GetOverlayTypeAsString	523
25.183.4.10	GetOverlayTypeFromString	523
25.183.4.10	GetRows	523
25.183.4.10	GetType	523
25.183.4.10	GetTypeAsEnum	523
25.183.4.10	GetUnpackBuffer	523
25.183.4.10	GetUnpackBuffer	523
25.183.4.10	GetUnpackBufferLength	523
25.183.4.10	GrabOverlayFromPixelData	523
25.183.4.20	Empty	523
25.183.4.20	InPixelData	523
25.183.4.20	InPixelData	523
25.183.4.20	Zero	523
25.183.4.20	Print	524
25.183.4.25	SetBitPosition	524
25.183.4.25	SetBitsAllocated	524
25.183.4.25	SetColumns	524
25.183.4.25	SetDescription	524
25.183.4.25	SetFrameOrigin	524
25.183.4.30	SetGroup	524
25.183.4.30	SetNumberOfFrames	524
25.183.4.30	SetOrigin	524
25.183.4.30	SetOverlay	524
25.183.4.30	SetRows	524
25.183.4.30	SetType	525
25.183.4.30	Update	525
25.184	dcm::ParseException Class Reference	525
25.184.1	Detailed Description	526
25.184.2	Constructor & Destructor Documentation	526
25.184.2.1	ParseException	526
25.184.2.2	~ParseException	526
25.184.3	Member Function Documentation	526
25.184.3.1	GetLastElement	526
25.184.3.2	operator=	526
25.184.3.3	SetLastElement	527
25.185	dcm::Parser Class Reference	527

25.185.1Detailed Description	528
25.185.2Member Typedef Documentation	528
25.185.2.1EndElementHandler	528
25.185.2.2StartElementHandler	528
25.185.3Member Enumeration Documentation	528
25.185.3.1ErrorType	528
25.185.4Constructor & Destructor Documentation	528
25.185.4.1Parser	528
25.185.4.2~Parser	528
25.185.5Member Function Documentation	528
25.185.5.1GetBuffer	528
25.185.5.2GetCurrentByteIndex	528
25.185.5.3GetErrorCode	528
25.185.5.4GetErrorString	528
25.185.5.5GetUserData	528
25.185.5.6Parse	529
25.185.5.7ParseBuffer	529
25.185.5.8Process	529
25.185.5.9SetElementHandler	529
25.185.5.10SetUserData	529
25.186dcm::Patient Class Reference	529
25.186.1Detailed Description	529
25.186.2Constructor & Destructor Documentation	529
25.186.2.1Patient	529
25.187dcm::network::PDataTFPDU Class Reference	529
25.187.1Detailed Description	531
25.187.2Member Typedef Documentation	531
25.187.2.1SizeType	531
25.187.3Constructor & Destructor Documentation	531
25.187.3.1PDataTFPDU	531
25.187.4Member Function Documentation	531
25.187.4.1AddPresentationDataValue	531
25.187.4.2GetNumberOfPresentationDataValues	531
25.187.4.3GetPresentationDataValue	531
25.187.4.4IsLastFragment	531
25.187.4.5Print	531
25.187.4.6Read	531

25.187.4.7ReadInto	531
25.187.4.8Size	531
25.187.4.9Write	531
25.188gdcmm::PDBelement Class Reference	532
25.188.1Detailed Description	533
25.188.2Constructor & Destructor Documentation	533
25.188.2.1PDBelement	533
25.188.3Member Function Documentation	533
25.188.3.1GetName	533
25.188.3.2GetValue	533
25.188.3.3operator==	533
25.188.3.4SetName	533
25.188.3.5SetValue	533
25.188.4Friends And Related Function Documentation	533
25.188.4.1operator<<	533
25.188.5Member Data Documentation	533
25.188.5.1NameField	533
25.188.5.2ValueField	533
25.189gdcmm::PDBHeader Class Reference	534
25.189.1Detailed Description	534
25.189.2Constructor & Destructor Documentation	535
25.189.2.1PDBHeader	535
25.189.2.2~PDBHeader	535
25.189.3Member Function Documentation	535
25.189.3.1FindPDBelementByName	535
25.189.3.2GetPDBeEnd	535
25.189.3.3GetPDBelementByName	535
25.189.3.4GetPDBInfoTag	535
25.189.3.5LoadFromDataElement	535
25.189.3.6Print	535
25.189.4Friends And Related Function Documentation	535
25.189.4.1operator<<	535
25.190gdcmm::PDFCodec Class Reference	536
25.190.1Detailed Description	537
25.190.2Constructor & Destructor Documentation	537
25.190.2.1PDFCodec	537
25.190.2.2~PDFCodec	537

25.190.3	Member Function Documentation	537
25.190.3.1	CanCode	537
25.190.3.2	CanDecode	537
25.190.3.3	Decode	537
25.190.4	dcm::network::PDUFactory Class Reference	537
25.191.1	Detailed Description	538
25.191.2	Member Function Documentation	538
25.191.2.1	ConstructAbortPDU	538
25.191.2.2	ConstructPDU	538
25.191.2.3	ConstructReleasePDU	538
25.191.2.4	CreateCEchoPDU	538
25.191.2.5	CreateCFindPDU	538
25.191.2.6	CreateCMovePDU	538
25.191.2.7	CreateCStoreRQPDU	538
25.191.2.8	CreateCStoreRSPPDU	538
25.191.2.9	DetermineEventByPDU	538
25.191.2.10	GetPDVs	538
25.190.5	dcm::PersonName Class Reference	539
25.192.1	Detailed Description	539
25.192.2	Member Function Documentation	539
25.192.2.1	GetMaxLength	539
25.192.2.2	GetNumberOfComponents	539
25.192.2.3	Print	539
25.192.2.4	SetBlob	539
25.192.2.5	SetComponents	539
25.192.2.6	SetComponents	539
25.192.3	Member Data Documentation	539
25.192.3.1	Component	540
25.192.3.2	MaxLength	540
25.192.3.3	MaxNumberOfComponents	540
25.192.3.4	Padding	540
25.192.3.5	Separator	540
25.190.6	dcm::PGXCodec Class Reference	540
25.193.1	Detailed Description	541
25.193.2	Constructor & Destructor Documentation	541
25.193.2.1	PGXCodec	541
25.193.2.2	~PGXCodec	541

25.193.3	Member Function Documentation	541
25.193.3.1	CanCode	541
25.193.3.2	CanDecode	541
25.193.3.3	GetHeaderInfo	542
25.193.3.4	Read	542
25.193.3.5	Write	542
25.194	dcm::PhotometricInterpretation Class Reference	542
25.194.1	Detailed Description	543
25.194.2	Member Enumeration Documentation	543
25.194.2.1	PIType	543
25.194.3	Constructor & Destructor Documentation	543
25.194.3.1	PhotometricInterpretation	543
25.194.4	Member Function Documentation	543
25.194.4.1	GetPIString	543
25.194.4.2	GetPIType	544
25.194.4.3	GetSamplesPerPixel	544
25.194.4.4	GetString	544
25.194.4.5	GetType	544
25.194.4.6	IsLossless	544
25.194.4.7	IsLossy	544
25.194.4.8	IsRetired	544
25.194.4.9	IsSameColorSpace	544
25.194.4.10	operator PIType	544
25.194.5	Friends And Related Function Documentation	544
25.194.5.1	operator <<	544
25.195	dcm::PixelFormat Class Reference	544
25.195.1	Detailed Description	546
25.195.2	Member Enumeration Documentation	546
25.195.2.1	ScalarType	546
25.195.3	Constructor & Destructor Documentation	546
25.195.3.1	PixelFormat	546
25.195.3.2	PixelFormat	546
25.195.3.3	~PixelFormat	546
25.195.4	Member Function Documentation	546
25.195.4.1	GetBitsAllocated	547
25.195.4.2	GetBitsStored	547
25.195.4.3	GetHighBit	547

25.195.4.4	GetMax	547
25.195.4.5	GetMin	547
25.195.4.6	GetPixelRepresentation	547
25.195.4.7	GetPixelSize	547
25.195.4.8	GetSamplesPerPixel	548
25.195.4.9	GetScalarType	548
25.195.4.10	GetScalarTypeAsString	548
25.195.4.11	IsValid	548
25.195.4.12	operator ScalarType	548
25.195.4.13	operator!=	548
25.195.4.14	operator!=	548
25.195.4.15	operator==	548
25.195.4.16	operator==	548
25.195.4.17	Print	548
25.195.4.18	SetBitsAllocated	548
25.195.4.19	SetBitsStored	548
25.195.4.20	SetHighBit	548
25.195.4.21	SetPixelRepresentation	548
25.195.4.22	SetSamplesPerPixel	548
25.195.4.23	SetScalarType	549
25.195.4.24	Validate	549
25.195.5	Friends And Related Function Documentation	549
25.195.5.1	Bitmap	549
25.195.5.2	operator<<	549
25.196	dcm::Pixmap Class Reference	549
25.196.1	Detailed Description	551
25.196.2	Constructor & Destructor Documentation	551
25.196.2.1	Pixmap	551
25.196.2.2	~Pixmap	551
25.196.3	Member Function Documentation	551
25.196.3.1	AreOverlaysInPixelData	551
25.196.3.2	GetCurve	552
25.196.3.3	GetCurve	552
25.196.3.4	GetIconImage	552
25.196.3.5	GetIconImage	552
25.196.3.6	GetNumberOfCurves	552
25.196.3.7	GetNumberOfOverlays	552

25.196.3.8GetOverlay	552
25.196.3.9GetOverlay	552
25.196.3.10Print	552
25.196.3.11RemoveOverlay	552
25.196.3.12SetIconImage	552
25.196.3.13SetNumberOfCurves	552
25.196.3.14SetNumberOfOverlays	552
25.196.4Member Data Documentation	552
25.196.4.1Curves	552
25.196.4.2Icon	552
25.196.4.3Overlays	552
25.197dcm::PixmapReader Class Reference	552
25.197.1Detailed Description	555
25.197.2Constructor & Destructor Documentation	555
25.197.2.1PixmapReader	555
25.197.2.2~PixmapReader	555
25.197.3Member Function Documentation	555
25.197.3.1GetPixmap	555
25.197.3.2GetPixmap	555
25.197.3.3Read	555
25.197.3.4ReadACRNEMAIImage	555
25.197.3.5ReadImage	555
25.197.3.6ReadImageInternal	556
25.197.4Member Data Documentation	556
25.197.4.1PixelData	556
25.198dcm::PixmapToPixmapFilter Class Reference	556
25.198.1Detailed Description	557
25.198.2Constructor & Destructor Documentation	557
25.198.2.1PixmapToPixmapFilter	557
25.198.2.2~PixmapToPixmapFilter	558
25.198.3Member Function Documentation	558
25.198.3.1GetInput	558
25.198.3.2GetOutput	558
25.198.3.3GetOutputAsPixmap	558
25.199dcm::PixmapWriter Class Reference	558
25.199.1Detailed Description	560
25.199.2Constructor & Destructor Documentation	560

25.199.2.1PixmapWriter	560
25.199.2.2~PixmapWriter	560
25.199.3Member Function Documentation	560
25.199.3.1DolconImage	560
25.199.3.2GetImage	560
25.199.3.3GetImage	560
25.199.3.4GetPixmap	561
25.199.3.5GetPixmap	561
25.199.3.6PrepareWrite	561
25.199.3.7SetImage	561
25.199.3.8SetPixmap	561
25.199.3.9Write	561
25.199.4Member Data Documentation	561
25.199.4.1PixelData	561
25.200dcm::PNMCodec Class Reference	561
25.200.1Detailed Description	563
25.200.2Constructor & Destructor Documentation	563
25.200.2.1PNMCodec	563
25.200.2.2~PNMCodec	563
25.200.3Member Function Documentation	563
25.200.3.1CanCode	563
25.200.3.2CanDecode	563
25.200.3.3GetBufferLength	563
25.200.3.4GetHeaderInfo	563
25.200.3.5Read	563
25.200.3.6SetBufferLength	563
25.200.3.7Write	564
25.200dcm::Preamble Class Reference	564
25.201.1Detailed Description	564
25.201.2Constructor & Destructor Documentation	565
25.201.2.1Preamble	565
25.201.2.2~Preamble	565
25.201.2.3Preamble	565
25.201.3Member Function Documentation	565
25.201.3.1Clear	565
25.201.3.2Create	565
25.201.3.3GetInternal	565

25.201.3.4	GetLength	565
25.201.3.5	IsEmpty	565
25.201.3.6	IsValid	565
25.201.3.7	operator=	565
25.201.3.8	Print	565
25.201.3.9	Read	565
25.201.3.10	Remove	565
25.201.3.11	Valid	565
25.201.3.12	Write	565
25.201.4	Friends And Related Function Documentation	565
25.201.4.1	operator<<	565
25.202	gdcmm::PresentationContext Class Reference	565
25.202.1	Detailed Description	566
25.202.2	Member Typedef Documentation	566
25.202.2.1	SizeType	566
25.202.2.2	TransferSyntaxArrayType	566
25.202.3	Constructor & Destructor Documentation	566
25.202.3.1	PresentationContext	566
25.202.3.2	PresentationContext	566
25.202.4	Member Function Documentation	566
25.202.4.1	AddTransferSyntax	566
25.202.4.2	GetAbstractSyntax	567
25.202.4.3	GetNumberOfTransferSyntaxes	567
25.202.4.4	GetPresentationContextID	567
25.202.4.5	GetTransferSyntax	567
25.202.4.6	operator==	567
25.202.4.7	Print	567
25.202.4.8	SetAbstractSyntax	567
25.202.4.9	SetPresentationContextID	567
25.203	gdcmm::network::PresentationContextAC Class Reference	567
25.203.1	Detailed Description	567
25.203.2	Constructor & Destructor Documentation	568
25.203.2.1	PresentationContextAC	568
25.203.3	Member Function Documentation	568
25.203.3.1	GetPresentationContextID	568
25.203.3.2	GetReason	568
25.203.3.3	GetTransferSyntax	568

25.203.3.4Print	568
25.203.3.5Read	568
25.203.3.6SetPresentationContextID	568
25.203.3.7SetReason	568
25.203.3.8SetTransferSyntax	568
25.203.3.9Size	568
25.203.3.10Write	568
25.204dcm::PresentationContextGenerator Class Reference	568
25.204.1Detailed Description	569
25.204.2Member Typedef Documentation	569
25.204.2.1PresentationContextArrayType	569
25.204.2.2SizeType	569
25.204.3Constructor & Destructor Documentation	569
25.204.3.1PresentationContextGenerator	569
25.204.4Member Function Documentation	569
25.204.4.1AddPresentationContext	570
25.204.4.2GenerateFromFilenames	570
25.204.4.3GenerateFromUID	570
25.204.4.4GetDefaultTransferSyntax	570
25.204.4.5GetPresentationContexts	570
25.204.4.6SetDefaultTransferSyntax	570
25.204.4.7SetMergeModeToAbstractSyntax	570
25.204.4.8SetMergeModeToTransferSyntax	570
25.205dcm::network::PresentationContextRQ Class Reference	570
25.205.1Detailed Description	571
25.205.2Member Typedef Documentation	571
25.205.2.1SizeType	571
25.205.3Constructor & Destructor Documentation	571
25.205.3.1PresentationContextRQ	571
25.205.3.2PresentationContextRQ	571
25.205.3.3PresentationContextRQ	571
25.205.4Member Function Documentation	571
25.205.4.1AddTransferSyntax	571
25.205.4.2GetAbstractSyntax	572
25.205.4.3GetAbstractSyntax	572
25.205.4.4GetNumberOfTransferSyntaxes	572
25.205.4.5GetPresentationContextID	572

25.205.4.6GetTransferSyntax	572
25.205.4.7GetTransferSyntax	572
25.205.4.8GetTransferSyntaxes	572
25.205.4.9operator==	572
25.205.4.10Print	572
25.205.4.11Read	572
25.205.4.12SetAbstractSyntax	572
25.205.4.13SetPresentationContextID	572
25.205.4.14Size	572
25.205.4.15Write	572
25.206dcm::network::PresentationDataValue Class Reference	572
25.206.1Detailed Description	573
25.206.2Constructor & Destructor Documentation	573
25.206.2.1PresentationDataValue	573
25.206.3Member Function Documentation	573
25.206.3.1ConcatenatePDVBlobs	573
25.206.3.2GetBlob	573
25.206.3.3GetIsCommand	573
25.206.3.4GetIsLastFragment	573
25.206.3.5GetMessageHeader	573
25.206.3.6GetPresentationContextID	573
25.206.3.7Print	573
25.206.3.8Read	573
25.206.3.9ReadInto	573
25.206.3.10SetBlob	573
25.206.3.11SetCommand	574
25.206.3.12DataSet	574
25.206.3.13SetLastFragment	574
25.206.3.14SetMessageHeader	574
25.206.3.15SetPresentationContextID	574
25.206.3.16Size	574
25.206.3.17Write	574
25.207dcm::Printer Class Reference	574
25.207.1Detailed Description	576
25.207.2Member Enumeration Documentation	576
25.207.2.1PrintStyles	576
25.207.3Constructor & Destructor Documentation	576

25.207.3.1Printer	. 576
25.207.3.2~Printer	. 576
25.207.4Member Function Documentation	. 576
25.207.4.1GetPrintStyle	. 576
25.207.4.2Print	. 576
25.207.4.3PrintDataElement	. 576
25.207.4.4PrintDataSet	. 576
25.207.4.5PrintSQ	. 577
25.207.4.6SetColor	. 577
25.207.4.7SetFile	. 577
25.207.4.8SetStyle	. 577
25.207.5Member Data Documentation	. 577
25.207.5.1F	. 577
25.207.5.2MaxPrintLength	. 577
25.207.5.3PrintStyle	. 577
25.208gdcmm::PrivateDict Class Reference	. 577
25.208.1Detailed Description	. 578
25.208.2Constructor & Destructor Documentation	. 578
25.208.2.1PrivateDict	. 578
25.208.2.2~PrivateDict	. 578
25.208.3Member Function Documentation	. 578
25.208.3.1AddDictEntry	. 578
25.208.3.2FindDictEntry	. 578
25.208.3.3GetDictEntry	. 578
25.208.3.4IsEmpty	. 578
25.208.3.5LoadDefault	. 578
25.208.3.6PrintXML	. 578
25.208.3.7RemoveDictEntry	. 578
25.208.4Friends And Related Function Documentation	. 578
25.208.4.1Dicts	. 578
25.208.4.2operator<<	. 578
25.209gdcmm::PrivateTag Class Reference	. 579
25.209.1Detailed Description	. 580
25.209.2Constructor & Destructor Documentation	. 580
25.209.2.1PrivateTag	. 580
25.209.3Member Function Documentation	. 580
25.209.3.1GetOwner	. 580

25.209.3.2operator<	580
25.209.3.3ReadFromCommaSeparatedString	580
25.209.3.4SetOwner	580
25.209.4Friends And Related Function Documentation	580
25.209.4.1operator<<	580
25.210gdcmm::ProgressEvent Class Reference	580
25.210.1Detailed Description	582
25.210.2Member Typedef Documentation	582
25.210.2.1Self	582
25.210.2.2Superclass	582
25.210.3Constructor & Destructor Documentation	582
25.210.3.1ProgressEvent	582
25.210.3.2~ProgressEvent	582
25.210.3.3ProgressEvent	582
25.210.4Member Function Documentation	582
25.210.4.1CheckEvent	582
25.210.4.2GetEventName	582
25.210.4.3GetProgress	582
25.210.4.4MakeObject	582
25.210.4.5SetProgress	582
25.211gdcmm::PVRGCodec Class Reference	583
25.211.1Detailed Description	584
25.211.2Constructor & Destructor Documentation	584
25.211.2.1PVRGCodec	584
25.211.2.2~PVRGCodec	584
25.211.3Member Function Documentation	584
25.211.3.1CanCode	584
25.211.3.2CanDecode	584
25.211.3.3Code	584
25.211.3.4Decode	585
25.212gdcmm::PythonFilter Class Reference	585
25.212.1Detailed Description	585
25.212.2Constructor & Destructor Documentation	585
25.212.2.1PythonFilter	585
25.212.2.2~PythonFilter	585
25.212.3Member Function Documentation	585
25.212.3.1GetFile	585

25.212.3.2	GetFile	585
25.212.3.3	SetDicts	585
25.212.3.4	SetFile	585
25.212.3.5	ToPyObject	585
25.212.3.6	UseDictAlways	586
25.213	gdcm::QueryBase Class Reference	586
25.213.1	Detailed Description	586
25.213.2	Constructor & Destructor Documentation	587
25.213.2.1	~QueryBase	587
25.213.3	Member Function Documentation	587
25.213.3.1	GetAllRequiredTags	587
25.213.3.2	GetAllTags	587
25.213.3.3	GetHierachicalSearchTags	587
25.213.3.4	GetName	587
25.213.3.5	GetOptionalTags	587
25.213.3.6	GetQueryLevel	587
25.213.3.7	GetRequiredTags	587
25.213.3.8	GetUniqueTags	588
25.214	gdcm::QueryFactory Class Reference	588
25.214.1	Detailed Description	588
25.214.2	Member Function Documentation	588
25.214.2.1	GetCharacterFromCurrentLocale	588
25.214.2.2	ListCharSets	588
25.214.2.3	ProduceCharacterSetDataElement	589
25.214.2.4	ProduceQuery	589
25.215	gdcm::QueryImage Class Reference	589
25.215.1	Detailed Description	590
25.215.2	Member Function Documentation	590
25.215.2.1	GetHierachicalSearchTags	590
25.215.2.2	GetName	590
25.215.2.3	GetOptionalTags	590
25.215.2.4	GetQueryLevel	591
25.215.2.5	GetRequiredTags	591
25.215.2.6	GetUniqueTags	591
25.216	gdcm::QueryPatient Class Reference	591
25.216.1	Detailed Description	592
25.216.2	Member Function Documentation	592

25.216.2.1	GetHierarchicalSearchTags	592
25.216.2.2	GetName	592
25.216.2.3	GetOptionalTags	592
25.216.2.4	GetQueryLevel	593
25.216.2.5	GetRequiredTags	593
25.216.2.6	GetUniqueTags	593
25.217	dcm::QuerySeries Class Reference	593
25.217.1	Detailed Description	594
25.217.2	Member Function Documentation	594
25.217.2.1	GetHierarchicalSearchTags	594
25.217.2.2	GetName	594
25.217.2.3	GetOptionalTags	594
25.217.2.4	GetQueryLevel	595
25.217.2.5	GetRequiredTags	595
25.217.2.6	GetUniqueTags	595
25.218	dcm::QueryStudy Class Reference	595
25.218.1	Detailed Description	596
25.218.2	Member Function Documentation	596
25.218.2.1	GetHierarchicalSearchTags	596
25.218.2.2	GetName	596
25.218.2.3	GetOptionalTags	596
25.218.2.4	GetQueryLevel	597
25.218.2.5	GetRequiredTags	597
25.218.2.6	GetUniqueTags	597
25.219	dcm::RAWCodec Class Reference	597
25.219.1	Detailed Description	598
25.219.2	Constructor & Destructor Documentation	598
25.219.2.1	RAWCodec	598
25.219.2.2	~RAWCodec	598
25.219.3	Member Function Documentation	598
25.219.3.1	CanCode	599
25.219.3.2	CanDecode	599
25.219.3.3	Code	599
25.219.3.4	Decode	599
25.219.3.5	DecodeByStreams	599
25.219.3.6	DecodeBytes	599
25.219.3.7	GetHeaderInfo	599

25.220.0 dcm::Reader Class Reference	599
25.220.1 Detailed Description	601
25.220.2 Constructor & Destructor Documentation	602
25.220.2.1 Reader	602
25.220.2.2 ~Reader	602
25.220.3 Member Function Documentation	602
25.220.3.1 CanRead	602
25.220.3.2 GetFile	602
25.220.3.3 GetFile	603
25.220.3.4 GetStreamPtr	603
25.220.3.5 Read	603
25.220.3.6 ReadDataSet	603
25.220.3.7 ReadMetaInformation	603
25.220.3.8 ReadPreamble	603
25.220.3.9 ReadSelectedTags	603
25.220.3.10 ReadUpToTag	603
25.220.3.11 SetFile	603
25.220.3.12 SetFileName	604
25.220.3.13 SetStream	604
25.220.4 Friends And Related Function Documentation	604
25.220.4.1 StreamImageReader	604
25.220.5 Member Data Documentation	604
25.220.5.1 F	604
25.221.0 dcm::Region Class Reference	604
25.221.1 Detailed Description	605
25.221.2 Constructor & Destructor Documentation	605
25.221.2.1 Region	605
25.221.2.2 ~Region	605
25.221.3 Member Function Documentation	605
25.221.3.1 Area	605
25.221.3.2 Clone	606
25.221.3.3 ComputeBoundingBox	606
25.221.3.4 Empty	606
25.221.3.5 IsValid	606
25.221.3.6 Print	606
25.222.0 dcm::Rescaler Class Reference	606
25.222.1 Detailed Description	607

25.222.2	Constructor & Destructor Documentation	608
25.222.2.1	Rescaler	608
25.222.2.2	~Rescaler	608
25.222.3	Member Function Documentation	608
25.222.3.1	ComputeInterceptSlopePixelType	608
25.222.3.2	ComputePixelTypeFromMinMax	608
25.222.3.3	GetIntercept	608
25.222.3.4	GetSlope	608
25.222.3.5	InverseRescale	608
25.222.3.6	InverseRescaleFunctionIntoBestFit	608
25.222.3.7	Rescale	608
25.222.3.8	RescaleFunctionIntoBestFit	608
25.222.3.9	SetIntercept	608
25.222.3.10	SetMinMaxForPixelType	608
25.222.3.11	SetPixelFormat	609
25.222.3.12	SetSlope	609
25.222.3.13	SetTargetPixelType	609
25.222.3.14	SetUseTargetPixelType	609
25.223	dcm::RLECodec Class Reference	609
25.223.1	Detailed Description	611
25.223.2	Constructor & Destructor Documentation	611
25.223.2.1	RLECodec	611
25.223.2.2	~RLECodec	611
25.223.3	Member Function Documentation	611
25.223.3.1	CanCode	611
25.223.3.2	CanDecode	611
25.223.3.3	Code	612
25.223.3.4	Decode	612
25.223.3.5	DecodeByStreams	612
25.223.3.6	DecodeExtent	612
25.223.3.7	GetBufferLength	612
25.223.3.8	GetHeaderInfo	612
25.223.3.9	SetBufferLength	612
25.223.3.10	SetLength	612
25.223.4	Friends And Related Function Documentation	612
25.223.4.1	ImageRegionReader	612
25.224	dcm::network::RoleSelectionSub Class Reference	612

25.224.1Detailed Description	613
25.224.2Constructor & Destructor Documentation	613
25.224.2.1RoleSelectionSub	613
25.224.3Member Function Documentation	613
25.224.3.1Print	613
25.224.3.2Read	613
25.224.3.3SetTuple	613
25.224.3.4Size	613
25.224.3.5Write	613
25.225dcm::SerieHelper::Rule Struct Reference	613
25.225.1Member Data Documentation	614
25.225.1.1elem	614
25.225.1.2group	614
25.225.1.3op	614
25.225.1.4value	614
25.226dcm::Scanner Class Reference	614
25.226.1Detailed Description	617
25.226.2Member Typedef Documentation	617
25.226.2.1ConstIterator	617
25.226.2.2MappingType	617
25.226.2.3TagToValue	617
25.226.2.4TagToValueValueType	618
25.226.2.5ValuesType	618
25.226.3Constructor & Destructor Documentation	618
25.226.3.1Scanner	618
25.226.3.2~Scanner	618
25.226.4Member Function Documentation	618
25.226.4.1AddPrivateTag	618
25.226.4.2AddSkipTag	618
25.226.4.3AddTag	618
25.226.4.4Begin	618
25.226.4.5ClearSkipTags	618
25.226.4.6ClearTags	618
25.226.4.7End	618
25.226.4.8GetAllFileNamesFromTagToValue	618
25.226.4.9GetFilenameFromTagToValue	618
25.226.4.10GetFileNames	618

25.226.4.1	GetKeys	618
25.226.4.1	GetMapping	619
25.226.4.1	GetMappingFromTagToValue	619
25.226.4.1	GetMappings	619
25.226.4.1	GetOrderedValues	619
25.226.4.1	GetValue	619
25.226.4.1	GetValues	619
25.226.4.1	GetValues	619
25.226.4.1	Key	620
25.226.4.2	New	620
25.226.4.2	Print	620
25.226.4.2	ProcessPublicTag	620
25.226.4.2	Scan	620
25.226.5	Friends And Related Function Documentation	620
25.226.5.1	operator<<	620
25.227	dcm::Segment Class Reference	620
25.227.1	Detailed Description	622
25.227.2	Member Typedef Documentation	623
25.227.2.1	SurfaceVector	623
25.227.3	Member Enumeration Documentation	623
25.227.3.1	ALGOType	623
25.227.4	Constructor & Destructor Documentation	623
25.227.4.1	Segment	623
25.227.4.2	~Segment	623
25.227.5	Member Function Documentation	623
25.227.5.1	AddSurface	623
25.227.5.2	GetALGOType	623
25.227.5.3	GetALGOTypeString	623
25.227.5.4	GetAnatomicRegion	623
25.227.5.5	GetAnatomicRegion	623
25.227.5.6	GetPropertyCategory	623
25.227.5.7	GetPropertyCategory	623
25.227.5.8	GetPropertyType	623
25.227.5.9	GetPropertyType	623
25.227.5.10	GetSegmentAlgorithmName	623
25.227.5.10	GetSegmentAlgorithmType	623
25.227.5.10	GetSegmentDescription	623

25.227.5.10	GetSegmentLabel	623
25.227.5.10	GetSegmentNumber	623
25.227.5.10	GetSurface	623
25.227.5.10	GetSurfaceCount	624
25.227.5.10	GetSurfaces	624
25.227.5.10	GetSurfaces	624
25.227.5.10	GetAnatomicRegion	624
25.227.5.20	GetPropertyCategory	624
25.227.5.20	GetPropertyType	624
25.227.5.20	SetSegmentAlgorithmName	624
25.227.5.20	SetSegmentAlgorithmType	624
25.227.5.20	SetSegmentAlgorithmType	624
25.227.5.20	SetSegmentDescription	624
25.227.5.20	SetSegmentLabel	624
25.227.5.20	SetSegmentNumber	624
25.227.5.20	SetSurfaceCount	624
25.227.6	Member Data Documentation	624
25.227.6.1	AnatomicRegion	624
25.227.6.2	PropertyCategory	624
25.227.6.3	PropertyType	624
25.227.6.4	SegmentAlgorithmName	624
25.227.6.5	SegmentAlgorithmType	624
25.227.6.6	SegmentDescription	624
25.227.6.7	SegmentLabel	624
25.227.6.8	SegmentNumber	624
25.227.6.9	SurfaceCount	624
25.227.6.10	Surfaces	624
25.228	dcm::SegmentedPaletteColorLookupTable Class Reference	625
25.228.1	Detailed Description	626
25.228.2	Constructor & Destructor Documentation	626
25.228.2.1	SegmentedPaletteColorLookupTable	626
25.228.2.2	~SegmentedPaletteColorLookupTable	626
25.228.3	Member Function Documentation	626
25.228.3.1	Print	626
25.228.3.2	SetLUT	626
25.229	dcm::SegmentReader Class Reference	626
25.229.1	Detailed Description	628

25.229.2	Member Typedef Documentation	628
25.229.2.1	SegmentMap	628
25.229.2.2	SegmentVector	628
25.229.3	Constructor & Destructor Documentation	628
25.229.3.1	SegmentReader	628
25.229.3.2	~SegmentReader	628
25.229.4	Member Function Documentation	629
25.229.4.1	GetSegments	629
25.229.4.2	GetSegments	629
25.229.4.3	Read	629
25.229.4.4	ReadSegment	629
25.229.4.5	ReadSegments	629
25.229.5	Member Data Documentation	629
25.229.5.1	Segments	629
25.230	dcm::SegmentWriter Class Reference	629
25.230.1	Detailed Description	630
25.230.2	Member Typedef Documentation	631
25.230.2.1	SegmentVector	631
25.230.3	Constructor & Destructor Documentation	631
25.230.3.1	SegmentWriter	631
25.230.3.2	~SegmentWriter	631
25.230.4	Member Function Documentation	631
25.230.4.1	AddSegment	631
25.230.4.2	GetNumberOfSegments	631
25.230.4.3	GetSegment	631
25.230.4.4	GetSegments	631
25.230.4.5	GetSegments	631
25.230.4.6	PrepareWrite	631
25.230.4.7	SetNumberOfSegments	631
25.230.4.8	SetSegments	631
25.230.4.9	Write	631
25.230.5	Member Data Documentation	631
25.230.5.1	Segments	631
25.231	dcm::SequenceOfFragments Class Reference	631
25.231.1	Detailed Description	633
25.231.2	Member Typedef Documentation	634
25.231.2.1	ConstIterator	634

25.231.2.2	FragmentVector	634
25.231.2.3	Iterator	634
25.231.2.4	SizeType	634
25.231.3	Constructor & Destructor Documentation	634
25.231.3.1	SequenceOfFragments	634
25.231.4	Member Function Documentation	634
25.231.4.1	AddFragment	634
25.231.4.2	Begin	634
25.231.4.3	Begin	634
25.231.4.4	Clear	634
25.231.4.5	ComputeByteLength	634
25.231.4.6	ComputeLength	634
25.231.4.7	End	634
25.231.4.8	End	634
25.231.4.9	GetBuffer	634
25.231.4.10	GetFragBuffer	634
25.231.4.11	GetFragment	635
25.231.4.12	GetLength	635
25.231.4.13	GetNumberOfFragments	635
25.231.4.14	GetTable	635
25.231.4.15	GetTable	635
25.231.4.16	New	635
25.231.4.17	operator==	635
25.231.4.18	Print	635
25.231.4.19	Read	635
25.231.4.20	ReadPreValue	635
25.231.4.21	ReadValue	635
25.231.4.22	SetLength	636
25.231.4.23	Write	636
25.231.4.24	WriteBuffer	636
25.231.5	gdcmm::SequenceOfItems Class Reference	636
25.232.1	Detailed Description	638
25.232.2	Member Typedef Documentation	639
25.232.2.1	ConstIterator	639
25.232.2.2	ItemVector	639
25.232.2.3	Iterator	639
25.232.2.4	SizeType	639

25.232.3	Constructor & Destructor Documentation	639
25.232.3.1	SequenceOfItems	639
25.232.4	Member Function Documentation	639
25.232.4.1	AddItem	639
25.232.4.2	Begin	639
25.232.4.3	Begin	639
25.232.4.4	Clear	639
25.232.4.5	ComputeLength	639
25.232.4.6	End	639
25.232.4.7	End	640
25.232.4.8	FindDataElement	640
25.232.4.9	GetItem	640
25.232.4.10	GetItem	640
25.232.4.11	GetLength	640
25.232.4.12	GetNumberOfItems	640
25.232.4.13	UndefinedLength	640
25.232.4.14	New	640
25.232.4.15	operator=	640
25.232.4.16	operator==	640
25.232.4.17	Print	640
25.232.4.18	Read	641
25.232.4.19	SetLength	641
25.232.4.20	SetLengthToUndefined	641
25.232.4.21	SetNumberOfItems	641
25.232.4.22	Write	641
25.232.5	Member Data Documentation	641
25.232.5.1	Items	641
25.232.5.2	SequenceLengthField	641
25.233	gdcmm::SerieHelper Class Reference	642
25.233.1	Detailed Description	643
25.233.2	Member Typedef Documentation	643
25.233.2.1	SerieRestrictions	643
25.233.2.2	SingleSerieUIDFileSetmap	643
25.233.3	Constructor & Destructor Documentation	643
25.233.3.1	SerieHelper	643
25.233.3.2	~SerieHelper	643
25.233.4	Member Function Documentation	644

25.233.4.1AddFile	644
25.233.4.2AddFileName	644
25.233.4.3AddRestriction	644
25.233.4.4AddRestriction	644
25.233.4.5AddRestriction	644
25.233.4.6Clear	644
25.233.4.7CreateDefaultUniqueSeriesIdentifier	644
25.233.4.8CreateUniqueSeriesIdentifier	644
25.233.4.9FileNameOrdering	644
25.233.4.10GetFirstSingleSerieUIDFileSet	644
25.233.4.11GetNextSingleSerieUIDFileSet	644
25.233.4.12ImagePositionPatientOrdering	644
25.233.4.13OrderFileList	644
25.233.4.14SetDirectory	644
25.233.4.15SetLoadMode	644
25.233.4.16SetUseSeriesDetails	644
25.233.4.17UserOrdering	644
25.233.5Member Data Documentation	644
25.233.5.1ItFileSetHt	644
25.233.5.2SingleSerieUIDFileSetHT	644
25.234dcm::Series Class Reference	644
25.234.1Detailed Description	645
25.234.2Constructor & Destructor Documentation	645
25.234.2.1Series	645
25.235dcm::network::ServiceClassApplicationInformation Class Reference	645
25.235.1Detailed Description	645
25.235.2Constructor & Destructor Documentation	645
25.235.2.1ServiceClassApplicationInformation	645
25.235.3Member Function Documentation	645
25.235.3.1Print	645
25.235.3.2Read	645
25.235.3.3SetTuple	645
25.235.3.4Size	646
25.235.3.5Write	646
25.236dcm::ServiceClassUser Class Reference	646
25.236.1Detailed Description	648
25.236.2Constructor & Destructor Documentation	648

25.236.2.1ServiceClassUser	648
25.236.2.2~ServiceClassUser	648
25.236.3Member Function Documentation	648
25.236.3.1GetAETitle	648
25.236.3.2GetCalledAETitle	648
25.236.3.3GetTimeout	648
25.236.3.4InitializeConnection	648
25.236.3.5IsPresentationContextAccepted	649
25.236.3.6SendEcho	649
25.236.3.7SendFind	649
25.236.3.8SendMove	649
25.236.3.9SendMove	649
25.236.3.10SendMove	649
25.236.3.11SendStore	649
25.236.3.12SendStore	649
25.236.3.13SendStore	649
25.236.3.14SetAETitle	649
25.236.3.15SetCalledAETitle	650
25.236.3.16SetHostname	650
25.236.3.17SetPort	650
25.236.3.18SetPortSCP	650
25.236.3.19SetPresentationContexts	650
25.236.3.20SetTimeout	650
25.236.3.21StartAssociation	650
25.236.3.22StopAssociation	651
25.237dcm::SHA1 Class Reference	651
25.237.1Detailed Description	651
25.237.2Constructor & Destructor Documentation	651
25.237.2.1SHA1	651
25.237.2.2~SHA1	651
25.237.3Member Function Documentation	651
25.237.3.1Compute	652
25.237.3.2ComputeFile	652
25.238dcm::SimpleMemberCommand< T > Class Template Reference	652
25.238.1Detailed Description	654
25.238.2Member Typedef Documentation	654
25.238.2.1Self	654

25.238.2.2TMemberFunctionPointer	654
25.238.3Constructor & Destructor Documentation	654
25.238.3.1SimpleMemberCommand	654
25.238.3.2~SimpleMemberCommand	654
25.238.4Member Function Documentation	654
25.238.4.1Execute	654
25.238.4.2Execute	654
25.238.4.3New	655
25.238.4.4SetCallbackFunction	655
25.238.5Member Data Documentation	655
25.238.5.1m_MemberFunction	655
25.238.5.2m_This	655
25.239gdcmm::SimpleSubjectWatcher Class Reference	655
25.239.1Detailed Description	656
25.239.2Constructor & Destructor Documentation	656
25.239.2.1SimpleSubjectWatcher	656
25.239.2.2~SimpleSubjectWatcher	656
25.239.3Member Function Documentation	656
25.239.3.1EndFilter	656
25.239.3.2ShowAbort	656
25.239.3.3ShowAnonymization	656
25.239.3.4ShowData	656
25.239.3.5ShowDataSet	656
25.239.3.6ShowIteration	656
25.239.3.7ShowProgress	656
25.239.3.8StartFilter	656
25.239.3.9TestAbortOff	656
25.239.3.10TestAbortOn	656
25.240gdcmm::SmartPointer< ObjectType > Class Template Reference	657
25.240.1Detailed Description	658
25.240.2Constructor & Destructor Documentation	658
25.240.2.1SmartPointer	658
25.240.2.2SmartPointer	658
25.240.2.3SmartPointer	658
25.240.2.4SmartPointer	658
25.240.2.5~SmartPointer	659
25.240.3Member Function Documentation	659

25.240.3.1GetPointer	659
25.240.3.2operator ObjectType *	659
25.240.3.3operator*	659
25.240.3.4operator->	659
25.240.3.5operator=	659
25.240.3.6operator=	659
25.240.3.7operator=	659
25.240dcm::network::SOPClassExtendedNegotiationSub Class Reference	659
25.241.1Detailed Description	660
25.241.2Constructor & Destructor Documentation	660
25.241.2.1SOPClassExtendedNegotiationSub	660
25.241.3Member Function Documentation	660
25.241.3.1Print	660
25.241.3.2Read	660
25.241.3.3SetTuple	660
25.241.3.4Size	660
25.241.3.5Write	660
25.240dcm::SOPClassUIDToIOD Class Reference	660
25.242.1Detailed Description	661
25.242.2Member Typedef Documentation	661
25.242.2.1const	661
25.242.3Member Function Documentation	661
25.242.3.1GetIOD	661
25.242.3.2GetIODFromSOPClassUID	661
25.242.3.3GetNumberOfSOPClassToIOD	661
25.242.3.4GetSOPClassUIDFromIOD	661
25.242.3.5GetSOPClassUIDToIOD	661
25.242.3.6GetSOPClassUIDToIODs	661
25.240dcm::Sorter Class Reference	661
25.243.1Detailed Description	663
25.243.2Member Typedef Documentation	663
25.243.2.1SelectionMap	663
25.243.2.2SortFunction	663
25.243.3Constructor & Destructor Documentation	664
25.243.3.1Sorter	664
25.243.3.2~Sorter	664
25.243.4Member Function Documentation	664

25.243.4.1AddSelect	664
25.243.4.2GetFileNames	664
25.243.4.3Print	664
25.243.4.4SetSortFunction	664
25.243.4.5Sort	664
25.243.4.6StableSort	664
25.243.5Friends And Related Function Documentation	665
25.243.5.1operator<<	665
25.243.6Member Data Documentation	665
25.243.6.1FileNames	665
25.243.6.2Selection	665
25.243.6.3SortFunc	665
25.244dcm::Spacing Class Reference	665
25.244.1Detailed Description	665
25.244.2Member Enumeration Documentation	666
25.244.2.1SpacingType	666
25.244.3Constructor & Destructor Documentation	666
25.244.3.1Spacing	666
25.244.3.2~Spacing	666
25.244.4Member Function Documentation	666
25.244.4.1ComputePixelAspectRatioFromPixelSpacing	666
25.245dcm::Spectroscopy Class Reference	667
25.245.1Detailed Description	667
25.245.2Constructor & Destructor Documentation	667
25.245.2.1Spectroscopy	667
25.246dcm::SplitMosaicFilter Class Reference	667
25.246.1Detailed Description	668
25.246.2Constructor & Destructor Documentation	668
25.246.2.1SplitMosaicFilter	668
25.246.2.2~SplitMosaicFilter	668
25.246.3Member Function Documentation	668
25.246.3.1ComputeMOSAICDimensions	668
25.246.3.2GetFile	668
25.246.3.3GetFile	668
25.246.3.4GetImage	668
25.246.3.5GetImage	668
25.246.3.6SetFile	668

25.246.3.7SetImage	668
25.246.3.8Split	668
25.247dcm::StartEvent Class Reference	668
25.248dcm::static_assert_test< x > Struct Template Reference	670
25.249dcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference	670
25.250dcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference	670
25.250.1Member Enumeration Documentation	670
25.250.1.1anonymous enum	670
25.251dcm::StreamImageReader Class Reference	670
25.251.1Detailed Description	671
25.251.2Constructor & Destructor Documentation	671
25.251.2.1StreamImageReader	671
25.251.2.2~StreamImageReader	671
25.251.3Member Function Documentation	671
25.251.3.1CanReadImage	671
25.251.3.2DefinePixelExtent	671
25.251.3.3DefineProperBufferLength	672
25.251.3.4GetDimensionsValueForResolution	672
25.251.3.5GetFile	672
25.251.3.6Read	672
25.251.3.7ReadImageInformation	672
25.251.3.8SetFileName	673
25.251.3.9SetStream	673
25.252dcm::StreamImageWriter Class Reference	673
25.252.1Detailed Description	675
25.252.2Constructor & Destructor Documentation	676
25.252.2.1StreamImageWriter	676
25.252.2.2~StreamImageWriter	676
25.252.3Member Function Documentation	676
25.252.3.1CanWriteFile	676
25.252.3.2DefinePixelExtent	676
25.252.3.3DefineProperBufferLength	676
25.252.3.4SetFile	676
25.252.3.5SetFileName	676
25.252.3.6SetStream	677
25.252.3.7Write	677
25.252.3.8WriteImageInformation	677

25.252.3.9WriteImageSubregionRAW	677
25.252.3.10WriteRawHeader	677
25.252.4Member Data Documentation	678
25.252.4.1mElementOffsets	678
25.252.4.2mElementOffsets1	678
25.252.4.3mspFile	678
25.252.4.4mWriter	678
25.252.4.5mXMax	678
25.252.4.6mXMin	678
25.252.4.7mYMax	678
25.252.4.8mYMin	678
25.252.4.9mZMax	678
25.252.4.10mZMin	678
25.253dcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference	678
25.253.1Detailed Description	680
25.253.2Member Typedef Documentation	680
25.253.2.1const_iterator	680
25.253.2.2const_reference	681
25.253.2.3const_reverse_iterator	681
25.253.2.4difference_type	681
25.253.2.5iterator	681
25.253.2.6pointer	681
25.253.2.7reference	681
25.253.2.8reverse_iterator	681
25.253.2.9size_type	681
25.253.2.10value_type	681
25.253.3Constructor & Destructor Documentation	681
25.253.3.1String	681
25.253.3.2String	681
25.253.3.3String	681
25.253.3.4String	681
25.253.4Member Function Documentation	681
25.253.4.1IsValid	681
25.253.4.2operator const char *	682
25.253.4.3Trim	682
25.253.4.4Trim	682
25.253.4.5Truncate	682

25.254	dcm::StringFilter Class Reference	682
25.254.1	Detailed Description	683
25.254.2	Constructor & Destructor Documentation	683
25.254.2.1	StringFilter	683
25.254.2.2	~StringFilter	683
25.254.3	Member Function Documentation	683
25.254.3.1	ExecuteQuery	683
25.254.3.2	~ExecuteQuery	683
25.254.3.3	FromString	683
25.254.3.4	FromString	683
25.254.3.5	GetFile	683
25.254.3.6	GetFile	683
25.254.3.7	SetDicts	683
25.254.3.8	SetFile	683
25.254.3.9	ToString	684
25.254.3.10	ToStringPair	684
25.254.3.11	ToStringPair	684
25.254.3.12	UseDictAlways	684
25.255	dcm::Study Class Reference	684
25.255.1	Detailed Description	684
25.255.2	Constructor & Destructor Documentation	685
25.255.2.1	Study	685
25.256	dcm::Subject Class Reference	685
25.256.1	Detailed Description	686
25.256.2	Constructor & Destructor Documentation	686
25.256.2.1	Subject	686
25.256.2.2	~Subject	686
25.256.3	Member Function Documentation	686
25.256.3.1	AddObserver	686
25.256.3.2	AddObserver	686
25.256.3.3	GetCommand	686
25.256.3.4	HasObserver	686
25.256.3.5	InvokeEvent	687
25.256.3.6	InvokeEvent	687
25.256.3.7	RemoveAllObservers	687
25.256.3.8	RemoveObserver	687
25.257	dcm::Surface Class Reference	687

25.257.1Detailed Description	690
25.257.2Member Enumeration Documentation	690
25.257.2.1STATES	690
25.257.2.2VIEWType	690
25.257.3Constructor & Destructor Documentation	690
25.257.3.1Surface	690
25.257.3.2~Surface	690
25.257.4Member Function Documentation	691
25.257.4.1GetAlgorithmFamily	691
25.257.4.2GetAlgorithmFamily	691
25.257.4.3GetAlgorithmName	691
25.257.4.4GetAlgorithmVersion	691
25.257.4.5GetAxisOfRotation	691
25.257.4.6GetCenterOfRotation	691
25.257.4.7GetFiniteVolume	691
25.257.4.8GetManifold	691
25.257.4.9GetMaximumPointDistance	691
25.257.4.10GetMeanPointDistance	691
25.257.4.11GetMeshPrimitive	691
25.257.4.12GetMeshPrimitive	691
25.257.4.13GetNumberOfSurfacePoints	691
25.257.4.14GetNumberOfVectors	691
25.257.4.15GetPointCoordinatesData	691
25.257.4.16GetPointCoordinatesData	691
25.257.4.17GetPointPositionAccuracy	691
25.257.4.18GetPointsBoundingBoxCoordinates	691
25.257.4.19GetProcessingAlgorithm	692
25.257.4.20GetProcessingAlgorithm	692
25.257.4.21GetRecommendedDisplayCIELabValue	692
25.257.4.22GetRecommendedDisplayCIELabValue	692
25.257.4.23GetRecommendedDisplayGrayscaleValue	692
25.257.4.24GetRecommendedPresentationOpacity	692
25.257.4.25GetRecommendedPresentationType	692
25.257.4.26GetSTATES	692
25.257.4.27GetSTATESString	692
25.257.4.28GetSurfaceComments	692
25.257.4.29GetSurfaceNumber	692

25.257.4.30	GetSurfaceProcessing	692
25.257.4.30	GetSurfaceProcessingDescription	692
25.257.4.30	GetSurfaceProcessingRatio	692
25.257.4.30	GetVectorAccuracy	692
25.257.4.30	GetVectorCoordinateData	692
25.257.4.30	GetVectorCoordinateData	692
25.257.4.30	GetVectorDimensionality	692
25.257.4.30	GetVIEWType	692
25.257.4.30	GetVIEWTypeString	692
25.257.4.30	GetAlgorithmFamily	692
25.257.4.40	GetAlgorithmName	692
25.257.4.40	GetAlgorithmVersion	692
25.257.4.40	GetAxisOfRotation	692
25.257.4.40	GetCenterOfRotation	693
25.257.4.40	GetFiniteVolume	693
25.257.4.40	GetManifold	693
25.257.4.40	GetMaximumPointDistance	693
25.257.4.40	GetMeanPointDistance	693
25.257.4.40	GetMeshPrimitive	693
25.257.4.40	GetNumberOfSurfacePoints	693
25.257.4.50	GetNumberOfVectors	693
25.257.4.50	GetPointCoordinatesData	693
25.257.4.50	GetPointPositionAccuracy	693
25.257.4.50	GetPointsBoundingBoxCoordinates	693
25.257.4.50	GetProcessingAlgorithm	693
25.257.4.50	GetRecommendedDisplayCIELabValue	693
25.257.4.50	GetRecommendedDisplayCIELabValue	693
25.257.4.50	GetRecommendedDisplayCIELabValue	693
25.257.4.50	GetRecommendedDisplayGrayscaleValue	693
25.257.4.50	GetRecommendedPresentationOpacity	693
25.257.4.60	GetRecommendedPresentationType	693
25.257.4.60	GetSurfaceComments	693
25.257.4.60	GetSurfaceNumber	693
25.257.4.60	GetSurfaceProcessing	693
25.257.4.60	GetSurfaceProcessingDescription	693
25.257.4.60	GetSurfaceProcessingRatio	693
25.257.4.60	GetVectorAccuracy	693

25.257.4.6SetVectorCoordinateData	693
25.257.4.6SetVectorDimensionality	694
25.258gdcmm::SurfaceHelper Class Reference	694
25.258.1Detailed Description	694
25.258.2Member Typedef Documentation	694
25.258.2.1ColorArray	694
25.258.3Member Function Documentation	694
25.258.3.1RecommendedDisplayCIELabToRGB	694
25.258.3.2RecommendedDisplayCIELabToRGB	695
25.258.3.3RGBToRecommendedDisplayCIELab	695
25.258.3.4RGBToRecommendedDisplayGrayscale	696
25.259gdcmm::SurfaceReader Class Reference	696
25.259.1Detailed Description	697
25.259.2Constructor & Destructor Documentation	698
25.259.2.1SurfaceReader	698
25.259.2.2~SurfaceReader	698
25.259.3Member Function Documentation	698
25.259.3.1GetNumberOfSurfaces	698
25.259.3.2Read	698
25.259.3.3ReadPointMacro	698
25.259.3.4ReadSurface	698
25.259.3.5ReadSurfaces	698
25.260gdcmm::SurfaceWriter Class Reference	698
25.260.1Detailed Description	700
25.260.2Constructor & Destructor Documentation	700
25.260.2.1SurfaceWriter	700
25.260.2.2~SurfaceWriter	700
25.260.3Member Function Documentation	700
25.260.3.1ComputeNumberOfSurfaces	700
25.260.3.2GetNumberOfSurfaces	700
25.260.3.3PrepareWrite	700
25.260.3.4PrepareWritePointMacro	700
25.260.3.5SetNumberOfSurfaces	700
25.260.3.6Write	700
25.260.4Member Data Documentation	700
25.260.4.1NumberOfSurfaces	700
25.261gdcmm::SwapCode Class Reference	700

25.261.1	Detailed Description	701
25.261.2	Member Enumeration Documentation	701
25.261.2.1	SwapCodeType	701
25.261.3	Constructor & Destructor Documentation	702
25.261.3.1	SwapCode	702
25.261.4	Member Function Documentation	702
25.261.4.1	GetIndex	702
25.261.4.2	GetSwapCodeString	702
25.261.4.3	operator SwapCode::SwapCodeType	702
25.261.5	Friends And Related Function Documentation	702
25.261.5.1	operator<<	702
25.262	dcm::SwapperDoOp Class Reference	702
25.262.1	Member Function Documentation	702
25.262.1.1	Swap	702
25.262.1.2	SwapArray	702
25.263	dcm::SwapperNoOp Class Reference	703
25.263.1	Detailed Description	703
25.263.2	Member Function Documentation	703
25.263.2.1	Swap	703
25.263.2.2	SwapArray	703
25.264	dcm::System Class Reference	703
25.264.1	Detailed Description	704
25.264.2	Member Function Documentation	704
25.264.2.1	DeleteDirectory	704
25.264.2.2	EncodeBytes	704
25.264.2.3	FileExists	705
25.264.2.4	FileIsDirectory	705
25.264.2.5	FileIsSymlink	705
25.264.2.6	FileSize	705
25.264.2.7	FileTime	705
25.264.2.8	FormatDateTime	705
25.264.2.9	GetCurrentDateTime	705
25.264.2.10	GetCurrentModuleFileName	706
25.264.2.11	GetCurrentProcessFileName	706
25.264.2.12	GetCurrentResourcesDirectory	706
25.264.2.13	GetCurrentWD	706
25.264.2.14	GetHostName	706

25.264.2.1	Get LastSystemError	706
25.264.2.1	Get LocaleCharset	706
25.264.2.1	Get Permissions	706
25.264.2.1	Get TimezoneOffsetFromUTC	706
25.264.2.1	Make Directory	706
25.264.2.2	Parse DateTime	707
25.264.2.2	Parse DateTime	707
25.264.2.2	Remove File	707
25.264.2.2	Set Permissions	707
25.264.2.2	Str CaseCmp	707
25.264.2.2	Str NCaseCmp	707
25.264.2.2	Str TokR	707
25.265	dcm::Table Class Reference	707
25.265.1	Detailed Description	708
25.265.2	Member Typedef Documentation	708
25.265.2.1	MapTableEntry	708
25.265.3	Constructor & Destructor Documentation	708
25.265.3.1	Table	708
25.265.3.2	~Table	708
25.265.4	Member Function Documentation	708
25.265.4.1	GetTableEntry	708
25.265.4.2	InsertEntry	708
25.265.5	Friends And Related Function Documentation	708
25.265.5.1	operator<<	708
25.266	dcm::TableEntry Class Reference	708
25.266.1	Detailed Description	709
25.266.2	Constructor & Destructor Documentation	709
25.266.2.1	TableEntry	709
25.266.2.2	~TableEntry	709
25.267	dcm::TableReader Class Reference	709
25.267.1	Detailed Description	710
25.267.2	Constructor & Destructor Documentation	710
25.267.2.1	TableReader	710
25.267.2.2	~TableReader	710
25.267.3	Member Function Documentation	710
25.267.3.1	CharacterDataHandler	710
25.267.3.2	EndElement	710

25.267.3.3	GetDefs	710
25.267.3.4	GetFilename	710
25.267.3.5	HandleIOD	710
25.267.3.6	HandleIODEntry	710
25.267.3.7	HandleMacro	710
25.267.3.8	HandleMacroEntry	710
25.267.3.9	HandleMacroEntryDescription	710
25.267.3.10	HandleModule	710
25.267.3.11	HandleModuleEntry	711
25.267.3.12	HandleModuleEntryDescription	711
25.267.3.13	HandleModuleInclude	711
25.267.3.14	Read	711
25.267.3.15	SetFilename	711
25.267.3.16	StartElement	711
25.268	dcm::network::TableRow Class Reference	711
25.268.1	Constructor & Destructor Documentation	712
25.268.1.1	TableRow	712
25.268.1.2	~TableRow	712
25.268.2	Member Data Documentation	712
25.268.2.1	transitions	712
25.269	dcm::Tag Class Reference	712
25.269.1	Detailed Description	714
25.269.2	Constructor & Destructor Documentation	714
25.269.2.1	Tag	714
25.269.2.2	Tag	714
25.269.2.3	Tag	714
25.269.3	Member Function Documentation	714
25.269.3.1	GetElement	714
25.269.3.2	GetElementTag	715
25.269.3.3	GetGroup	715
25.269.3.4	GetLength	715
25.269.3.5	GetPrivateCreator	715
25.269.3.6	IsGroupLength	715
25.269.3.7	IsGroupXX	715
25.269.3.8	IsIllegal	715
25.269.3.9	IsPrivate	716
25.269.3.10	IsPrivateCreator	716

25.269.3.11	Public	716
25.269.3.12	operator!=	716
25.269.3.13	operator<	716
25.269.3.14	operator<=	716
25.269.3.15	operator=	716
25.269.3.16	operator==	716
25.269.3.17	operator[]	716
25.269.3.18	operator[]	717
25.269.3.19	PrintAsPipeSeparatedString	717
25.269.3.20	Read	717
25.269.3.21	ReadFromCommaSeparatedString	717
25.269.3.22	ReadFromPipeSeparatedString	717
25.269.3.23	SetElement	717
25.269.3.24	SetElementTag	717
25.269.3.25	SetElementTag	717
25.269.3.26	SetGroup	718
25.269.3.27	SetPrivateCreator	718
25.269.3.28	Write	718
25.269.4	Friends And Related Function Documentation	718
25.269.4.1	operator<<	718
25.269.4.2	operator>>	718
25.269.5	Member Data Documentation	718
25.269.5.1	bytes	718
25.269.5.2	tag	718
25.269.5.3	tags	718
25.270	gdcmm::TagPath Class Reference	718
25.270.1	Detailed Description	719
25.270.2	Constructor & Destructor Documentation	719
25.270.2.1	TagPath	719
25.270.2.2	~TagPath	719
25.270.3	Member Function Documentation	719
25.270.3.1	ConstructFromString	719
25.270.3.2	ConstructFromTagList	719
25.270.3.3	IsValid	719
25.270.3.4	Print	720
25.270.3.5	Push	720
25.270.3.6	Push	720

25.271.0 dcm::Testing Class Reference	720
25.271.1 Detailed Description	721
25.271.2 Member Typedef Documentation	721
25.271.2.1 MD5DataImagesType	721
25.271.2.2 MediaStorageDataFilesType	721
25.271.3 Constructor & Destructor Documentation	721
25.271.3.1 Testing	721
25.271.3.2 ~Testing	721
25.271.4 Member Function Documentation	721
25.271.4.1 ComputeFileMD5	721
25.271.4.2 ComputeMD5	722
25.271.4.3 GetDataExtraRoot	722
25.271.4.4 GetDataRoot	722
25.271.4.5 GetFileName	722
25.271.4.6 GetFileNames	722
25.271.4.7 GetLossyFlagFromFile	722
25.271.4.8 GetMD5DataImage	722
25.271.4.9 GetMD5DataImages	722
25.271.4.10 GetMD5FromBrokenFile	722
25.271.4.11 GetMD5FromFile	723
25.271.4.12 GetMediaStorageDataFile	723
25.271.4.13 GetMediaStorageDataFiles	723
25.271.4.14 GetMediaStorageFromFile	723
25.271.4.15 GetNumberOfFileNames	723
25.271.4.16 GetNumberOfMD5DataImages	723
25.271.4.17 GetNumberOfMediaStorageDataFiles	723
25.271.4.18 GetPixelSpacingDataRoot	723
25.271.4.19 GetSelectedTagsOffsetFromFile	723
25.271.4.20 GetSourceDirectory	723
25.271.4.21 GetStreamOffsetFromFile	723
25.271.4.22 GetTempDirectory	723
25.271.4.23 GetTempDirectoryW	723
25.271.4.24 GetTempFilename	723
25.271.4.25 GetTempFilenameW	723
25.271.4.26 Print	724
25.272.0 dcm::Trace Class Reference	724
25.272.1 Detailed Description	725

25.272.2	Constructor & Destructor Documentation	725
25.272.2.1	Trace	725
25.272.2.2	~Trace	725
25.272.3	Member Function Documentation	725
25.272.3.1	DebugOff	725
25.272.3.2	DebugOn	725
25.272.3.3	ErrorOff	725
25.272.3.4	ErrorOn	725
25.272.3.5	GetDebugFlag	725
25.272.3.6	GetDebugStream	725
25.272.3.7	GetErrorFlag	725
25.272.3.8	GetErrorStream	725
25.272.3.9	GetStream	725
25.272.3.10	GetWarningFlag	726
25.272.3.11	GetWarningStream	726
25.272.3.12	SetDebug	726
25.272.3.13	SetDebugStream	726
25.272.3.14	SetError	726
25.272.3.15	SetErrorStream	726
25.272.3.16	SetStream	726
25.272.3.17	SetStreamToFile	726
25.272.3.18	SetWarning	726
25.272.3.19	SetWarningStream	726
25.272.3.20	WarningOff	727
25.272.3.21	WarningOn	727
25.273	gdcmm::TransferSyntax Class Reference	727
25.273.1	Detailed Description	728
25.273.2	Member Enumeration Documentation	729
25.273.2.1	NegotiatedType	729
25.273.2.2	TSType	729
25.273.3	Constructor & Destructor Documentation	729
25.273.3.1	TransferSyntax	729
25.273.4	Member Function Documentation	729
25.273.4.1	CanStoreLossy	729
25.273.4.2	GetNegotiatedType	730
25.273.4.3	GetString	730
25.273.4.4	GetSwapCode	730

25.273.4.5	GetTSSString	. 730
25.273.4.6	GetTSType	. 730
25.273.4.7	IsEncapsulated	. 730
25.273.4.8	IsEncoded	. 730
25.273.4.9	IsExplicit	. 730
25.273.4.10	IsImplicit	. 730
25.273.4.11	IsLossless	. 730
25.273.4.12	IsLossy	. 730
25.273.4.13	IsValid	. 730
25.273.4.14	operator TSType	. 730
25.273.5	Friends And Related Function Documentation	. 730
25.273.5.1	operator<<	. 731
25.274	dcm::network::TransferSyntaxSub Class Reference	. 731
25.274.1	Detailed Description	. 731
25.274.2	Constructor & Destructor Documentation	. 731
25.274.2.1	TransferSyntaxSub	. 731
25.274.3	Member Function Documentation	. 731
25.274.3.1	GetName	. 731
25.274.3.2	operator==	. 731
25.274.3.3	Print	. 731
25.274.3.4	Read	. 731
25.274.3.5	SetName	. 731
25.274.3.6	SetNameFromUID	. 731
25.274.3.7	Size	. 732
25.274.3.8	Write	. 732
25.275	dcm::network::Transition Struct Reference	. 732
25.275.1	Constructor & Destructor Documentation	. 732
25.275.1.1	Transition	. 733
25.275.1.2	~Transition	. 733
25.275.1.3	Transition	. 733
25.275.2	Member Function Documentation	. 733
25.275.2.1	MakeNew	. 733
25.275.3	Member Data Documentation	. 733
25.275.3.1	mAction	. 733
25.275.3.2	mEnd	. 733
25.276	dcm::Type Class Reference	. 733
25.276.1	Detailed Description	. 734

25.276.2	Member Enumeration Documentation	734
25.276.2.1	TypeType	734
25.276.3	Constructor & Destructor Documentation	735
25.276.3.1	Type	735
25.276.4	Member Function Documentation	735
25.276.4.1	GetTypeString	735
25.276.4.2	GetTypeType	735
25.276.4.3	operator TypeType	735
25.276.5	Friends And Related Function Documentation	735
25.276.5.1	operator<<	735
25.277	dcm::UI Struct Reference	735
25.277.1	Friends And Related Function Documentation	735
25.277.1.1	operator<<	735
25.277.2	Member Data Documentation	735
25.277.2.1	Internal	735
25.278	dcm::UIDGenerator Class Reference	736
25.278.1	Detailed Description	736
25.278.2	Constructor & Destructor Documentation	736
25.278.2.1	UIDGenerator	736
25.278.3	Member Function Documentation	737
25.278.3.1	Generate	737
25.278.3.2	GenerateUUID	737
25.278.3.3	GetGDCMUID	737
25.278.3.4	GetRoot	737
25.278.3.5	IsValid	737
25.278.3.6	SetRoot	737
25.279	dcm::UIDs Class Reference	737
25.279.1	Detailed Description	742
25.279.2	Member Typedef Documentation	742
25.279.2.1	TransferSyntaxStringsType	742
25.279.3	Member Enumeration Documentation	742
25.279.3.1	TSName	742
25.279.3.2	TSType	749
25.279.4	Member Function Documentation	755
25.279.4.1	GetName	755
25.279.4.2	GetNumberOfTransferSyntaxStrings	756
25.279.4.3	GetString	756

25.279.4.4	GetTransferSyntaxString	756
25.279.4.5	GetTransferSyntaxStrings	756
25.279.4.6	GetUIDName	756
25.279.4.7	GetUIDString	756
25.279.4.8	operator TSType	756
25.279.4.9	SetFromUID	756
25.280	dcm::network::ULAction Class Reference	756
25.280.1	Detailed Description	758
25.280.2	Constructor & Destructor Documentation	758
25.280.2.1	ULAction	758
25.280.2.2	~ULAction	758
25.280.3	Member Function Documentation	758
25.280.3.1	PerformAction	758
25.281	dcm::network::ULActionAA1 Class Reference	759
25.281.1	Member Function Documentation	759
25.281.1.1	PerformAction	759
25.282	dcm::network::ULActionAA2 Class Reference	760
25.282.1	Member Function Documentation	760
25.282.1.1	PerformAction	761
25.283	dcm::network::ULActionAA3 Class Reference	761
25.283.1	Member Function Documentation	762
25.283.1.1	PerformAction	762
25.284	dcm::network::ULActionAA4 Class Reference	762
25.284.1	Member Function Documentation	763
25.284.1.1	PerformAction	763
25.285	dcm::network::ULActionAA5 Class Reference	763
25.285.1	Member Function Documentation	764
25.285.1.1	PerformAction	764
25.286	dcm::network::ULActionAA6 Class Reference	764
25.286.1	Member Function Documentation	765
25.286.1.1	PerformAction	765
25.287	dcm::network::ULActionAA7 Class Reference	766
25.287.1	Member Function Documentation	766
25.287.1.1	PerformAction	766
25.288	dcm::network::ULActionAA8 Class Reference	767
25.288.1	Member Function Documentation	767
25.288.1.1	PerformAction	768

25.289	dcm::network::ULActionAE1 Class Reference	. 768
25.289.1	Member Function Documentation	. 769
25.289.1.1	PerformAction	. 769
25.290	dcm::network::ULActionAE2 Class Reference	. 769
25.290.1	Member Function Documentation	. 770
25.290.1.1	PerformAction	. 770
25.291	dcm::network::ULActionAE3 Class Reference	. 770
25.291.1	Member Function Documentation	. 771
25.291.1.1	PerformAction	. 771
25.292	dcm::network::ULActionAE4 Class Reference	. 771
25.292.1	Member Function Documentation	. 772
25.292.1.1	PerformAction	. 772
25.293	dcm::network::ULActionAE5 Class Reference	. 773
25.293.1	Member Function Documentation	. 773
25.293.1.1	PerformAction	. 773
25.294	dcm::network::ULActionAE6 Class Reference	. 774
25.294.1	Member Function Documentation	. 774
25.294.1.1	PerformAction	. 775
25.295	dcm::network::ULActionAE7 Class Reference	. 775
25.295.1	Member Function Documentation	. 776
25.295.1.1	PerformAction	. 776
25.296	dcm::network::ULActionAE8 Class Reference	. 776
25.296.1	Member Function Documentation	. 777
25.296.1.1	PerformAction	. 777
25.297	dcm::network::ULActionAR1 Class Reference	. 777
25.297.1	Member Function Documentation	. 778
25.297.1.1	PerformAction	. 778
25.298	dcm::network::ULActionAR10 Class Reference	. 778
25.298.1	Member Function Documentation	. 779
25.298.1.1	PerformAction	. 779
25.299	dcm::network::ULActionAR2 Class Reference	. 780
25.299.1	Member Function Documentation	. 780
25.299.1.1	PerformAction	. 780
25.300	dcm::network::ULActionAR3 Class Reference	. 781
25.300.1	Member Function Documentation	. 781
25.300.1.1	PerformAction	. 782
25.301	dcm::network::ULActionAR4 Class Reference	. 782

25.301.1Member Function Documentation	783
25.301.1.1PerformAction	783
25.302dcm::network::ULActionAR5 Class Reference	783
25.302.1Member Function Documentation	784
25.302.1.1PerformAction	784
25.303dcm::network::ULActionAR6 Class Reference	784
25.303.1Member Function Documentation	785
25.303.1.1PerformAction	785
25.304dcm::network::ULActionAR7 Class Reference	785
25.304.1Member Function Documentation	786
25.304.1.1PerformAction	786
25.305dcm::network::ULActionAR8 Class Reference	787
25.305.1Member Function Documentation	787
25.305.1.1PerformAction	787
25.306dcm::network::ULActionAR9 Class Reference	788
25.306.1Member Function Documentation	788
25.306.1.1PerformAction	789
25.307dcm::network::ULActionDT1 Class Reference	789
25.307.1Member Function Documentation	790
25.307.1.1PerformAction	790
25.308dcm::network::ULActionDT2 Class Reference	790
25.308.1Member Function Documentation	791
25.308.1.1PerformAction	791
25.309dcm::network::ULBasicCallback Class Reference	791
25.309.1Detailed Description	792
25.309.2Constructor & Destructor Documentation	792
25.309.2.1ULBasicCallback	792
25.309.2.2~ULBasicCallback	792
25.309.3Member Function Documentation	792
25.309.3.1GetDataSets	792
25.309.3.2GetResponses	792
25.309.3.3HandleDataSet	792
25.309.3.4HandleResponse	793
25.310dcm::network::ULConnection Class Reference	793
25.310.1Detailed Description	794
25.310.2Constructor & Destructor Documentation	794
25.310.2.1ULConnection	794

25.310.2.2~ULConnection	794
25.310.3Member Function Documentation	794
25.310.3.1AddAcceptedPresentationContext	794
25.310.3.2FindContext	794
25.310.3.3GetAcceptedPresentationContexts	794
25.310.3.4GetAcceptedPresentationContexts	794
25.310.3.5GetConnectionInfo	794
25.310.3.6GetMaxPDUSize	794
25.310.3.7GetPresentationContextACByID	794
25.310.3.8GetPresentationContextIDFromPresentationContext	794
25.310.3.9GetPresentationContextRQByID	794
25.310.3.10GetPresentationContexts	794
25.310.3.11GetProtocol	794
25.310.3.12GetState	794
25.310.3.13GetTimer	795
25.310.3.14InitializeConnection	795
25.310.3.15InitializeIncomingConnection	795
25.310.3.16SetMaxPDUSize	795
25.310.3.17SetPresentationContexts	795
25.310.3.18SetPresentationContexts	795
25.310.3.19SetState	795
25.310.3.20StopProtocol	795
25.310dcm::network::ULConnectionCallback Class Reference	795
25.311.1Detailed Description	796
25.311.2Constructor & Destructor Documentation	796
25.311.2.1ULConnectionCallback	796
25.311.2.2~ULConnectionCallback	796
25.311.3Member Function Documentation	796
25.311.3.1DataSetHandled	796
25.311.3.2DataSetHandles	796
25.311.3.3HandleDataSet	796
25.311.3.4HandleResponse	796
25.311.3.5ResetHandledDataSet	796
25.310dcm::network::ULConnectionInfo Class Reference	797
25.312.1Detailed Description	797
25.312.2Constructor & Destructor Documentation	797
25.312.2.1ULConnectionInfo	797

25.312.3	Member Function Documentation	. 797
25.312.3.1	GetCalledAETitle	. 797
25.312.3.2	GetCalledComputerName	. 797
25.312.3.3	GetCalledIPAddress	. 797
25.312.3.4	GetCalledIPPort	. 797
25.312.3.5	GetCallingAETitle	. 797
25.312.3.6	GetMaxPDULength	. 797
25.312.3.7	Initialize	. 797
25.312.3.8	SetMaxPDULength	. 798
25.313	dcm::network::ULConnectionManager Class Reference	. 798
25.313.1	Detailed Description	. 800
25.313.2	Constructor & Destructor Documentation	. 800
25.313.2.1	ULConnectionManager	. 800
25.313.2.2	~ULConnectionManager	. 800
25.313.3	Member Function Documentation	. 800
25.313.3.1	BreakConnection	. 800
25.313.3.2	BreakConnectionNow	. 800
25.313.3.3	EstablishConnection	. 800
25.313.3.4	EstablishConnectionMove	. 800
25.313.3.5	SendEcho	. 800
25.313.3.6	SendFind	. 800
25.313.3.7	SendFind	. 800
25.313.3.8	SendMove	. 800
25.313.3.9	SendMove	. 800
25.313.3.10	SendStore	. 800
25.313.3.11	SendStore	. 801
25.314	dcm::network::ULEvent Class Reference	. 801
25.314.1	Detailed Description	. 801
25.314.2	Constructor & Destructor Documentation	. 801
25.314.2.1	ULEvent	. 801
25.314.2.2	ULEvent	. 801
25.314.2.3	~ULEvent	. 801
25.314.3	Member Function Documentation	. 801
25.314.3.1	GetEvent	. 801
25.314.3.2	GetPDUs	. 801
25.314.3.3	SetEvent	. 801
25.314.3.4	SetPDU	. 802

25.315	dcm::network::ULTransitionTable Class Reference	802
25.315.1	Detailed Description	802
25.315.2	Constructor & Destructor Documentation	802
25.315.2.1	ULTransitionTable	802
25.315.3	Member Function Documentation	802
25.315.3.1	HandleEvent	802
25.315.3.2	PrintTable	802
25.316	dcm::network::ULWritingCallback Class Reference	802
25.316.1	Constructor & Destructor Documentation	803
25.316.1.1	ULWritingCallback	804
25.316.1.2	~ULWritingCallback	804
25.316.2	Member Function Documentation	804
25.316.2.1	HandleDataSet	804
25.316.2.2	HandleResponse	804
25.316.2.3	SetDirectory	804
25.317	dcm::UNExplicitDataElement Class Reference	804
25.317.1	Detailed Description	805
25.317.2	Member Function Documentation	805
25.317.2.1	GetLength	805
25.317.2.2	Read	806
25.317.2.3	ReadPreValue	806
25.317.2.4	ReadValue	806
25.317.2.5	ReadWithLength	806
25.318	dcm::UNExplicitImplicitDataElement Class Reference	806
25.318.1	Detailed Description	807
25.318.2	Member Function Documentation	807
25.318.2.1	GetLength	808
25.318.2.2	Read	808
25.318.2.3	ReadPreValue	808
25.318.2.4	ReadValue	808
25.319	dcm::Unpacker12Bits Class Reference	808
25.319.1	Detailed Description	808
25.319.2	Member Function Documentation	808
25.319.2.1	Pack	808
25.319.2.2	Unpack	809
25.320	dcm::Usage Class Reference	809
25.320.1	Detailed Description	809

25.320.2	Member Enumeration Documentation	810
25.320.2.1	UsageType	810
25.320.3	Constructor & Destructor Documentation	810
25.320.3.1	Usage	810
25.320.4	Member Function Documentation	810
25.320.4.1	GetUsageString	810
25.320.4.2	GetUsageType	810
25.320.4.3	operator UsageType	810
25.320.5	Friends And Related Function Documentation	810
25.320.5.1	operator<<	810
25.321	dcm::UserEvent Class Reference	810
25.322	dcm::network::UserInformation Class Reference	812
25.322.1	Detailed Description	812
25.322.2	Constructor & Destructor Documentation	812
25.322.2.1	UserInformation	812
25.322.2.2	~UserInformation	812
25.322.3	Member Function Documentation	812
25.322.3.1	AddRoleSelectionSub	812
25.322.3.2	AddSOPClassExtendedNegociationSub	812
25.322.3.3	GetMaximumLengthSub	812
25.322.3.4	GetMaximumLengthSub	812
25.322.3.5	operator=	812
25.322.3.6	Print	812
25.322.3.7	Read	812
25.322.3.8	Size	813
25.322.3.9	Write	813
25.323	dcm::Validate Class Reference	813
25.323.1	Detailed Description	814
25.323.2	Constructor & Destructor Documentation	814
25.323.2.1	Validate	814
25.323.2.2	~Validate	814
25.323.3	Member Function Documentation	814
25.323.3.1	GetValidatedFile	814
25.323.3.2	SetFile	814
25.323.3.3	Validation	814
25.323.4	Member Data Documentation	814
25.323.4.1	F	814

25.323.4.2V	814
25.324dcm::Value Class Reference	814
25.324.1Detailed Description	815
25.324.2Constructor & Destructor Documentation	815
25.324.2.1Value	815
25.324.2.2~Value	815
25.324.3Member Function Documentation	815
25.324.3.1Clear	815
25.324.3.2GetLength	816
25.324.3.3operator==	816
25.324.3.4SetLength	816
25.325dcm::ValueIO< TDE, TSwap, TType > Class Template Reference	816
25.325.1Detailed Description	816
25.325.2Member Function Documentation	816
25.325.2.1Read	816
25.325.2.2Write	816
25.326dcm::Version Class Reference	817
25.326.1Detailed Description	817
25.326.2Constructor & Destructor Documentation	817
25.326.2.1Version	817
25.326.2.2~Version	817
25.326.3Member Function Documentation	817
25.326.3.1GetBuildVersion	817
25.326.3.2GetMajorVersion	817
25.326.3.3GetMinorVersion	817
25.326.3.4GetVersion	817
25.326.3.5Print	817
25.326.4Friends And Related Function Documentation	818
25.326.4.1operator<<	818
25.327dcm::VL Class Reference	818
25.327.1Detailed Description	819
25.327.2Member Typedef Documentation	819
25.327.2.1Type	819
25.327.3Constructor & Destructor Documentation	819
25.327.3.1VL	819
25.327.4Member Function Documentation	819
25.327.4.1GetLength	819

25.327.4.2GetVL16Max	819
25.327.4.3GetVL32Max	819
25.327.4.4IsOdd	819
25.327.4.5IsUndefined	819
25.327.4.6operator uint32_t	819
25.327.4.7operator++	819
25.327.4.8operator++	819
25.327.4.9operator+=	819
25.327.4.10Read	820
25.327.4.11Read16	820
25.327.4.12SetToUndefined	820
25.327.4.13Write	820
25.327.4.14Write16	820
25.327.5Friends And Related Function Documentation	820
25.327.5.1operator<<	820
25.328gdcmm::VM Class Reference	820
25.328.1Detailed Description	822
25.328.2Member Enumeration Documentation	822
25.328.2.1VMType	822
25.328.3Constructor & Destructor Documentation	823
25.328.3.1VM	823
25.328.4Member Function Documentation	823
25.328.4.1Compatible	823
25.328.4.2GetIndex	823
25.328.4.3GetLength	823
25.328.4.4GetNumberOfElementsFromArray	823
25.328.4.5GetVMString	823
25.328.4.6GetVMType	824
25.328.4.7GetVMTypeFromLength	824
25.328.4.8IsValid	824
25.328.4.9operator VMType	824
25.328.5Friends And Related Function Documentation	824
25.328.5.1operator<<	824
25.328gdcmm::VMToLength< T > Struct Template Reference	824
25.329gdcmm::VR Class Reference	824
25.330.1Detailed Description	826
25.330.2Member Enumeration Documentation	826

25.330.2.1VRType	826
25.330.3Constructor & Destructor Documentation	827
25.330.3.1VR	827
25.330.4Member Function Documentation	827
25.330.4.1CanDisplay	827
25.330.4.2Compatible	827
25.330.4.3GetLength	828
25.330.4.4GetLength	828
25.330.4.5GetSize	828
25.330.4.6GetSizeof	828
25.330.4.7GetVRString	828
25.330.4.8GetVRStringFromFile	828
25.330.4.9GetVRType	828
25.330.4.10GetVRTypeFromFile	828
25.330.4.11ASCII	828
25.330.4.12ASCII2	828
25.330.4.13Binary	828
25.330.4.14Binary2	828
25.330.4.15Dual	828
25.330.4.16Swap	828
25.330.4.17Valid	828
25.330.4.18Valid	828
25.330.4.19VRFile	828
25.330.4.20operator VRType	828
25.330.4.21Read	828
25.330.4.22Write	828
25.330.5Friends And Related Function Documentation	829
25.330.5.1operator <<	829
25.330dcm::VR16ExplicitDataElement Class Reference	829
25.331.1Detailed Description	830
25.331.2Member Function Documentation	830
25.331.2.1GetLength	830
25.331.2.2Read	831
25.331.2.3ReadPreValue	831
25.331.2.4ReadValue	831
25.331.2.5ReadWithLength	831
25.330dcm::VRToEncoding< T > Struct Template Reference	831

25.333	dcm::VRToType< T > Struct Template Reference	831
25.333	Detailed Description	831
25.334	dcm::VRVLSIZE< T > Class Template Reference	832
25.335	dcm::VRVLSIZE< 0 > Class Template Reference	832
25.335	Member Function Documentation	832
25.335.1	1Read	832
25.335.1	2Write	832
25.336	dcm::VRVLSIZE< 1 > Class Template Reference	832
25.336	Member Function Documentation	832
25.336.1	1Read	832
25.336.1	2Write	832
25.337	vtkGDCMImageReader Class Reference	833
25.337	Detailed Description	835
25.337	Constructor & Destructor Documentation	835
25.337.2	1vtkGDCMImageReader	835
25.337.2	2~vtkGDCMImageReader	835
25.337.3	Member Function Documentation	835
25.337.3.1	1CanReadFile	835
25.337.3.2	2ExecuteData	835
25.337.3.3	3ExecuteInformation	835
25.337.3.4	4FillMedicalImageInformation	836
25.337.3.5	5GetDescriptiveName	836
25.337.3.6	6GetFileExtensions	836
25.337.3.7	7GetIconImage	836
25.337.3.8	8GetOverlay	836
25.337.3.9	9LoadSingleFile	836
25.337.3.10	10New	836
25.337.3.11	11PrintSelf	836
25.337.3.12	12RequestDataCompat	836
25.337.3.13	13RequestInformationCompat	836
25.337.3.14	14SetCurve	836
25.337.3.15	15SetFileNames	836
25.337.3.16	16SetFilePattern	836
25.337.3.17	17SetFilePrefix	836
25.337.3.18	18SetMedicalImageProperties	836
25.337.3.19	19SetBooleanMacro	836
25.337.3.20	20SetBooleanMacro	836

25.337.3.21tkBooleanMacro	836
25.337.3.22tkBooleanMacro	837
25.337.3.23tkBooleanMacro	837
25.337.3.24tkGetMacro	837
25.337.3.25tkGetMacro	837
25.337.3.26tkGetMacro	837
25.337.3.27tkGetMacro	837
25.337.3.28tkGetMacro	837
25.337.3.29tkGetMacro	837
25.337.3.30tkGetMacro	837
25.337.3.31tkGetMacro	837
25.337.3.32tkGetMacro	837
25.337.3.33tkGetMacro	837
25.337.3.34tkGetObjectMacro	837
25.337.3.35tkGetObjectMacro	837
25.337.3.36tkGetObjectMacro	837
25.337.3.37tkGetObjectMacro	837
25.337.3.38tkGetStringMacro	837
25.337.3.39tkGetStringMacro	837
25.337.3.40tkGetVector3Macro	837
25.337.3.41tkGetVector6Macro	837
25.337.3.42tkSetMacro	837
25.337.3.43tkSetMacro	837
25.337.3.44tkSetMacro	837
25.337.3.45tkSetMacro	837
25.337.3.46tkSetVector6Macro	837
25.337.3.47tkTypeRevisionMacro	837
25.337.4Member Data Documentation	838
25.337.4.1ApplyInverseVideo	838
25.337.4.2ApplyLookupTable	838
25.337.4.3ApplyPlanarConfiguration	838
25.337.4.4ApplyShiftScale	838
25.337.4.5ApplyYBRToRGB	838
25.337.4.6Curve	838
25.337.4.7DirectionCosines	838
25.337.4.8FileNames	838
25.337.4.9ForceRescale	838

25.337.4.10	IconDataScalarType	838
25.337.4.11	IconImageDataExtent	838
25.337.4.12	IconNumberOfScalarComponents	838
25.337.4.13	IconImageFormat	838
25.337.4.14	IconImageOrientationPatient	838
25.337.4.15	IconImagePositionPatient	838
25.337.4.16	LoadIconImage	838
25.337.4.17	LoadOverlays	838
25.337.4.18	LossyFlag	838
25.337.4.19	MedicalImageProperties	838
25.337.4.20	NumberOfIconImages	838
25.337.4.21	NumberOfOverlays	838
25.337.4.22	PlanarConfiguration	838
25.337.4.23	Scale	838
25.337.4.24	Shift	838
25.338	vtkGDCMImageWriter Class Reference	839
25.338.1	Detailed Description	840
25.338.2	Member Enumeration Documentation	841
25.338.2.1	CompressionTypes	841
25.338.3	Constructor & Destructor Documentation	841
25.338.3.1	vtkGDCMImageWriter	841
25.338.3.2	~vtkGDCMImageWriter	841
25.338.4	Member Function Documentation	841
25.338.4.1	GetDescriptiveName	841
25.338.4.2	GetFileExtensions	841
25.338.4.3	GetFileName	841
25.338.4.4	New	841
25.338.4.5	PrintSelf	841
25.338.4.6	SetDirectionCosines	841
25.338.4.7	SetDirectionCosinesFromImageOrientationPatient	841
25.338.4.8	SetFileNames	841
25.338.4.9	SetMedicalImageProperties	842
25.338.4.10	BooleanMacro	842
25.338.4.11	BooleanMacro	842
25.338.4.12	GetMacro	842
25.338.4.13	GetMacro	842
25.338.4.14	GetMacro	842

25.338.4.16	GetMacro	842
25.338.4.16	GetMacro	842
25.338.4.17	GetMacro	842
25.338.4.18	GetMacro	842
25.338.4.19	GetObjectMacro	842
25.338.4.20	GetObjectMacro	842
25.338.4.21	GetObjectMacro	842
25.338.4.22	GetStringMacro	842
25.338.4.23	GetStringMacro	842
25.338.4.24	SetMacro	842
25.338.4.25	SetMacro	842
25.338.4.26	SetMacro	842
25.338.4.27	SetMacro	842
25.338.4.28	SetMacro	842
25.338.4.29	SetMacro	842
25.338.4.30	SetMacro	842
25.338.4.31	SetStringMacro	842
25.338.4.32	SetStringMacro	843
25.338.4.33	TypeRevisionMacro	843
25.338.4.34	Write	843
25.338.4.35	WriteGDCMData	843
25.338.4.36	WriteSlice	843
25.339	vtkGDCMMedicalImageProperties Class Reference	843
25.339.1	Constructor & Destructor Documentation	844
25.339.1.1	vtkGDCMMedicalImageProperties	844
25.339.1.2	~vtkGDCMMedicalImageProperties	844
25.339.2	Member Function Documentation	844
25.339.2.1	Clear	845
25.339.2.2	GetFile	845
25.339.2.3	New	845
25.339.2.4	PrintSelf	845
25.339.2.5	PushBackFile	845
25.339.2.6	vtkTypeRevisionMacro	845
25.339.3	Friends And Related Function Documentation	845
25.339.3.1	vtkGDCMImageReader	845
25.339.3.2	vtkGDCMImageWriter	845
25.340	vtkGDCMPolyDataReader Class Reference	845

25.340.1	Detailed Description	847
25.340.2	Constructor & Destructor Documentation	847
25.340.2.1	vtkGDCMPolyDataReader	847
25.340.2.2	~vtkGDCMPolyDataReader	847
25.340.3	Member Function Documentation	847
25.340.3.1	FillMedicalImageInformation	847
25.340.3.2	New	847
25.340.3.3	PrintSelf	847
25.340.3.4	RequestData	847
25.340.3.5	RequestData_HemodynamicWaveformStorage	847
25.340.3.6	RequestData_RTStructureSetStorage	847
25.340.3.7	RequestInformation	847
25.340.3.8	RequestInformation_HemodynamicWaveformStorage	847
25.340.3.9	RequestInformation_RTStructureSetStorage	847
25.340.3.10	GetObjectMacro	847
25.340.3.11	GetObjectMacro	847
25.340.3.12	GetStringMacro	848
25.340.3.13	SetStringMacro	848
25.340.3.14	TypeRevisionMacro	848
25.340.4	Member Data Documentation	848
25.340.4.1	FileName	848
25.340.4.2	MedicalImageProperties	848
25.340.4.3	RTStructSetProperties	848
25.341	vtkGDCMPolyDataWriter Class Reference	848
25.341.1	Detailed Description	850
25.341.2	Constructor & Destructor Documentation	850
25.341.2.1	vtkGDCMPolyDataWriter	850
25.341.2.2	~vtkGDCMPolyDataWriter	850
25.341.3	Member Function Documentation	850
25.341.3.1	InitializeRTStructSet	850
25.341.3.2	New	850
25.341.3.3	PrintSelf	850
25.341.3.4	SetMedicalImageProperties	850
25.341.3.5	SetNumberOfInputPorts	850
25.341.3.6	SetRTStructSetProperties	850
25.341.3.7	TypeRevisionMacro	851
25.341.3.8	WriteData	851

25.341.3.9WriteRTSTRUCTData	851
25.341.3.10WriteRTSTRUCTInfo	851
25.341.4Member Data Documentation	851
25.341.4.1MedicalImageProperties	851
25.341.4.2RTStructSetProperties	851
25.342.vtkGDCMTesting Class Reference	851
25.342.1Detailed Description	852
25.342.2Member Typedef Documentation	853
25.342.2.1MD5MetaImagesType	853
25.342.3Constructor & Destructor Documentation	853
25.342.3.1vtkGDCMTesting	853
25.342.3.2~vtkGDCMTesting	853
25.342.4Member Function Documentation	853
25.342.4.1GetGDCMDataRoot	853
25.342.4.2GetMD5MetaImage	853
25.342.4.3GetMHDMD5FromFile	853
25.342.4.4GetNumberOfMD5MetaImages	853
25.342.4.5GetRAWMD5FromFile	853
25.342.4.6GetVTKDataRoot	853
25.342.4.7New	853
25.342.4.8PrintSelf	853
25.342.4.9vtkTypeRevisionMacro	853
25.343.vtkGDCMThreadedImageReader Class Reference	853
25.343.1Constructor & Destructor Documentation	855
25.343.1.1vtkGDCMThreadedImageReader	855
25.343.1.2~vtkGDCMThreadedImageReader	855
25.343.2Member Function Documentation	855
25.343.2.1ExecuteData	855
25.343.2.2ExecuteInformation	855
25.343.2.3New	855
25.343.2.4PrintSelf	855
25.343.2.5ReadFiles	855
25.343.2.6RequestDataCompat	855
25.343.2.7vtkBooleanMacro	855
25.343.2.8vtkGetMacro	855
25.343.2.9vtkSetMacro	855
25.343.2.10vtkSetMacro	855

25.343.2.1	tkSetMacro	855
25.343.2.1	tkTypeRevisionMacro	855
25.344	tkGDCMThreadedImageReader2 Class Reference	856
25.344	Constructor & Destructor Documentation	857
25.344.1	tkGDCMThreadedImageReader2	857
25.344.1.2	~tkGDCMThreadedImageReader2	857
25.344.2	Member Function Documentation	857
25.344.2.1	GetFileName	857
25.344.2.2	New	857
25.344.2.3	PrintSelf	858
25.344.2.4	RequestInformation	858
25.344.2.5	SetFileName	858
25.344.2.6	SetFileNames	858
25.344.2.7	SplitExtent	858
25.344.2.8	ThreadedRequestData	858
25.344.2.9	tkBooleanMacro	858
25.344.2.10	tkBooleanMacro	858
25.344.2.11	tkBooleanMacro	858
25.344.2.12	tkGetMacro	858
25.344.2.13	tkGetMacro	858
25.344.2.14	tkGetMacro	858
25.344.2.15	tkGetMacro	858
25.344.2.16	tkGetMacro	858
25.344.2.17	tkGetMacro	858
25.344.2.18	tkGetMacro	858
25.344.2.19	tkGetMacro	858
25.344.2.20	tkGetObjectMacro	858
25.344.2.21	tkGetVector3Macro	858
25.344.2.22	tkGetVector3Macro	858
25.344.2.23	tkGetVector6Macro	858
25.344.2.24	tkSetMacro	858
25.344.2.25	tkSetMacro	858
25.344.2.26	tkSetMacro	858
25.344.2.27	tkSetMacro	858
25.344.2.28	tkSetMacro	859
25.344.2.29	tkSetMacro	859
25.344.2.30	tkSetMacro	859

25.344.2.31tkSetVector3Macro	859
25.344.2.32tkSetVector3Macro	859
25.344.2.33tkSetVector6Macro	859
25.344.2.34tkTypeRevisionMacro	859
25.345.1tkImageColorViewer Class Reference	859
25.345.1.1Detailed Description	862
25.345.2Member Enumeration Documentation	862
25.345.2.1anonymous enum	862
25.345.3Constructor & Destructor Documentation	862
25.345.3.1vtkImageColorViewer	862
25.345.3.2~vtkImageColorViewer	862
25.345.4Member Function Documentation	862
25.345.4.1AddInput	862
25.345.4.2AddInputConnection	862
25.345.4.3GetColorLevel	862
25.345.4.4GetColorWindow	862
25.345.4.5GetInput	862
25.345.4.6GetOffScreenRendering	862
25.345.4.7GetOverlayVisibility	862
25.345.4.8GetPosition	863
25.345.4.9GetSize	863
25.345.4.10GetSliceMax	863
25.345.4.10GetSliceMin	863
25.345.4.10GetSliceRange	863
25.345.4.10GetSliceRange	863
25.345.4.10GetSliceRange	863
25.345.4.10GetWindowName	863
25.345.4.11InstallPipeline	863
25.345.4.11New	863
25.345.4.11PrintSelf	863
25.345.4.11Render	863
25.345.4.20GetColorLevel	863
25.345.4.23SetColorWindow	863
25.345.4.22SetDisplayId	863
25.345.4.25SetInput	863
25.345.4.23SetInputConnection	863
25.345.4.25SetOffScreenRendering	863

25.345.4.26	SetOverlayVisibility	863
25.345.4.27	SetParentId	864
25.345.4.28	SetPosition	864
25.345.4.29	SetPosition	864
25.345.4.30	SetRenderer	864
25.345.4.31	SetRenderWindow	864
25.345.4.32	SetSize	864
25.345.4.33	SetSize	864
25.345.4.34	SetSlice	864
25.345.4.35	SetSliceOrientation	864
25.345.4.36	SetSliceOrientationToXY	864
25.345.4.37	SetSliceOrientationToXZ	864
25.345.4.38	SetSliceOrientationToYZ	864
25.345.4.39	SetupInteractor	864
25.345.4.40	SetWindowId	865
25.345.4.41	UninstallPipeline	865
25.345.4.42	UpdateDisplayExtent	865
25.345.4.43	UpdateOrientation	865
25.345.4.44	TK_LEGACY	865
25.345.4.45	TK_LEGACY	865
25.345.4.46	TK_LEGACY	865
25.345.4.47	TK_LEGACY	865
25.345.4.48	BooleanMacro	865
25.345.4.49	GetMacro	865
25.345.4.50	GetMacro	865
25.345.4.51	GetObjectMacro	865
25.345.4.52	GetObjectMacro	865
25.345.4.53	GetObjectMacro	865
25.345.4.54	GetObjectMacro	865
25.345.4.55	GetObjectMacro	865
25.345.4.56	TypeRevisionMacro	865
25.345.5	Member Data Documentation	865
25.345.5.1	FirstRender	865
25.345.5.2	ImageActor	865
25.345.5.3	Interactor	865
25.345.5.4	InteractorStyle	865
25.345.5.5	OverlayImageActor	865

25.345.5.6	Renderer	865
25.345.5.7	RenderWindow	865
25.345.5.8	Slice	865
25.345.5.9	SliceOrientation	866
25.345.5.10	WindowLevel	866
25.346.0	vtkImageMapToColors16 Class Reference	866
25.346.1	Constructor & Destructor Documentation	867
25.346.1.1	vtkImageMapToColors16	867
25.346.1.2	~vtkImageMapToColors16	867
25.346.2	Member Function Documentation	867
25.346.2.1	GetMTime	867
25.346.2.2	New	867
25.346.2.3	PrintSelf	868
25.346.2.4	RequestData	868
25.346.2.5	RequestInformation	868
25.346.2.6	SetLookupTable	868
25.346.2.7	SetOutputFormatToLuminance	868
25.346.2.8	SetOutputFormatToLuminanceAlpha	868
25.346.2.9	SetOutputFormatToRGB	868
25.346.2.10	SetOutputFormatToRGBA	868
25.346.2.11	ThreadedRequestData	868
25.346.2.12	vtkBooleanMacro	868
25.346.2.13	vtkGetMacro	868
25.346.2.14	vtkGetMacro	868
25.346.2.15	vtkGetMacro	868
25.346.2.16	vtkGetObjectMacro	868
25.346.2.17	vtkSetMacro	868
25.346.2.18	vtkSetMacro	868
25.346.2.19	vtkSetMacro	868
25.346.2.20	vtkTypeRevisionMacro	868
25.346.3	Member Data Documentation	868
25.346.3.1	ActiveComponent	868
25.346.3.2	DataWasPassed	868
25.346.3.3	LookupTable	868
25.346.3.4	OutputFormat	868
25.346.3.5	PassAlphaToOutput	869
25.347	vtkImageMapToWindowLevelColors2 Class Reference	869

25.347. Constructor & Destructor Documentation	870
25.347.1.1vtkImageMapToWindowLevelColors2	870
25.347.1.2~vtkImageMapToWindowLevelColors2	870
25.347.2 Member Function Documentation	870
25.347.2.1New	870
25.347.2.2PrintSelf	870
25.347.2.3RequestData	870
25.347.2.4RequestInformation	870
25.347.2.5ThreadedRequestData	870
25.347.2.6vtkGetMacro	870
25.347.2.7vtkGetMacro	870
25.347.2.8vtkSetMacro	871
25.347.2.9vtkSetMacro	871
25.347.2.10vtkTypeRevisionMacro	871
25.347.3 Member Data Documentation	871
25.347.3.1Level	871
25.347.3.2Window	871
25.348.vtkImagePlanarComponentsToComponents Class Reference	871
25.348. Constructor & Destructor Documentation	872
25.348.1.1vtkImagePlanarComponentsToComponents	872
25.348.1.2~vtkImagePlanarComponentsToComponents	872
25.348.2 Member Function Documentation	872
25.348.2.1New	872
25.348.2.2PrintSelf	872
25.348.2.3RequestData	873
25.348.2.4vtkTypeRevisionMacro	873
25.349.vtkImageRGBToYBR Class Reference	873
25.349. Constructor & Destructor Documentation	874
25.349.1.1vtkImageRGBToYBR	874
25.349.1.2~vtkImageRGBToYBR	874
25.349.2 Member Function Documentation	874
25.349.2.1New	874
25.349.2.2PrintSelf	874
25.349.2.3ThreadedExecute	874
25.349.2.4vtkTypeRevisionMacro	874
25.350.vtkImageYBRToRGB Class Reference	874
25.350. Constructor & Destructor Documentation	876

25.350.1.1	vtkImageYBRToRGB	876
25.350.1.2	~vtkImageYBRToRGB	876
25.350.2	Member Function Documentation	876
25.350.2.1	New	876
25.350.2.2	PrintSelf	876
25.350.2.3	ThreadedExecute	876
25.350.2.4	vtkTypeRevisionMacro	876
25.351	vtkLookupTable16 Class Reference	876
25.351.1	Constructor & Destructor Documentation	877
25.351.1.1	vtkLookupTable16	877
25.351.1.2	~vtkLookupTable16	877
25.351.2	Member Function Documentation	877
25.351.2.1	Build	878
25.351.2.2	GetPointer	878
25.351.2.3	MapScalarsThroughTable2	878
25.351.2.4	New	878
25.351.2.5	PrintSelf	878
25.351.2.6	SetNumberOfTableValues	878
25.351.2.7	vtkTypeRevisionMacro	878
25.351.2.8	WritePointer	878
25.351.3	Member Data Documentation	878
25.351.3.1	Table16	878
25.352	vtkRTStructSetProperties Class Reference	878
25.352.1	Detailed Description	880
25.352.2	Constructor & Destructor Documentation	880
25.352.2.1	vtkRTStructSetProperties	880
25.352.2.2	~vtkRTStructSetProperties	880
25.352.3	Member Function Documentation	880
25.352.3.1	AddContourReferencedFrameOfReference	881
25.352.3.2	AddReferencedFrameOfReference	881
25.352.3.3	AddStructureSetROI	881
25.352.3.4	AddStructureSetROIObservation	881
25.352.3.5	Clear	881
25.352.3.6	DeepCopy	881
25.352.3.7	GetContourReferencedFrameOfReferenceClassUID	881
25.352.3.8	GetContourReferencedFrameOfReferenceInstanceUID	881
25.352.3.9	GetNumberOfContourReferencedFrameOfReferences	881

25.352.3.10	GetNumberOfContourReferencedFrameOfReferences	881
25.352.3.11	GetNumberOfReferencedFrameOfReferences	881
25.352.3.12	GetNumberOfStructureSetROIs	881
25.352.3.13	GetReferencedFrameOfReferenceClassUID	881
25.352.3.14	GetReferencedFrameOfReferenceInstanceUID	881
25.352.3.15	GetStructureSetObservationNumber	881
25.352.3.16	GetStructureSetROIDescription	881
25.352.3.17	GetStructureSetROIGenerationAlgorithm	881
25.352.3.18	GetStructureSetROIName	881
25.352.3.19	GetStructureSetROINumber	881
25.352.3.20	GetStructureSetROIObservationLabel	881
25.352.3.21	GetStructureSetROIRefFrameRefUID	881
25.352.3.22	GetStructureSetRTROIInterpretedType	881
25.352.3.23	New	882
25.352.3.24	PrintSelf	882
25.352.3.25	tkGetStringMacro	882
25.352.3.26	tkGetStringMacro	882
25.352.3.27	tkGetStringMacro	882
25.352.3.28	tkGetStringMacro	882
25.352.3.29	tkGetStringMacro	882
25.352.3.30	tkGetStringMacro	882
25.352.3.31	tkGetStringMacro	882
25.352.3.32	tkGetStringMacro	882
25.352.3.33	tkGetStringMacro	882
25.352.3.34	tkSetStringMacro	882
25.352.3.35	tkSetStringMacro	882
25.352.3.36	tkSetStringMacro	882
25.352.3.37	tkSetStringMacro	882
25.352.3.38	tkSetStringMacro	882
25.352.3.39	tkSetStringMacro	882
25.352.3.40	tkSetStringMacro	882
25.352.3.41	tkSetStringMacro	882
25.352.3.42	tkSetStringMacro	882
25.352.3.43	tkTypeRevisionMacro	882
25.352.4	Member Data Documentation	882
25.352.4.1	Internals	882
25.352.4.2	ReferenceFrameOfReferenceUID	882

25.352.4.3ReferenceSeriesInstanceUID	883
25.352.4.4SeriesInstanceUID	883
25.352.4.5SOPInstanceUID	883
25.352.4.6StructureSetDate	883
25.352.4.7StructureSetLabel	883
25.352.4.8StructureSetName	883
25.352.4.9StructureSetTime	883
25.352.4.10StudyInstanceUID	883
25.353dcm::Waveform Class Reference	883
25.353.1Detailed Description	883
25.353.2Constructor & Destructor Documentation	883
25.353.2.1Waveform	883
25.354dcm::Writer Class Reference	883
25.354.1Detailed Description	886
25.354.2Constructor & Destructor Documentation	887
25.354.2.1Writer	887
25.354.2.2~Writer	887
25.354.3Member Function Documentation	887
25.354.3.1CheckFileMetaInformationOff	887
25.354.3.2CheckFileMetaInformationOn	887
25.354.3.3GetFile	887
25.354.3.4GetStreamPtr	887
25.354.3.5SetCheckFileMetaInformation	887
25.354.3.6SetFile	887
25.354.3.7SetFileName	888
25.354.3.8SetStream	888
25.354.3.9SetWriteDataSetOnly	888
25.354.3.10Write	888
25.354.4Friends And Related Function Documentation	888
25.354.4.1StreamImageWriter	888
25.354.5Member Data Documentation	888
25.354.5.1Ofstream	888
25.354.5.2Stream	888
25.355dcm::XMLDictReader Class Reference	889
25.355.1Detailed Description	890
25.355.2Constructor & Destructor Documentation	890
25.355.2.1XMLDictReader	890

25.355.2.2~XMLDictReader	890
25.355.3Member Function Documentation	890
25.355.3.1CharacterDataHandler	890
25.355.3.2EndElement	890
25.355.3.3GetDict	890
25.355.3.4HandleDescription	890
25.355.3.5HandleEntry	890
25.355.3.6StartElement	890
25.356gdcmm::XMLPrivateDictReader Class Reference	890
25.356.1Detailed Description	891
25.356.2Constructor & Destructor Documentation	892
25.356.2.1XMLPrivateDictReader	892
25.356.2.2~XMLPrivateDictReader	892
25.356.3Member Function Documentation	892
25.356.3.1CharacterDataHandler	892
25.356.3.2EndElement	892
25.356.3.3GetPrivateDict	892
25.356.3.4HandleDescription	892
25.356.3.5HandleEntry	892
25.356.3.6StartElement	892
26 File Documentation	893
26.1 gdcmm2pnm.man File Reference	893
26.2 gdcmm2vtk.man File Reference	893
26.3 gdcmmAAAbortPDU.h File Reference	893
26.4 gdcmmAAAssociateACPDU.h File Reference	894
26.5 gdcmmAAAssociateRJPDU.h File Reference	894
26.6 gdcmmAAAssociateRQPDU.h File Reference	895
26.7 gdcmmAbstractSyntax.h File Reference	896
26.8 gdcmmanon.man File Reference	897
26.9 gdcmmAnonymizeEvent.h File Reference	897
26.10gdcmmAnonymizer.h File Reference	899
26.11gdcmmApplicationContext.h File Reference	899
26.12gdcmmApplicationEntity.h File Reference	900
26.13gdcmmAReleaseRPPDU.h File Reference	901
26.14gdcmmAReleaseRQPDU.h File Reference	902
26.15gdcmmARTIMTimer.h File Reference	903

26.16gdcmlASN1.h File Reference	904
26.17gdcmlAsynchronousOperationsWindowSub.h File Reference	905
26.18gdcmlAttribute.h File Reference	905
26.19gdcmlAudioCodec.h File Reference	907
26.20gdcmlBase64.h File Reference	907
26.21gdcmlBaseCompositeMessage.h File Reference	908
26.22gdcmlBasePDU.h File Reference	909
26.23gdcmlBaseRootQuery.h File Reference	910
26.24gdcmlBasicOffsetTable.h File Reference	912
26.25gdcmlBitmap.h File Reference	913
26.26gdcmlBitmapToBitmapFilter.h File Reference	914
26.27gdcmlBoxRegion.h File Reference	915
26.28gdcmlByteBuffer.h File Reference	915
26.29gdcmlByteSwap.h File Reference	916
26.30gdcmlByteSwapFilter.h File Reference	917
26.31gdcmlByteValue.h File Reference	918
26.32gdcmlCEchoMessages.h File Reference	919
26.33gdcmlCFindMessages.h File Reference	919
26.34gdcmlCMoveMessages.h File Reference	920
26.35gdcmlCodec.h File Reference	921
26.36gdcmlCoder.h File Reference	922
26.37gdcmlCodeString.h File Reference	923
26.38gdcmlCommand.h File Reference	924
26.39gdcmlCommandDataSet.h File Reference	926
26.40gdcmlCompositeMessageFactory.h File Reference	926
26.41gdcmlCompositeNetworkFunctions.h File Reference	927
26.42gdcmlConstCharWrapper.h File Reference	928
26.43gdcmlconv.man File Reference	928
26.44gdcmlCP246ExplicitDataElement.h File Reference	929
26.45gdcmlCryptographicMessageSyntax.h File Reference	929
26.46gdcmlCSAElement.h File Reference	930
26.47gdcmlCSAHeader.h File Reference	931
26.48gdcmlCSAHeaderDict.h File Reference	932
26.49gdcmlCSAHeaderDictEntry.h File Reference	933
26.50gdcmlCStoreMessages.h File Reference	934
26.51gdcmlCurve.h File Reference	935
26.52gdcmlDataElement.h File Reference	936

26.53gdcmDataEvent.h File Reference	938
26.54gdcmDataSet.h File Reference	939
26.55gdcmDataSetEvent.h File Reference	940
26.56gdcmDataSetHelper.h File Reference	940
26.57gdcmDecoder.h File Reference	941
26.58gdcmDefinedTerms.h File Reference	942
26.59gdcmDeflateStream.h File Reference	943
26.60gdcmDefs.h File Reference	943
26.61gdcmDeltaEncodingCodec.h File Reference	945
26.62gdcmDICOmdir.h File Reference	945
26.63gdcmDICOmdirGenerator.h File Reference	946
26.64gdcmDict.h File Reference	947
26.65gdcmDictConverter.h File Reference	949
26.66gdcmDictEntry.h File Reference	949
26.67gdcmDictPrinter.h File Reference	951
26.68gdcmDicts.h File Reference	951
26.69gdcmdiff.man File Reference	952
26.70gdcmDIMSE.h File Reference	953
26.71gdcmDirectionCosines.h File Reference	953
26.72gdcmDirectory.h File Reference	954
26.73gdcmDirectoryHelper.h File Reference	955
26.74gdcmDummyValueGenerator.h File Reference	956
26.75gdcmdump.man File Reference	956
26.76gdcmDumper.h File Reference	957
26.77gdcmElement.h File Reference	957
26.78gdcmEncapsulatedDocument.h File Reference	959
26.79gdcmEnumeratedValues.h File Reference	959
26.80gdcmEvent.h File Reference	960
26.80.1 Macro Definition Documentation	961
26.80.1.1 gdcmEventMacro	961
26.81gdcmException.h File Reference	962
26.82gdcmExplicitDataElement.h File Reference	963
26.83gdcmExplicitImplicitDataElement.h File Reference	963
26.84gdcmFiducials.h File Reference	964
26.85gdcmFile.h File Reference	965
26.86gdcmFileAnonymizer.h File Reference	966
26.87gdcmFileDerivation.h File Reference	966

26.88gdcmlFileExplicitFilter.h File Reference	967
26.89gdcmlFileMetaInformation.h File Reference	968
26.90gdcmlFilename.h File Reference	969
26.91gdcmlFilenameGenerator.h File Reference	969
26.92gdcmlFileSet.h File Reference	970
26.93gdcmlFindPatientRootQuery.h File Reference	972
26.94gdcmlFindStudyRootQuery.h File Reference	973
26.95gdcmlFragment.h File Reference	973
26.96gdcmlgendir.man File Reference	975
26.97gdcmlGlobal.h File Reference	975
26.98gdcmlGroupDict.h File Reference	976
26.99gdcmlIconImage.h File Reference	976
26.100gdcmlIconImageFilter.h File Reference	977
26.101gdcmlIconImageGenerator.h File Reference	978
26.102gdcmlImage.h File Reference	979
26.103gdcmlImageApplyLookupTable.h File Reference	980
26.104gdcmlImageChangePhotometricInterpretation.h File Reference	981
26.105gdcmlImageChangePlanarConfiguration.h File Reference	982
26.106gdcmlImageChangeTransferSyntax.h File Reference	983
26.107gdcmlImageCodec.h File Reference	984
26.108gdcmlImageConverter.h File Reference	985
26.109gdcmlImageFragmentSplitter.h File Reference	986
26.110gdcmlImageHelper.h File Reference	987
26.111gdcmlImageReader.h File Reference	988
26.112gdcmlImageRegionReader.h File Reference	989
26.113gdcmlImageToImageFilter.h File Reference	990
26.114gdcmlImageWriter.h File Reference	991
26.115gdcmlimg.man File Reference	992
26.116gdcmlImplementationClassUIDSub.h File Reference	992
26.117gdcmlImplementationUIDSub.h File Reference	993
26.118gdcmlImplementationVersionNameSub.h File Reference	994
26.119gdcmlImplicitDataElement.h File Reference	996
26.120gdcmlinfo.man File Reference	996
26.121gdcmlIOD.h File Reference	996
26.122gdcmlIODEntry.h File Reference	998
26.123gdcmlIODs.h File Reference	1001
26.124gdcmlIPPSorter.h File Reference	1002

26.125dcmItem.h File Reference	1003
26.126dcmJPEG12Codec.h File Reference	1005
26.127dcmJPEG16Codec.h File Reference	1005
26.128dcmJPEG2000Codec.h File Reference	1006
26.129dcmJPEG8Codec.h File Reference	1007
26.130dcmJPEGCodec.h File Reference	1008
26.131dcmJPEGLSCCodec.h File Reference	1009
26.132dcmKAKADUCodec.h File Reference	1010
26.133dcmLegacyMacro.h File Reference	1011
26.133.1Macro Definition Documentation	1012
26.133.1.1GDCM_LEGACY	1012
26.133.1.2GDCM_LEGACY_BODY	1012
26.133.1.3GDCM_LEGACY_REPLACED_BODY	1012
26.134dcmLO.h File Reference	1012
26.135dcmLookupTable.h File Reference	1013
26.136dcmMacro.h File Reference	1014
26.137dcmMacroEntry.h File Reference	1016
26.137.1Macro Definition Documentation	1018
26.137.1.1GDCMMACROENTRY_H	1018
26.138dcmMacros.h File Reference	1018
26.139dcmMaximumLengthSub.h File Reference	1020
26.140dcmMD5.h File Reference	1021
26.141dcmMediaStorage.h File Reference	1022
26.142dcmMeshPrimitive.h File Reference	1023
26.143dcmModule.h File Reference	1025
26.144dcmModuleEntry.h File Reference	1026
26.145dcmModules.h File Reference	1028
26.146dcmMovePatientRootQuery.h File Reference	1029
26.147dcmMoveStudyRootQuery.h File Reference	1030
26.148dcmNestedModuleEntries.h File Reference	1031
26.149dcmNetworkEvents.h File Reference	1033
26.150dcmNetworkStatelD.h File Reference	1034
26.151dcmObject.h File Reference	1035
26.152dcmOrientation.h File Reference	1036
26.153dcmOverlay.h File Reference	1036
26.154dcmParseException.h File Reference	1037
26.155dcmParser.h File Reference	1039

26.156dcmPatient.h File Reference	1039
26.157dcmPDataTFPDU.h File Reference	1040
26.158dcmPDBElement.h File Reference	1041
26.159dcmPDBHeader.h File Reference	1043
26.160dcmpdf.man File Reference	1043
26.161dcmPDFCodec.h File Reference	1043
26.162dcmPDUFactory.h File Reference	1044
26.163dcmPersonName.h File Reference	1045
26.164dcmPGXCodec.h File Reference	1046
26.165dcmPhotometricInterpretation.h File Reference	1046
26.166dcmPixelFormat.h File Reference	1047
26.167dcmPixmap.h File Reference	1048
26.168dcmPixmapReader.h File Reference	1049
26.169dcmPixmapToPixmapFilter.h File Reference	1051
26.170dcmPixmapWriter.h File Reference	1051
26.171dcmPNMCodec.h File Reference	1052
26.172dcmPreamble.h File Reference	1053
26.173dcmPresentationContext.h File Reference	1054
26.174dcmPresentationContextAC.h File Reference	1055
26.175dcmPresentationContextGenerator.h File Reference	1057
26.176dcmPresentationContextRQ.h File Reference	1057
26.177dcmPresentationDataValue.h File Reference	1058
26.178dcmPrinter.h File Reference	1059
26.179dcmPrivateTag.h File Reference	1060
26.180dcmProgressEvent.h File Reference	1062
26.181dcmPVRGCodec.h File Reference	1062
26.182dcmPythonFilter.h File Reference	1063
26.183dcmQueryBase.h File Reference	1064
26.184dcmQueryFactory.h File Reference	1065
26.185dcmQueryImage.h File Reference	1066
26.186dcmQueryPatient.h File Reference	1067
26.187dcmQuerySeries.h File Reference	1068
26.188dcmQueryStudy.h File Reference	1069
26.189dcmraw.man File Reference	1070
26.190dcmRAWCodec.h File Reference	1070
26.191dcmReader.h File Reference	1071
26.192dcmRegion.h File Reference	1072

26.193dcmRescaler.h File Reference	1073
26.194dcmRLECodec.h File Reference	1074
26.195dcmRoleSelectionSub.h File Reference	1074
26.196dcmScanner.h File Reference	1075
26.197dcmscanner.man File Reference	1076
26.198dcmscu.man File Reference	1076
26.199dcmSegment.h File Reference	1076
26.200dcmSegmentedPaletteColorLookupTable.h File Reference	1078
26.201dcmSegmentHelper.h File Reference	1078
26.202dcmSegmentReader.h File Reference	1080
26.203dcmSegmentWriter.h File Reference	1081
26.204dcmSequenceOfFragments.h File Reference	1082
26.205dcmSequenceOfItems.h File Reference	1083
26.206dcmSerieHelper.h File Reference	1083
26.207dcmSeries.h File Reference	1085
26.208dcmServiceClassApplicationInformation.h File Reference	1086
26.209dcmServiceClassUser.h File Reference	1087
26.210dcmSHA1.h File Reference	1087
26.211dcmSimpleSubjectWatcher.h File Reference	1088
26.212dcmSmartPointer.h File Reference	1089
26.213dcmSOPClassExtendedNegociationSub.h File Reference	1090
26.214dcmSOPClassUIDToIOD.h File Reference	1091
26.215dcmSorter.h File Reference	1092
26.216dcmSpacing.h File Reference	1094
26.217dcmSpectroscopy.h File Reference	1094
26.218dcmSplitMosaicFilter.h File Reference	1095
26.219dcmStaticAssert.h File Reference	1096
26.219.1Macro Definition Documentation	1096
26.219.1.1GDCM_DO_JOIN	1096
26.219.1.2GDCM_DO_JOIN2	1096
26.219.1.3GDCM_JOIN	1096
26.219.1.4GDCM_STATIC_ASSERT	1097
26.220dcmStreamImageReader.h File Reference	1097
26.221dcmStreamImageWriter.h File Reference	1097
26.222dcmString.h File Reference	1098
26.223dcmStringFilter.h File Reference	1099
26.224dcmStudy.h File Reference	1100

26.225	gdcmSubject.h File Reference	1101
26.226	gdcmSurface.h File Reference	1102
26.227	gdcmSurfaceHelper.h File Reference	1103
26.228	gdcmSurfaceReader.h File Reference	1104
26.229	gdcmSurfaceWriter.h File Reference	1105
26.230	gdcmSwapCode.h File Reference	1106
26.231	gdcmSwapper.h File Reference	1107
26.232	gdcmSystem.h File Reference	1108
26.233	gdcmTable.h File Reference	1109
26.234	gdcmTableEntry.h File Reference	1110
26.235	gdcmTableReader.h File Reference	1111
26.236	gdcmTag.h File Reference	1113
26.237	gdcmTagPath.h File Reference	1114
26.238	gdcmTagToVR.h File Reference	1114
26.239	gdcm.tar.man File Reference	1114
26.240	gdcmTerminal.h File Reference	1115
26.241	gdcmTestDriver.h File Reference	1116
26.242	gdcmTesting.h File Reference	1116
26.243	gdcmTrace.h File Reference	1117
26.243.1	Macro Definition Documentation	1118
26.243.1.1	GDGCM_FUNCTION	1118
26.243.1.2	gdcmAssertAlwaysMacro	1118
26.243.1.3	gdcmAssertMacro	1118
26.243.1.4	gdcmDebugMacro	1119
26.243.1.5	gdcmErrorMacro	1119
26.243.1.6	gdcmWarningMacro	1119
26.244	gdcmTransferSyntax.h File Reference	1120
26.245	gdcmTransferSyntaxSub.h File Reference	1121
26.246	gdcmType.h File Reference	1122
26.247	gdcmTypes.h File Reference	1124
26.248	gdcmUIDGenerator.h File Reference	1124
26.249	gdcmUIDs.h File Reference	1125
26.250	gdcmULAction.h File Reference	1126
26.251	gdcmULActionAA.h File Reference	1127
26.252	gdcmULActionAE.h File Reference	1128
26.253	gdcmULActionAR.h File Reference	1129
26.254	gdcmULActionDT.h File Reference	1129

26.255dcmULBasicCallback.h File Reference	1130
26.256dcmULConnection.h File Reference	1131
26.257dcmULConnectionCallback.h File Reference	1132
26.258dcmULConnectionInfo.h File Reference	1133
26.259dcmULConnectionManager.h File Reference	1134
26.260dcmULEvent.h File Reference	1135
26.261dcmULTransitionTable.h File Reference	1136
26.262dcmULWritingCallback.h File Reference	1137
26.263dcmUNExplicitDataElement.h File Reference	1137
26.264dcmUNExplicitImplicitDataElement.h File Reference	1138
26.265dcmUnpacker12Bits.h File Reference	1139
26.266dcmUsage.h File Reference	1139
26.267dcmUserInformation.h File Reference	1142
26.268dcmValidate.h File Reference	1143
26.269dcmValue.h File Reference	1143
26.270dcmValueIO.h File Reference	1144
26.271dcmVersion.h File Reference	1145
26.272dcmviewer.man File Reference	1146
26.273dcmVL.h File Reference	1146
26.274dcmVM.h File Reference	1147
26.274.1Macro Definition Documentation	1148
26.274.1.1TYPETOLENGTH	1148
26.275dcmVR.h File Reference	1148
26.275.1Macro Definition Documentation	1150
26.275.1.1TYPETOENCODING	1150
26.275.1.2VRTemplateCase	1150
26.276dcmVR16ExplicitDataElement.h File Reference	1151
26.277dcmWaveform.h File Reference	1151
26.278dcmWin32.h File Reference	1152
26.278.1Macro Definition Documentation	1152
26.278.1.1GDCM_EXPORT	1152
26.279dcmWriter.h File Reference	1152
26.280dcmXMLDictReader.h File Reference	1153
26.281dcmXMLPrivateDictReader.h File Reference	1154
26.282README.txt File Reference	1154
26.283TestsList.txt File Reference	1154
26.284tkGDCMImageReader.h File Reference	1155

26.284.1Macro Definition Documentation	1156
26.284.1.1VTK_CMYK	1156
26.284.1.2VTK_INVERSE_LUMINANCE	1156
26.284.1.3VTK_LOOKUP_TABLE	1156
26.284.1.4VTK_YBR	1156
26.285tkGDCMImageWriter.h File Reference	1156
26.286tkGDCMMedicalImageProperties.h File Reference	1156
26.287tkGDCMPolyDataReader.h File Reference	1157
26.288tkGDCMPolyDataWriter.h File Reference	1158
26.289tkGDCMTesting.h File Reference	1158
26.290tkGDCMThreadedImageReader.h File Reference	1159
26.291tkGDCMThreadedImageReader2.h File Reference	1160
26.292tkImageColorViewer.h File Reference	1160
26.293tkImageMapToColors16.h File Reference	1161
26.294tkImageMapToWindowLevelColors2.h File Reference	1161
26.295tkImagePlanarComponentsToComponents.h File Reference	1162
26.296tkImageRGBToYBR.h File Reference	1162
26.297tkImageYBRToRGB.h File Reference	1163
26.298tkLookupTable16.h File Reference	1163
26.299tkRTStructSetProperties.h File Reference	1164
27 Example Documentation	1165
27.1 AWTMedical3.java	1165
27.2 BasicAnonymizer.cs	1169
27.3 BasicImageAnonymizer.cs	1170
27.4 CastConvertPhilips.py	1172
27.5 ChangeSequenceUltrasound.cxx	1174
27.6 CheckBigEndianBug.cxx	1175
27.7 ClinicalTrialAnnotate.cxx	1177
27.8 ClinicalTrialIdentificationWorkflow.cs	1178
27.9 CompressImage.cxx	1181
27.10CompressLossyJPEG.cs	1182
27.11Convert16BitsTo8Bits.cxx	1183
27.12ConvertMPL.py	1184
27.13ConvertMultiFrameToSingleFrame.cxx	1185
27.14ConvertNumpy.py	1186
27.15ConvertPIL.py	1187

27.16ConvertRGBToLuminance.cxx	1188
27.17ConvertSingleBitTo8Bits.cxx	1189
27.18ConvertToQImage.cxx	1190
27.19CreateARGBImage.cxx	1192
27.20CreateCMYKImage.cxx	1193
27.21CreateJPIPDataSet.cxx	1194
27.22CreateRAWStorage.py	1195
27.23csa2img.cxx	1197
27.24CStoreQtProgress.cxx	1199
27.25DecompressImage.cs	1201
27.26DecompressImage.java	1202
27.27DecompressImage.py	1203
27.28DecompressImageMultiframe.cs	1204
27.29DecompressJPEGFile.cs	1206
27.30DecompressPixmap.java	1207
27.31DiffFile.cxx	1208
27.32DiscriminateVolume.cxx	1209
27.33DumbAnonymizer.py	1213
27.34DumpADAC.cxx	1214
27.35DumpGEMSMovieGroup.cxx	1219
27.36DumpImageHeaderInfo.cxx	1225
27.37DumpToSQLITE3.cxx	1227
27.38DuplicatePCDE.cxx	1229
27.39ELSCINT1WaveToText.cxx	1231
27.40EncapsulateFileInRawData.cxx	1233
27.41ExtractEncapsulatedFile.cs	1234
27.42ExtractEncryptedContent.cxx	1235
27.43ExtractIconFromFile.cxx	1236
27.44ExtractImageRegion.cs	1237
27.45ExtractImageRegionWithLUT.cs	1239
27.46Extracting_All_Resolution.cxx	1240
27.47ExtractOneFrame.cs	1246
27.48Fake_Image_Using_Stream_Image_Writer.cxx	1247
27.49FileAnonymize.cs	1250
27.50FileAnonymize.java	1251
27.51FindAllPatientName.py	1252
27.52FixBrokenJ2K.cxx	1252

27.53FixCommaBug.py	1254
27.54FixJAIBugJPEGLS.cxx	1255
27.55gdcmmorthoplanes.cxx	1258
27.56gdcmmreslice.cxx	1264
27.57gdcmmrtionplan.cxx	1266
27.58gdcmmrtplan.cxx	1270
27.59gdcmmscene.cxx	1273
27.60gdcmmtexture.cxx	1275
27.61gdcmmvolume.cxx	1277
27.62GenAllVR.cxx	1278
27.63GenerateDICOMDIR.cs	1280
27.64GenerateRTSTRUCT.cxx	1281
27.65GenerateStandardSOPClasses.cxx	1284
27.66GenFakeIdentifyFile.cxx	1285
27.67GenFakeImage.cxx	1287
27.68GenLongSeqs.cxx	1289
27.69GenSeqs.cxx	1290
27.70GetArray.cs	1291
27.71GetJPEGSamplePrecision.cxx	1293
27.72GetPortionCSAHeader.py	1294
27.73GetSequenceUltrasound.cxx	1295
27.74GetSubSequenceData.cxx	1297
27.75headsq2dcm.py	1300
27.76HelloActiviz.cs	1300
27.77HelloActiviz2.cs	1302
27.78HelloActiviz3.cs	1303
27.79HelloActiviz4.cs	1304
27.80HelloActiviz5.cs	1304
27.81HelloSimple.java	1306
27.82HelloVizWorld.cxx	1306
27.83HelloVTKWorld.cs	1307
27.84HelloVTKWorld.java	1308
27.85HelloVTKWorld2.cs	1309
27.86HelloWorld.cxx	1310
27.87HelloWorld.py	1311
27.88iU22tomultisc.cxx	1312
27.89LargeVRDSExplicit.cxx	1313

27.90MagnifyFile.cxx	1315
27.91ManipulateFile.cs	1316
27.92ManipulateFile.py	1317
27.93ManipulateSequence.py	1319
27.94MergeFile.py	1320
27.95MergeTwoFiles.cxx	1321
27.96MetalImageMD5Activiz.cs	1322
27.97MIPViewer.java	1323
27.98MPRViewer.java	1326
27.99MPRViewer2.java	1328
27.10MrProtocol.cxx	1332
27.10NewSequence.cs	1339
27.10NewSequence.py	1340
27.10Offscreenimage.cxx	1341
27.10PatchFile.cxx	1342
27.10PhilipsPrivateRescaleInterceptSlope.py	1343
27.10PlaySound.py	1344
27.10Pmsct_rgb1.cxx	1346
27.10PrivateDict.py	1349
27.10PublicDict.cxx	1349
27.11ReadAndDumpDICOMDIR.cxx	1350
27.11ReadAndDumpDICOMDIR.py	1353
27.11ReadAndPrintAttributes.cxx	1356
27.11ReadExplicitLengthSQIVR.cxx	1357
27.11ReadFiles.java	1358
27.11ReadGEMSSDO.cxx	1359
27.11ReadMultiTimesException.cxx	1362
27.11ReadSeriesIntoVTK.java	1362
27.11ReadUTF8QtDir.cxx	1364
27.11RefCounting.cs	1365
27.12ReformatFile.cs	1366
27.12RemovePrivateTags.py	1367
27.12RescaleImage.cs	1368
27.12Reslicesphere.cxx	1369
27.12ReWriteSCAsMR.py	1377
27.12Re2img.cxx	1378
27.12Constructapp.cxx	1380

27.12	ScanDirectory.cs	1382
27.12	ScanDirectory.java	1383
27.12	ScanDirectory.py	1386
27.13	SendFileSCU.cs	1387
27.13	SimplePrint.cs	1388
27.13	SimplePrintPatientName.cs	1389
27.13	SimpleScanner.cxx	1390
27.13	SortImage.cxx	1391
27.13	SortImage.py	1393
27.13	SortImage2.cs	1393
27.13	StandardizeFiles.cs	1394
27.13	StreamImageReaderTest.cxx	1395
27.13	TestByteSwap.cxx	1399
27.14	TestReader.cxx	1401
27.14	TestReader.py	1402
27.14	Threadgdcmm.cxx	1403
27.14	TraverseModules.cxx	1406
27.14	uid_unique.cxx	1407
27.14	VolumeSorter.cxx	1408
27.14	WriteBuffer.py	1410

Index

1412

Chapter 1

GDCM Documentation

This is the developpers documentation.

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

<http://gdcm.sourceforge.net/2.2/gdcm-2.2.4.pdf>

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

<http://gdcm.sourceforge.net/2.2/gdcm-2.2.4-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 DICOM output filename

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:

```
$ gdcmm2vtk 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 CVS). Only use it,

3.5.1 CONVERT MetaImage (mhd, mha)

```
$ gdcmm2vtk 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

```
$ gdc2vtk inputfile output.mha
```

or

```
$ gdc2vtk inputfile output.mhd
```

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

3.5.3 CONVERT VTI

```
$ gdc2vtk inputfile output.vti
```

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

3.5.4 CONVERT VTK

```
$ gdc2vtk inputfile output.vtk
```

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

3.6 CONVERT DICOM

```
$ gdc2vtk 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

```
$ gdc2vtk input_ybr.dcm output.mhd
$ gdc2vtk --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 gdc2vtk notes

IMPORTANT NOTE: The internal VTK structured will be filled from the input DICOM, and then pass to the output DICOM 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_OPENSSL` 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
```

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 `diff(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_irreversible.dcm
```

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 contain 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

gdcconv(1), **gdcmdump(1)**, **gdcminfo(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 compute 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

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

```
$ gdcmmconv input.dcm output.dcm
```

or if you prefer being explicit:

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

Even though **gdcmmconv** 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 **gdcmmconv** 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 **gdcmmconv** Quality Control section (down below).

5.6 Typical usage

5.6.1 File Meta Header

Running

```
$ gdcmmconv input.dcm output.dcm
```

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

```
$ gdcmmconv --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 `gdcmmData`.

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:

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

5.6.3 Compressing to lossless JPEG

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

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

5.6.4 Compressing to lossy JPEG

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

```
$ gdcconv --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:

```
$ gdcconv --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:

```
$ gdcconv --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:

```
$ gdcconv --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:

```
$ gdcconv --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:

```
$ gdcconv --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):

```
$ gdcconv --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

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

5.6.13 Compressing an uncompressed Icon

By default when compressing a DICOM Image file, `gdcconv` 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:

```
$ gdcconv --jpeg gdcData/simpleImageWithIcon.dcm uncompressed_icon.dcm
```

In the following example we will explicitly compress the Icon Image Sequence Pixel Data attribute. 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:

```
$ gdcconv --jpeg --compress-icon gdcData/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:

```
$ gdcconv --raw --generate-icon gdcData/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:

```
$ gdcconv --raw --generate-icon --icon-minmax 0,192 gdcData/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 :

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

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

```
$ gdcconv --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://gdc.sourceforge.net/wiki/index.php/Color_Space_Transformations

5.8 Quality Control

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

You can use 3rd party tool to check the output of gdcconv 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 gdcconv.

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

For reference, gdcconv -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 %!gdcm dump -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), **gdcmmraw(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 file: 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 diff 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,Philips)
-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_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-ig-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\0\255] # 16
SP_coil_t_pos [255\128\255\0\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/1,1 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), gdcmrw(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 `gdcconv` command line tool:

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

8.7 SEE ALSO

`gdcconv(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.

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
```

```
file-out    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) gdcming

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 $[0,100] \times [0,100]$ of a DICOM image simply do:

```
$ gdcimg --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 recompressed.

9.6.2 Convert RAW to DICOM

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

```
$ gdcimg --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:

```
$ gdcimg --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`

```
$ gdcimg --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 endianness manually:

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

or

```
$ gdcimg --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)

```
$ gdcimg -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)

```
$ gdcimg --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)

```
$ gdcimg -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)

```
$ gdcimg -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.

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

9.7 Multiple Files

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

```
$ gdcimg 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:

```
$ gdcimg --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)`, `gdcmdump(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

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

10.4.2 general options

<code>-h --help</code>	print this help text and exit
<code>-v --version</code>	print version information and exit
<code>-V --verbose</code>	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 gdcminfo

Using data from gdcminfo:

```
$ gdcminfo gdcminfo/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 David 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:

```
$ gdcmmconv --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 or not. 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 gdcmmData/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 PDF to PDF/DICOM.

11.1 SYNOPSIS

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

11.2 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.

11.3 PARAMETERS

file-in PDF input filename

file-out DICOM output filename

11.4 options

11.4.1 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
```

11.5 Usage Example

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

To re-extract the encapsulated pdf file:

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

11.6 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:   pdfTeX-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 [pdfTeX-1.21a] # 14, 1 SoftwareVersions
(0020,000d) UI [1.2.826.0.1.3680043.2.1143.1868121832223417351654232480755123133] # 64, 1 StudyInstanceUID
(0020,000e) UI [1.2.826.0.1.3680043.2.1143.1330099150825746617507846107663964311] # 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
(ffff,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 MIMETimeTypeOfEncapsulatedDocument

```

```

$ stat gdc.m.pdf
  File: `gdc.m.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 gdc.mpdf 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.

11.7 SEE ALSO

gdc.mconv(1), gdc.mraw(1), pdfinfo(1)

11.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 12

Extract Data Element Value Field.

12.1 SYNOPSIS

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

12.2 DESCRIPTION

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

12.3 PARAMETERS

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

```
file-out    output filename
```

12.4 options

12.4.1 PARAMETERS

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

12.4.2 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.
```

12.4.3 general options

```
-h    --help
```



```
-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
```

12.6 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.

12.7 SEE ALSO

gdcmdump(1), **gdcmrw(1)**

12.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 13

Scan a directory containing DICOM files.

13.1 SYNOPSIS

```
gdcmscanner [options] directory
```

13.2 DESCRIPTION

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

13.2.1 PARAMETERS

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

13.2.2 options

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

13.2.3 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
```

13.3 Typical usage

13.4 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
```

13.5 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.

13.6 SEE ALSO

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

13.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 14

Tool to execute a DICOM Query/Retrieve operation

14.1 SYNOPSIS

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

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

14.2 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).

14.3 PARAMETERS

14.4 options

14.4.1 options

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

14.4.2 mode options

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

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

14.4.3 C-STORE options

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

14.4.4 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
```

14.4.5 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
```

14.4.6 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
```

14.4.7 environment variable

```
GDCM_ROOT_UID Root UID
```

14.5 C-ECHO usage

gdcm SCU 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:

```
$ gdcm SCU dicom.example.com
```

or if you prefer being explicit:

```
$ gdcm SCU --echo dicom.example.com 104
```

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

```
$ gdcm SCU --echo dicom.example.com 11112 --call SERVSCP
```

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

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

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

```
$ gdcm SCU www.dicomserver.co.uk 11112
```

14.6 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`

```
$ gdcm SCU --store dicom.example.com 104 myfile.dcm
```

or if you prefer being explicit:

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

You can even send multiple files using the same association:

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

14.7 C-FIND usage

gdcm SCU 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:

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

We also support a DCMTK compatible convention:

```
$ gdcm SCU --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:

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

14.8 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="ABCD1234"
```

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 your 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).

14.9 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*"
```

14.10 Debugging

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

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 its 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).

14.11 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.

14.12 C-STORE Warnings

When constructing a C-STORE operation, `gdcm SCU` will always use the Media Storage 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
```

14.13 C-MOVE Warnings

At the moment `gdcm SCU` only supports non-compressed transfer syntax. It will always request DataSet using Implicit VR Little Endian Transfer Syntax during a C-MOVE operation.

14.14 C-FIND IMAGE level (Composite Object Instance)

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

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

instead of the more explicit form

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

This would also be equivalent to:

```
$ find SCU --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
```

14.15 Storing the Query

It is also possible to store the query:

```
gdcm SCU --find --patient --patientroot dicom.example.com 11112 --key 10,20="*" --key 10,10 --store-query query.dcm
```

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 `gdcm SCU` to be UTF-8.

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

```
$ gdcm SCU --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.

14.16 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>

14.17 SEE ALSO

`gdcmconv(1)`

14.18 COPYRIGHT

Copyright Insight Software Consortium

Chapter 15

Concatenate/Extract DICOM files.

15.1 SYNOPSIS

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

15.2 DESCRIPTION

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

15.3 PARAMETERS

file-in DICOM input filename

file-out DICOM output filename

15.4 options

15.4.1 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.
```

15.4.2 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
```

15.4.3 environment variable

GDCM_ROOT_UID Root UID

15.5 Typical usage

15.5.1 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)  
\&...
```

15.6 SEE ALSO

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

15.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 16

Simple DICOM viewer.

16.1 SYNOPSIS

```
gdcviewer [options] file-in
```

16.2 DESCRIPTION

The **gdcviewer** is a simple tool that show how to use [vtkGDCMImageReader](#). The class that use gdc to make a layer to VTK. **gdcviewer** 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 5.8, but only with VTK 5.2 (or above) can only play with the widgets (as described below).

16.3 PARAMETERS

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

16.4 options

16.4.1 options

<code>--force-rescale</code>	force rescale (advanced users)
<code>--force-spacing</code>	force spacing (advanced users)
<code>-r --recursive</code>	Recursively descend directory

16.4.2 general options

<code>-h</code>	<code>--help</code>	print this help text and exit
<code>-v</code>	<code>--version</code>	print version information and exit
<code>-V</code>	<code>--verbose</code>	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
```

16.5 Typical usage

16.6 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.

16.7 Wiki Link

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

16.8 SEE ALSO

`gdcmdump(1)`, `gdcm2vtk(1)`

16.9 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

Chapter 17

Todo List

Class [gdcm::CSAHeader](#)

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

Class [gdcm::Overlay](#)

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

Class [gdcm::SequenceOfFragments](#)

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

Class [gdcm::TransferSyntax](#)

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

Member [gdcm::UIDGenerator::IsValid](#) (const char *uid)

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

Chapter 18

Deprecated List

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

Member `gdcm::DataElement::GetSequenceOfItems` () `const`

Replaced by `DataElement::GetValueAsSQ()` as of GDCM 2.2.

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 19

Bug List

Class `gdcm::DICOMDIRGenerator`

: There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the `gdcm::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:

Chapter 20

Namespace Index

20.1 Namespace List

Here is a list of all namespaces with brief descriptions:

gdc	103
gdc::network	124
gdc::SegmentHelper	130
gdc::terminal	
Class for Terminal Allow one to print in color in a shell	130

Chapter 21

Hierarchical Index

21.1 Class Hierarchy

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

gdcn::network::AbstractSyntax	144
gdcn::network::ApplicationContext	154
gdcn::ApplicationEntity	155
gdcn::network::ARTIMTimer	160
gdcn::ASN1	161
gdcn::network::AsynchronousOperationsWindowSub	162
gdcn::Attribute< Group, Element, TVR, TVM >	163
gdcn::Attribute< Group, Element, TVR, VM::VM1 >	170
gdcn::Attribute< Group, Element, TVR, VM::VM1_n >	177
gdcn::Attribute< Group, Element, TVR, VM::VM1_3 >	175
gdcn::Attribute< Group, Element, TVR, VM::VM1_8 >	176
gdcn::Attribute< Group, Element, TVR, VM::VM2_n >	183
gdcn::Attribute< Group, Element, TVR, VM::VM2_2n >	182
gdcn::Attribute< Group, Element, TVR, VM::VM3_n >	186
gdcn::Attribute< Group, Element, TVR, VM::VM3_3n >	185
gdcn::Base64	190
gdcn::network::BaseCompositeMessage	192
gdcn::network::CEchoRQ	225
gdcn::network::CEchoRSP	227
gdcn::network::CFindCancelRQ	229
gdcn::network::CFindRQ	230
gdcn::network::CFindRSP	231
gdcn::network::CMoveCancelRq	232
gdcn::network::CMoveRQ	234
gdcn::network::CMoveRSP	235
gdcn::network::CStoreRQ	267
gdcn::network::CStoreRSP	268
gdcn::network::BasePDU	194
gdcn::network::AAabortPDU	133
gdcn::network::AAssociateACPDU	135
gdcn::network::AAssociateRJPDU	138
gdcn::network::AAssociateRQPDU	139
gdcn::network::AReleaseRPPDU	157

gdcmm::network::AReleaseRQPDU	158
gdcmm::network::PDataTFPDU	529
std::basic_string< Char >	
std::string	
gdcmm::String< TDelimiter, TMaxLength, TPadChar >	678
gdcmm::SegmentHelper::BasicCodedEntry	200
gdcmm::BitmapToBitmapFilter	213
gdcmm::PixmapToPixmapFilter	556
gdcmm::ImageToImageFilter	431
gdcmm::ImageApplyLookupTable	400
gdcmm::ImageChangePhotometricInterpretation	402
gdcmm::ImageChangePlanarConfiguration	406
gdcmm::ImageChangeTransferSyntax	409
gdcmm::ImageFragmentSplitter	419
gdcmm::ByteBuffer	218
gdcmm::ByteSwap< T >	219
gdcmm::ByteSwapFilter	220
gdcmm::network::CFind	228
gdcmm::Coder	237
gdcmm::Codec	236
gdcmm::AudioCodec	188
gdcmm::ImageCodec	413
gdcmm::DeltaEncodingCodec	298
gdcmm::JPEG2000Codec	459
gdcmm::JPEGCodec	464
gdcmm::JPEG12Codec	455
gdcmm::JPEG16Codec	457
gdcmm::JPEG8Codec	462
gdcmm::JPEGLSCoec	468
gdcmm::KAKADUCoec	471
gdcmm::PGXCoec	540
gdcmm::PNMCoec	561
gdcmm::PVRGCoec	583
gdcmm::RAWCoec	597
gdcmm::RLECoec	609
gdcmm::PDFCoec	536
gdcmm::CodeString	239
gdcmm::network::CompositeMessageFactory	245
gdcmm::CompositeNetworkFunctions	246
gdcmm::ConstCharWrapper	249
gdcmm::CryptographicMessageSyntax	252
gdcmm::CSAElement	253
gdcmm::CSAHeader	258
gdcmm::CSAHeaderDict	262
gdcmm::CSAHeaderDictEntry	264
gdcmm::DataElement	272
gdcmm::CP246ExplicitDataElement	250
gdcmm::ExplicitDataElement	353
gdcmm::ExplicitImplicitDataElement	354
gdcmm::Fragment	384
gdcmm::BasicOffsetTable	203
gdcmm::ImplicitDataElement	438
gdcmm::Item	450

gdcmm::UNExplicitDataElement	804
gdcmm::UNExplicitImplicitDataElement	806
gdcmm::VR16ExplicitDataElement	829
gdcmm::DataSet	284
gdcmm::CommandDataSet	243
gdcmm::FileMetaInformation	367
gdcmm::DataSetHelper	293
gdcmm::Decoder	294
gdcmm::Codec	236
gdcmm::DefinedTerms	295
gdcmm::Defs	296
gdcmm::DICOMDIR	300
gdcmm::DICOMDIRGenerator	300
gdcmm::Dict	303
gdcmm::DictConverter	305
gdcmm::DictEntry	307
gdcmm::Dicts	312
gdcmm::network::DIMSE	314
gdcmm::DirectionCosines	316
gdcmm::Directory	318
gdcmm::DirectoryHelper	320
gdcmm::DummyValueGenerator	322
gdcmm::Element< TVR, TVM >	325
gdcmm::Element< TVR, VM::VM1_n >	329
gdcmm::Element< TVR, VM::VM1_2 >	328
gdcmm::Element< TVR, VM::VM2_n >	334
gdcmm::Element< TVR, VM::VM2_2n >	332
gdcmm::Element< TVR, VM::VM3_n >	337
gdcmm::Element< TVR, VM::VM3_3n >	335
gdcmm::Element< VR::AS, VM::VM5 >	338
gdcmm::Element< VR::OB, VM::VM1_n >	325
gdcmm::Element< VR::OB, VM::VM1 >	339
gdcmm::Element< VR::OW, VM::VM1_n >	325
gdcmm::Element< VR::OW, VM::VM1 >	340
gdcmm::ElementDisableCombinations< TVR, TVM >	342
gdcmm::ElementDisableCombinations< VR::OB, VM::VM1_n >	343
gdcmm::ElementDisableCombinations< VR::OW, VM::VM1_n >	343
gdcmm::EncapsulatedDocument	343
gdcmm::EncodingImplementation< T >	344
gdcmm::EncodingImplementation< VR::VRASCII >	344
gdcmm::EncodingImplementation< VR::VRBINARY >	345
gdcmm::EnumeratedValues	347
gdcmm::Event	348
gdcmm::AnyEvent	152
gdcmm::AbortEvent	143
gdcmm::AnonymizeEvent	145
gdcmm::DataEvent	282
gdcmm::DataSetEvent	291
gdcmm::EndEvent	346
gdcmm::ExitEvent	351
gdcmm::InitializeEvent	440
gdcmm::IterationEvent	453

gdcmm::ModifiedEvent	499
gdcmm::ProgressEvent	580
gdcmm::StartEvent	668
gdcmm::UserEvent	810
gdcmm::NoEvent	514
std::exception	
gdcmm::CSAHeaderDictException	266
gdcmm::DataElementException	281
gdcmm::Exception	350
gdcmm::ParseException	525
gdcmm::Fiducials	356
gdcmm::FileDerivation	363
gdcmm::FileExplicitFilter	365
gdcmm::Filename	373
gdcmm::FilenameGenerator	375
gdcmm::FileSet	377
gdcmm::Global	387
gdcmm::GroupDict	389
gdcmm::IconImageFilter	391
gdcmm::IconImageGenerator	393
gdcmm::ignore_char	395
gdcmm::ImageConverter	418
gdcmm::ImageHelper	421
gdcmm::network::ImplementationClassUIDSub	436
gdcmm::network::ImplementationUIDSub	436
gdcmm::network::ImplementationVersionNameSub	437
gdcmm::IOD	441
gdcmm::IODEntry	443
gdcmm::IODs	445
gdcmm::Scanner::ltstr	480
gdcmm::Macro	480
gdcmm::Macros	482
gdcmm::network::MaximumLengthSub	483
gdcmm::MD5	484
gdcmm::MediaStorage	485
gdcmm::Module	501
gdcmm::ModuleEntry	503
gdcmm::NestedModuleEntries	511
gdcmm::Modules	505
gdcmm::Object	514
gdcmm::BaseRootQuery	196
gdcmm::FindPatientRootQuery	380
gdcmm::FindStudyRootQuery	382
gdcmm::MovePatientRootQuery	507
gdcmm::MoveStudyRootQuery	509
gdcmm::Bitmap	205
gdcmm::Pixmap	549
gdcmm::Image	396
gdcmm::Curve	269
gdcmm::File	357
gdcmm::FileWithName	378
gdcmm::LookupTable	476
gdcmm::SegmentedPaletteColorLookupTable	625

gdcmmesh::MeshPrimitive	496
gdcmmesh::Overlay	519
gdcmmesh::Segment	620
gdcmmesh::Subject	685
gdcmmesh::Anonymizer	148
gdcmmesh::Command	241
gdcmmesh::MemberCommand< T >	492
gdcmmesh::SimpleMemberCommand< T >	652
gdcmmesh::FileAnonymizer	360
gdcmmesh::network::ULConnectionManager	798
gdcmmesh::Scanner	614
gdcmmesh::ServiceClassUser	646
gdcmmesh::Surface	687
gdcmmesh::Value	814
gdcmmesh::ByteValue	221
gdcmmesh::SequenceOfFragments	631
gdcmmesh::SequenceOfItems	636
gdcmmesh::Orientation	517
gdcmmesh::Parser	527
gdcmmesh::Patient	529
gdcmmesh::PDBelement	532
gdcmmesh::PDBHeader	534
gdcmmesh::network::PDUFactory	537
gdcmmesh::PersonName	539
gdcmmesh::PhotometricInterpretation	542
gdcmmesh::PixelFormat	544
gdcmmesh::Preamble	564
gdcmmesh::PresentationContext	565
gdcmmesh::network::PresentationContextAC	567
gdcmmesh::PresentationContextGenerator	568
gdcmmesh::network::PresentationContextRQ	570
gdcmmesh::network::PresentationDataValue	572
gdcmmesh::Printer	574
gdcmmesh::DictPrinter	310
gdcmmesh::Dumper	323
gdcmmesh::PrivateDict	577
gdcmmesh::PythonFilter	585
gdcmmesh::QueryBase	586
gdcmmesh::QueryImage	589
gdcmmesh::QueryPatient	591
gdcmmesh::QuerySeries	593
gdcmmesh::QueryStudy	595
gdcmmesh::QueryFactory	588
gdcmmesh::Reader	599
gdcmmesh::PixmapReader	552
gdcmmesh::ImageReader	424
gdcmmesh::ImageRegionReader	428
gdcmmesh::SegmentReader	626
gdcmmesh::SurfaceReader	696
gdcmmesh::Region	604
gdcmmesh::BoxRegion	215
gdcmmesh::Rescaler	606

gdcmm::network::RoleSelectionSub	612
gdcmm::SerieHelper::Rule	613
gdcmm::SerieHelper	642
gdcmm::Series	644
gdcmm::network::ServiceClassApplicationInformation	645
gdcmm::SHA1	651
gdcmm::SimpleSubjectWatcher	655
gdcmm::SmartPointer< ObjectType >	657
gdcmm::SmartPointer< gdcmm::Bitmap >	657
gdcmm::SmartPointer< gdcmm::File >	657
gdcmm::SmartPointer< gdcmm::gdcmm::Subject >	657
gdcmm::SmartPointer< gdcmm::Image >	657
gdcmm::SmartPointer< gdcmm::MemberCommand >	657
gdcmm::SmartPointer< gdcmm::MeshPrimitive >	657
gdcmm::SmartPointer< gdcmm::Pixmap >	657
gdcmm::SmartPointer< gdcmm::SimpleMemberCommand >	657
gdcmm::SmartPointer< LookupTable >	657
gdcmm::SmartPointer< Segment >	657
gdcmm::SmartPointer< Surface >	657
gdcmm::SmartPointer< Value >	657
gdcmm::network::SOPClassExtendedNegociationSub	659
gdcmm::SOPClassUIDToIOD	660
gdcmm::Sorter	661
gdcmm::IPPSorter	446
gdcmm::Spacing	665
gdcmm::Spectroscopy	667
gdcmm::SplitMosaicFilter	667
gdcmm::static_assert_test< x >	670
gdcmm::STATIC_ASSERTION_FAILURE< x >	670
gdcmm::STATIC_ASSERTION_FAILURE< true >	670
gdcmm::StreamImageReader	670
gdcmm::StreamImageWriter	673
String<'\', 64 >	
gdcmm::LO	473
gdcmm::StringFilter	682
gdcmm::Study	684
gdcmm::SurfaceHelper	694
gdcmm::SwapCode	700
gdcmm::SwapperDoOp	702
gdcmm::SwapperNoOp	703
gdcmm::System	703
gdcmm::Table	707
gdcmm::TableEntry	708
gdcmm::TableReader	709
gdcmm::XMLDictReader	889
gdcmm::XMLPrivateDictReader	890
gdcmm::network::TableRow	711
gdcmm::Tag	712
gdcmm::PrivateTag	579
gdcmm::TagPath	718
gdcmm::Testing	720
gdcmm::Trace	724
gdcmm::TransferSyntax	727

gdcm::network::TransferSyntaxSub	731
gdcm::network::Transition	732
gdcm::Type	733
gdcm::UI	735
gdcm::UIDGenerator	736
gdcm::UIDs	737
gdcm::network::ULAction	756
gdcm::network::ULActionAA1	759
gdcm::network::ULActionAA2	760
gdcm::network::ULActionAA3	761
gdcm::network::ULActionAA4	762
gdcm::network::ULActionAA5	763
gdcm::network::ULActionAA6	764
gdcm::network::ULActionAA7	766
gdcm::network::ULActionAA8	767
gdcm::network::ULActionAE1	768
gdcm::network::ULActionAE2	769
gdcm::network::ULActionAE3	770
gdcm::network::ULActionAE4	771
gdcm::network::ULActionAE5	773
gdcm::network::ULActionAE6	774
gdcm::network::ULActionAE7	775
gdcm::network::ULActionAE8	776
gdcm::network::ULActionAR1	777
gdcm::network::ULActionAR10	778
gdcm::network::ULActionAR2	780
gdcm::network::ULActionAR3	781
gdcm::network::ULActionAR4	782
gdcm::network::ULActionAR5	783
gdcm::network::ULActionAR6	784
gdcm::network::ULActionAR7	785
gdcm::network::ULActionAR8	787
gdcm::network::ULActionAR9	788
gdcm::network::ULActionDT1	789
gdcm::network::ULActionDT2	790
gdcm::network::ULConnection	793
gdcm::network::ULConnectionCallback	795
gdcm::network::ULBasicCallback	791
gdcm::network::ULWritingCallback	802
gdcm::network::ULConnectionInfo	797
gdcm::network::ULEvent	801
gdcm::network::ULTransitionTable	802
gdcm::Unpacker12Bits	808
gdcm::Usage	809
gdcm::network::UserInformation	812
gdcm::Validate	813
gdcm::ValueIO< TDE, TSwap, TType >	816
gdcm::Version	817
gdcm::VL	818
gdcm::VM	820
gdcm::VMToLength< T >	824
gdcm::VR	824
gdcm::VRToEncoding< T >	831
gdcm::VRToType< T >	831

gdcm::VRToType< TVR >	831
gdcm::VRVLSIZE< T >	832
gdcm::VRVLSIZE< 0 >	832
gdcm::VRVLSIZE< 1 >	832
vtkImageAlgorithm	
vtkImagePlanarComponentsToComponents	871
vtkImageMapToColors	
vtkImageMapToWindowLevelColors2	869
vtkImageWriter	
vtkGDCMImageWriter	839
vtkLookupTable	
vtkLookupTable16	876
vtkMedicalImageProperties	
vtkGDCMMedicalImageProperties	843
vtkMedicalImageReader2	
vtkGDCMImageReader	833
vtkGDCMThreadedImageReader	853
vtkObject	
vtkGDCMTesting	851
vtkImageColorViewer	859
vtkRTStructSetProperties	878
vtkPolyDataAlgorithm	
vtkGDCMPolyDataReader	845
vtkPolyDataWriter	
vtkGDCMPolyDataWriter	848
vtkThreadedImageAlgorithm	
vtkGDCMThreadedImageReader2	856
vtkImageMapToColors16	866
vtkImageRGBToYBR	873
vtkImageYBRToRGB	874
gdcm::Waveform	883
gdcm::Writer	883
gdcm::PixmapWriter	558
gdcm::ImageWriter	433
gdcm::SegmentWriter	629
gdcm::SurfaceWriter	698

Chapter 22

Class Index

22.1 Class List

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

gdcn::network::AAAbortPDU	
AAAbortPDU Table 9-26 A-ABORT PDU FIELDS	133
gdcn::network::AAssociateACPDU	
AAssociateACPDU Table 9-17 ASSOCIATE-AC PDU fields	135
gdcn::network::AAssociateRJPDU	
AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS	138
gdcn::network::AAssociateRQPDU	
AAssociateRQPDU Table 9-11 ASSOCIATE-RQ PDU fields	139
gdcn::AbortEvent	143
gdcn::network::AbstractSyntax	
AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS	144
gdcn::AnonymizeEvent	
AnonymizeEvent Special type of event triggered during the Anonymization process	145
gdcn::Anonymizer	
Anonymizer This class is a multi purpose anonymizer. It can work in 2 mode:	148
gdcn::AnyEvent	152
gdcn::network::ApplicationContext	
ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Con- text can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)	154
gdcn::ApplicationEntity	
ApplicationEntity	155
gdcn::network::AReleaseRPPDU	
AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields	157
gdcn::network::AReleaseRQPDU	
AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS	158
gdcn::network::ARTIMTimer	
ARTIMTimer This file contains the code for the ARTIM timer	160
gdcn::ASN1	
Class for ASN1	161
gdcn::network::AsynchronousOperationsWindowSub	
AsynchronousOperationsWindowSub PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WIND- OW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	162

gdcm::Attribute< Group, Element, TVR, TVM >	
Attribute class This class use template metaprograming tricks to let the user know when the template instantiation does not match the public dictionary	163
gdcm::Attribute< Group, Element, TVR, VM::VM1 >	170
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >	175
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >	176
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >	177
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >	182
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >	183
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >	185
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >	186
gdcm::AudioCodec	
AudioCodec	188
gdcm::Base64	
Class for Base64	190
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	192
gdcm::network::BasePDU	
BasePDU base class for PDUs	194
gdcm::BaseRootQuery	
BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root	196
gdcm::SegmentHelper::BasicCodedEntry	
This structure defines a basic coded entry with all of its attributes	200
gdcm::BasicOffsetTable	
Class to represent a BasicOffsetTable	203
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)	205
gdcm::BitmapToBitmapFilter	
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image	213
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)	215
gdcm::ByteBuffer	
ByteBuffer	218
gdcm::ByteSwap< T >	
ByteSwap	219
gdcm::ByteSwapFilter	
ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??	220
gdcm::ByteValue	
Class to represent binary value (array of bytes)	221
gdcm::network::CEchoRQ	
CEchoRQ this file defines the messages for the cecho action	225
gdcm::network::CEchoRSP	
CEchoRSP this file defines the messages for the cecho action	227
gdcm::network::CFind	228
gdcm::network::CFindCancelRQ	
CFindCancelRQ this file defines the messages for the cfind action	229
gdcm::network::CFindRQ	
CFindRQ this file defines the messages for the cfind action	230

gdcmm::network::CFindRSP	
CFindRSP this file defines the messages for the cfind action	231
gdcmm::network::CMoveCancelRq	232
gdcmm::network::CMoveRQ	
CMoveRQ this file defines the messages for the cmove action	234
gdcmm::network::CMoveRSP	
CMoveRSP this file defines the messages for the cmove action	235
gdcmm::Codec	
Codec class	236
gdcmm::Coder	
Coder	237
gdcmm::CodeString	
CodeString This is an implementation of DICOM VR: CS The ctor will properly Trim so that operator== is correct	239
gdcmm::Command	
Command superclass for callback/observer methods	241
gdcmm::CommandDataSet	
Class to represent a Command DataSet	243
gdcmm::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)	245
gdcmm::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:	246
gdcmm::ConstCharWrapper	
Do not use me	249
gdcmm::CP246ExplicitDataElement	
Class to read/write a DataElement as CP246Explicit Data Element	250
gdcmm::CryptographicMessageSyntax	
Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities	252
gdcmm::CSAElement	
Class to represent a CSA Element	253
gdcmm::CSAHeader	
Class for CSAHeader	258
gdcmm::CSAHeaderDict	
Class to represent a map of CSAHeaderDictEntry	262
gdcmm::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 gdcmm::Tag to the needed information	264
gdcmm::CSAHeaderDictException	266
gdcmm::network::CStoreRQ	
CStoreRQ this file defines the messages for the cecho action	267

gdcm::network::CStoreRSP	
CStoreRSP this file defines the messages for the cecho action	268
gdcm::Curve	
Curve class to handle element 50xx,3000 Curve Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004	269
gdcm::DataElement	
Class to represent a Data Element either Implicit or Explicit	272
gdcm::DataElementException	281
gdcm::DataEvent	
DataEvent	282
gdcm::DataSet	
Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information Object	284
gdcm::DataSetEvent	
DataSetEvent Special type of event triggered during the DataSet store/move process	291
gdcm::DataSetHelper	
DataSetHelper (internal class, not intended for user level)	293
gdcm::Decoder	
Decoder	294
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	295
gdcm::Defs	
FIXME I do not like the name ' Defs '	296
gdcm::DeltaEncodingCodec	
DeltaEncodingCodec compression used by some private vendor	298
gdcm::DICOMDIR	
DICOMDIR class	300
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	300
gdcm::Dict	
Class to represent a map of DictEntry	303
gdcm::DictConverter	
Class to convert a .dic file into something else:	305
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	307
gdcm::DictPrinter	
DictPrinter class	310
gdcm::Dicts	
Class to manipulate the sum of knowledge (all the dict user load)	312
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)	314
gdcm::DirectionCosines	
Class to handle DirectionCosines	316

gdcm::Directory	
Class for manipulation directories	318
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	320
gdcm::DummyValueGenerator	
Class for generating dummy value	322
gdcm::Dumper	
Codec class	323
gdcm::Element< TVR, TVM >	
Element class	325
gdcm::Element< TVR, VM::VM1_2 >	328
gdcm::Element< TVR, VM::VM1_n >	329
gdcm::Element< TVR, VM::VM2_2n >	332
gdcm::Element< TVR, VM::VM2_n >	334
gdcm::Element< TVR, VM::VM3_3n >	335
gdcm::Element< TVR, VM::VM3_n >	337
gdcm::Element< VR::AS, VM::VM5 >	338
gdcm::Element< VR::OB, VM::VM1 >	339
gdcm::Element< VR::OW, VM::VM1 >	340
gdcm::ElementDisableCombinations< TVR, TVM >	
A class which is used to produce compile errors for an invalid combination of template parameters	342
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n >	343
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n >	343
gdcm::EncapsulatedDocument	
EncapsulatedDocument	343
gdcm::EncodingImplementation< T >	
EncodingImplementation	344
gdcm::EncodingImplementation< VR::VRASCII >	344
gdcm::EncodingImplementation< VR::VRBINARY >	345
gdcm::EndEvent	346
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:	347
gdcm::Event	
Superclass for callback/observer methods	348
gdcm::Exception	
Exception	350
gdcm::ExitEvent	351
gdcm::ExplicitDataElement	
Class to read/write a DataElement as Explicit Data Element	353
gdcm::ExplicitImplicitDataElement	
Class to read/write a DataElement as ExplicitImplicit Data Element	354
gdcm::Fiducials	
Fiducials	356
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 be written, read and/or deleted	357

gdcm::FileAnonymizer	
FileAnonymizer	360
gdcm::FileDerivation	
FileDerivation class See PS 3.16 - 2008 For the list of Code Value that can be used for in Derivation	
Code Sequence	363
gdcm::FileExplicitFilter	
FileExplicitFilter class After changing a file from Implicit to Explicit representation (see ImageChange-TransferSyntax) one operation is to make sure the VR of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit representation, the VR is not stored directly in the file	365
gdcm::FileMetaInformation	
Class to represent a File Meta Information	367
gdcm::Filename	
Class to manipulate file name's	373
gdcm::FilenameGenerator	
FilenameGenerator	375
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	377
gdcm::FileWithName	
FileWithName	378
gdcm::FindPatientRootQuery	
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root	380
gdcm::FindStudyRootQuery	
FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root . .	382
gdcm::Fragment	
Class to represent a Fragment	384
gdcm::Global	
Global	387
gdcm::GroupDict	
Class to represent the mapping from group number to its abbreviation and name	389
gdcm::IconImageFilter	
IconImageFilter This filter will extract icons from a gdcm::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	391
gdcm::IconImageGenerator	
IconImageGenerator This filter will generate a valid Icon from the Pixel Data element (an instance of gdcm::Pixmap). To generate a valid Icon, one is only allowed the following Photometric Interpretation: 393	
gdcm::ignore_char	395
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:	396
gdcm::ImageApplyLookupTable	
ImageApplyLookupTable class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a PhotometricInterpretation =RGB image	400
gdcm::ImageChangePhotometricInterpretation	
ImageChangePhotometricInterpretation class Class to change the Photometric Interpretation of an input DICOM	402
gdcm::ImageChangePlanarConfiguration	
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0	406
gdcm::ImageChangeTransferSyntax	
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM	409

gdcm::ImageCodec	
ImageCodec	413
gdcm::ImageConverter	
Image Converter	418
gdcm::ImageFragmentSplitter	
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments	419
gdcm::ImageHelper	
ImageHelper (internal class, not intended for user level)	421
gdcm::ImageReader	
ImageReader	424
gdcm::ImageRegionReader	
ImageRegionReader	428
gdcm::ImageToImageFilter	
ImageToImageFilter class Super class for all filter taking an image and producing an output image	431
gdcm::ImageWriter	
ImageWriter	433
gdcm::network::ImplementationClassUIDSub	
ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	436
gdcm::network::ImplementationUIDSub	
ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	436
gdcm::network::ImplementationVersionNameSub	
ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	437
gdcm::ImplicitDataElement	
Class to represent an <i>Implicit VR</i> Data Element	438
gdcm::InitializeEvent	440
gdcm::IOD	
Class for representing a IOD	441
gdcm::IODEntry	
Class for representing a IODEntry	443
gdcm::IODs	
Class for representing a IODs	445
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	446
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 Standard Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit	450
gdcm::IterationEvent	453
gdcm::JPEG12Codec	
Class to do JPEG 12bits (lossy & lossless)	455
gdcm::JPEG16Codec	
Class to do JPEG 16bits (lossless)	457
gdcm::JPEG2000Codec	
Class to do JPEG 2000	459
gdcm::JPEG8Codec	
Class to do JPEG 8bits (lossy & lossless)	462

gdcm::JPEGCodec	
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: gdcm::JPEG8Codec , gdcm::JPEG12Codec & gdcm::JPEG16Codec	
It also support inconsistency in between DICOM header and JPEG compressed stream ImageCodec implementation for the JPEG case	464
gdcm::JPEGLSCodec	
JPEG-LS	468
gdcm::KAKADUCodec	
KAKADUCodec	471
gdcm::LO	
LO	473
gdcm::LookupTable	
LookupTable class	476
gdcm::Scanner::ltstr	480
gdcm::Macro	
Class for representing a Macro	480
gdcm::Macros	
Class for representing a Modules	482
gdcm::network::MaximumLengthSub	
MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIAT-E-RQ)	483
gdcm::MD5	
Class for MD5	484
gdcm::MediaStorage	
MediaStorage	485
gdcm::MemberCommand< T >	
Command subclass that calls a pointer to a member function	492
gdcm::MeshPrimitive	
This class defines surface mesh primitives. It is designed from surface mesh primitives macro	496
gdcm::ModifiedEvent	499
gdcm::Module	
Class for representing a Module	501
gdcm::ModuleEntry	
Class for representing a ModuleEntry	503
gdcm::Modules	
Class for representing a Modules	505
gdcm::MovePatientRootQuery	
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root	507
gdcm::MoveStudyRootQuery	
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root .	509
gdcm::NestedModuleEntries	
Class for representing a NestedModuleEntries	511
gdcm::NoEvent	514
gdcm::Object	
Object	514
gdcm::Orientation	
Class to handle Orientation	517
gdcm::Overlay	
Overlay class	519
gdcm::ParseException	
ParseException Standard exception handling object	525
gdcm::Parser	
Parser ala XML_Parser from expat (SAX)	527

gdcm::Patient	
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54	529
gdcm::network::PDataTFPDU	
PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS	529
gdcm::PDBElement	
Class to represent a PDB Element	532
gdcm::PDBHeader	
Class for PDBHeader	534
gdcm::PDFCodec	
PDFCodec class	536
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	537
gdcm::PersonName	
PersonName class	539
gdcm::PGXCodec	
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images	540
gdcm::PhotometricInterpretation	
Class to represent an PhotometricInterpretation	542
gdcm::PixelFormat	
PixelFormat	544
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)	549
gdcm::PixmapReader	
PixmapReader	552
gdcm::PixmapToPixmapFilter	
PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image	556
gdcm::PixmapWriter	
PixmapWriter This class will takes two inputs:	558
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/	561
gdcm::Preamble	
DICOM Preamble (Part 10)	564
gdcm::PresentationContext	
PresentationContext	565
gdcm::network::PresentationContextAC	
PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS	567
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	568
gdcm::network::PresentationContextRQ	
PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS	570
gdcm::network::PresentationDataValue	
PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS	572
gdcm::Printer	
Printer class	574
gdcm::PrivateDict	
Private Dict	577
gdcm::PrivateTag	
Class to represent a Private DICOM Data Element (Attribute) Tag (Group, Element , Owner)	579

gdcm::ProgressEvent	
ProgressEvent	Special type of event triggered during 580
gdcm::PVRGCodec	
PVRGCodec 583
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 585
gdcm::QueryBase	
QueryBase	contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE . 586
gdcm::QueryFactory	
QueryFactory.h 588
gdcm::QueryImage	
QueryImage	contains: class to construct an image-based query for C-FIND and C-MOVE 589
gdcm::QueryPatient	
QueryPatient	contains: class to construct a patient-based query for c-find and c-move 591
gdcm::QuerySeries	
QuerySeries	contains: class to construct a series-based query for c-find and c-move 593
gdcm::QueryStudy	
QueryStudy.h	contains: class to construct a study-based query for C-FIND and C-MOVE 595
gdcm::RAWCodec	
RAWCodec	class 597
gdcm::Reader	
Reader	ala DOM (Document Object Model) 599
gdcm::Region	
Class for manipulation region 604
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 606
gdcm::RLECodec	
Class to do RLE 609
gdcm::network::RoleSelectionSub	
RoleSelectionSub	PS 3.7 Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ) 612
gdcm::SerieHelper::Rule 613
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 minimum amount of information in each file in order to retrieve the user specified set of DICOM Attribute 614
gdcm::Segment	
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface 620
gdcm::SegmentedPaletteColorLookupTable	
SegmentedPaletteColorLookupTable	class 625
gdcm::SegmentReader	
This class defines a segment reader. It reads attributes of group 0x0062 626
gdcm::SegmentWriter	
This class defines a segment writer. It writes attributes of group 0x0062 629

gdcm::SequenceOfFragments	
Class to represent a Sequence Of Fragments	631
gdcm::SequenceOfItems	
Class to represent a Sequence Of Items (value representation : SQ)	636
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	642
gdcm::Series	
Series	644
gdcm::network::ServiceClassApplicationInformation	645
gdcm::ServiceClassUser	
ServiceClassUser	646
gdcm::SHA1	
Class for SHA1	651
gdcm::SimpleMemberCommand< T >	
Command subclass that calls a pointer to a member function	652
gdcm::SimpleSubjectWatcher	
SimpleSubjectWatcher This is a typical Subject Watcher class. It will observe all events	655
gdcm::SmartPointer< ObjectType >	
Class for Smart Pointer	657
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)	659
gdcm::SOPClassUIDToIOD	
Class convert a class SOP Class UID into IOD	660
gdcm::Sorter	
Sorter General class to do sorting using a custom function You simply need to provide a function of type: Sorter::SortFunction	661
gdcm::Spacing	
Class for Spacing	665
gdcm::Spectroscopy	
Spectroscopy class	667
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	667
gdcm::StartEvent	668
gdcm::static_assert_test< x >	670
gdcm::STATIC_ASSERTION_FAILURE< x >	670
gdcm::STATIC_ASSERTION_FAILURE< true >	670
gdcm::StreamImageReader	
StreamImageReader	670
gdcm::StreamImageWriter	
StreamImageReader	673
gdcm::String< TDelimiter, TMaxLength, TPadChar >	
String	678
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	682
gdcm::Study	
Study	684
gdcm::Subject	
Subject	685

gdcm::Surface	
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes	687
gdcm::SurfaceHelper	
SurfaceHelper Helper class for Surface object	694
gdcm::SurfaceReader	
This class defines a SURFACE IE reader. It reads surface mesh module attributes	696
gdcm::SurfaceWriter	
This class defines a SURFACE IE writer. It writes surface mesh module attributes	698
gdcm::SwapCode	
SwapCode representation	700
gdcm::SwapperDoOp	702
gdcm::SwapperNoOp	703
gdcm::System	
Class to do system operation	703
gdcm::Table	
Table	707
gdcm::TableEntry	
TableEntry	708
gdcm::TableReader	
Class for representing a TableReader	709
gdcm::network::TableRow	711
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)	712
gdcm::TagPath	
Class to handle a path of tag	718
gdcm::Testing	
Class for testing	720
gdcm::Trace	
Trace	724
gdcm::TransferSyntax	
Class to manipulate Transfer Syntax	727
gdcm::network::TransferSyntaxSub	
TransferSyntaxSub Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS	731
gdcm::network::Transition	732
gdcm::Type	
Type	733
gdcm::UI	735
gdcm::UIDGenerator	
Class for generating unique UID	736
gdcm::UIDs	
All known uids	737
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	756
gdcm::network::ULActionAA1	759
gdcm::network::ULActionAA2	760
gdcm::network::ULActionAA3	761
gdcm::network::ULActionAA4	762
gdcm::network::ULActionAA5	763
gdcm::network::ULActionAA6	764
gdcm::network::ULActionAA7	766
gdcm::network::ULActionAA8	767

gdcmm::network::ULActionAE1	768
gdcmm::network::ULActionAE2	769
gdcmm::network::ULActionAE3	770
gdcmm::network::ULActionAE4	771
gdcmm::network::ULActionAE5	773
gdcmm::network::ULActionAE6	774
gdcmm::network::ULActionAE7	775
gdcmm::network::ULActionAE8	776
gdcmm::network::ULActionAR1	777
gdcmm::network::ULActionAR10	778
gdcmm::network::ULActionAR2	780
gdcmm::network::ULActionAR3	781
gdcmm::network::ULActionAR4	782
gdcmm::network::ULActionAR5	783
gdcmm::network::ULActionAR6	784
gdcmm::network::ULActionAR7	785
gdcmm::network::ULActionAR8	787
gdcmm::network::ULActionAR9	788
gdcmm::network::ULActionDT1	789
gdcmm::network::ULActionDT2	790
gdcmm::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	791
gdcmm::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	793
gdcmm::network::ULConnectionCallback	795
gdcmm::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	797
gdcmm::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)	798
gdcmm::network::ULError	
ULError base class for network events	801
gdcmm::network::ULTransitionTable	
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates	802
gdcmm::network::ULWritingCallback	802
gdcmm::UNExplicitDataElement	
Class to read/write a DataElement as UNExplicit Data Element	804
gdcmm::UNExplicitImplicitDataElement	
Class to read/write a DataElement as ExplicitImplicit Data Element This class gather two known bugs:	806
gdcmm::Unpacker12Bits	
Pack/Unpack 12 bits pixel into 16bits	808
gdcmm::Usage	
Usage	809
gdcmm::UserEvent	810

gdcm::network::UserInformation	
UserInformation Table 9-16 USER INFORMATION ITEM FIELDS	812
gdcm::Validate	
Validate class	813
gdcm::Value	
Class to represent the value of a Data Element	814
gdcm::ValueIO< TDE, TSwap, TType >	
Class to dispatch template calls	816
gdcm::Version	
Major/minor and build version	817
gdcm::VL	
Value Length	818
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	820
gdcm::VMToLength< T >	824
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	824
gdcm::VR16ExplicitDataElement	
Class to read/write a DataElement as Explicit Data Element	829
gdcm::VRToEncoding< T >	831
gdcm::VRToType< T >	831
gdcm::VRVLSize< T >	832
gdcm::VRVLSize< 0 >	832
gdcm::VRVLSize< 1 >	832
vtkGDCMImageReader	833
vtkGDCMImageWriter	839
vtkGDCMMedicalImageProperties	843
vtkGDCMPolyDataReader	845
vtkGDCMPolyDataWriter	848
vtkGDCMTesting	851
vtkGDCMThreadedImageReader	853
vtkGDCMThreadedImageReader2	856
vtkImageColorViewer	859
vtkImageMapToColors16	866
vtkImageMapToWindowLevelColors2	869
vtkImagePlanarComponentsToComponents	871
vtkImageRGBToYBR	873
vtkImageYBRToRGB	874
vtkLookupTable16	876
vtkRTStructSetProperties	878
gdcm::Waveform	
Waveform class	883
gdcm::Writer	
Writer ala DOM (Document Object Model) This class is a non-validating writer, it will only performs well- formedness check only	883
gdcm::XMLDictReader	
Class for representing a XMLDictReader	889
gdcm::XMLPrivateDictReader	
Class for representing a XMLPrivateDictReader	890

Chapter 23

File Index

23.1 File List

Here is a list of all files with brief descriptions:

gdc2pnm.man	893
gdc2vtk.man	893
gdcmAAabortPDU.h	893
gdcmAAAssociateACPDU.h	894
gdcmAAAssociateRJPDU.h	894
gdcmAAAssociateRQPDU.h	895
gdcmAbstractSyntax.h	896
gdcmanon.man	897
gdcmAnonymizeEvent.h	897
gdcmAnonymizer.h	899
gdcmApplicationContext.h	899
gdcmApplicationEntity.h	900
gdcmAReleaseRPPDU.h	901
gdcmAReleaseRQPDU.h	902
gdcmARTIMTimer.h	903
gdcmASN1.h	904
gdcmAsynchronousOperationsWindowSub.h	905
gdcmAttribute.h	905
gdcmAudioCodec.h	907
gdcmBase64.h	907
gdcmBaseCompositeMessage.h	908
gdcmBasePDU.h	909
gdcmBaseRootQuery.h	910
gdcmBasicOffsetTable.h	912
gdcmBitmap.h	913
gdcmBitmapToBitmapFilter.h	914
gdcmBoxRegion.h	915
gdcmByteBuffer.h	915
gdcmByteSwap.h	916
gdcmByteSwapFilter.h	917
gdcmByteValue.h	918
gdcmCEchoMessages.h	919
gdcmCFindMessages.h	919
gdcmCMoveMessages.h	920

gdcmCodec.h	921
gdcmCoder.h	922
gdcmCodeString.h	923
gdcmCommand.h	924
gdcmCommandDataSet.h	926
gdcmCompositeMessageFactory.h	926
gdcmCompositeNetworkFunctions.h	927
gdcmConstCharWrapper.h	928
gdcmconv.man	928
gdcmCP246ExplicitDataElement.h	929
gdcmCryptographicMessageSyntax.h	929
gdcmCSAElement.h	930
gdcmCSAHeader.h	931
gdcmCSAHeaderDict.h	932
gdcmCSAHeaderDictEntry.h	933
gdcmCStoreMessages.h	934
gdcmCurve.h	935
gdcmDataElement.h	936
gdcmDataEvent.h	938
gdcmDataSet.h	939
gdcmDataSetEvent.h	940
gdcmDataSetHelper.h	940
gdcmDecoder.h	941
gdcmDefinedTerms.h	942
gdcmDeflateStream.h	943
gdcmDefs.h	943
gdcmDeltaEncodingCodec.h	945
gdcmDICOMDIR.h	945
gdcmDICOMDIRGenerator.h	946
gdcmDict.h	947
gdcmDictConverter.h	949
gdcmDictEntry.h	949
gdcmDictPrinter.h	951
gdcmDicts.h	951
gdcmdiff.man	952
gdcmDIMSE.h	953
gdcmDirectionCosines.h	953
gdcmDirectory.h	954
gdcmDirectoryHelper.h	955
gdcmDummyValueGenerator.h	956
gdcmdump.man	956
gdcmDumper.h	957
gdcmElement.h	957
gdcmEncapsulatedDocument.h	959
gdcmEnumeratedValues.h	959
gdcmEvent.h	960
gdcmException.h	962
gdcmExplicitDataElement.h	963
gdcmExplicitImplicitDataElement.h	963
gdcmFiducials.h	964
gdcmFile.h	965
gdcmFileAnonymizer.h	966
gdcmFileDerivation.h	966
gdcmFileExplicitFilter.h	967

gdcmFileMetaInformation.h	968
gdcmFilename.h	969
gdcmFilenameGenerator.h	969
gdcmFileSet.h	970
gdcmFindPatientRootQuery.h	972
gdcmFindStudyRootQuery.h	973
gdcmFragment.h	973
gdcmgendir.man	975
gdcmGlobal.h	975
gdcmGroupDict.h	976
gdcmIconImage.h	976
gdcmIconImageFilter.h	977
gdcmIconImageGenerator.h	978
gdcmImage.h	979
gdcmImageApplyLookupTable.h	980
gdcmImageChangePhotometricInterpretation.h	981
gdcmImageChangePlanarConfiguration.h	982
gdcmImageChangeTransferSyntax.h	983
gdcmImageCodec.h	984
gdcmImageConverter.h	985
gdcmImageFragmentSplitter.h	986
gdcmImageHelper.h	987
gdcmImageReader.h	988
gdcmImageRegionReader.h	989
gdcmImageToImageFilter.h	990
gdcmImageWriter.h	991
gdcmimg.man	992
gdcmImplementationClassUIDSub.h	992
gdcmImplementationUIDSub.h	993
gdcmImplementationVersionNameSub.h	994
gdcmImplicitDataElement.h	996
gdcminfo.man	996
gdcmIOD.h	996
gdcmIODEntry.h	998
gdcmIODs.h	1001
gdcmIPPSorter.h	1002
gdcmItem.h	1003
gdcmJPEG12Codec.h	1005
gdcmJPEG16Codec.h	1005
gdcmJPEG2000Codec.h	1006
gdcmJPEG8Codec.h	1007
gdcmJPEGCodec.h	1008
gdcmJPEGLSCodec.h	1009
gdcmKAKADUCodec.h	1010
gdcmLegacyMacro.h	1011
gdcmLO.h	1012
gdcmLookupTable.h	1013
gdcmMacro.h	1014
gdcmMacroEntry.h	1016
gdcmMacros.h	1018
gdcmMaximumLengthSub.h	1020
gdcmMD5.h	1021
gdcmMediaStorage.h	1022
gdcmMeshPrimitive.h	1023

gdcmModule.h	1025
gdcmModuleEntry.h	1026
gdcmModules.h	1028
gdcmMovePatientRootQuery.h	1029
gdcmMoveStudyRootQuery.h	1030
gdcmNestedModuleEntries.h	1031
gdcmNetworkEvents.h	1033
gdcmNetworkStateID.h	1034
gdcmObject.h	1035
gdcmOrientation.h	1036
gdcmOverlay.h	1036
gdcmParseException.h	1037
gdcmParser.h	1039
gdcmPatient.h	1039
gdcmPDataTFPDU.h	1040
gdcmPDBElement.h	1041
gdcmPDBHeader.h	1043
gdcmpdf.man	1043
gdcmPDFCodec.h	1043
gdcmPDUFactory.h	1044
gdcmPersonName.h	1045
gdcmPGXCodec.h	1046
gdcmPhotometricInterpretation.h	1046
gdcmPixelFormat.h	1047
gdcmPixmap.h	1048
gdcmPixmapReader.h	1049
gdcmPixmapToPixmapFilter.h	1051
gdcmPixmapWriter.h	1051
gdcmPNMCodec.h	1052
gdcmPreamble.h	1053
gdcmPresentationContext.h	1054
gdcmPresentationContextAC.h	1055
gdcmPresentationContextGenerator.h	1057
gdcmPresentationContextRQ.h	1057
gdcmPresentationDataValue.h	1058
gdcmPrinter.h	1059
gdcmPrivateTag.h	1060
gdcmProgressEvent.h	1062
gdcmPVRGCodec.h	1062
gdcmPythonFilter.h	1063
gdcmQueryBase.h	1064
gdcmQueryFactory.h	1065
gdcmQueryImage.h	1066
gdcmQueryPatient.h	1067
gdcmQuerySeries.h	1068
gdcmQueryStudy.h	1069
gdcmraw.man	1070
gdcmRAWCodec.h	1070
gdcmReader.h	1071
gdcmRegion.h	1072
gdcmRescaler.h	1073
gdcmRLECodec.h	1074
gdcmRoleSelectionSub.h	1074
gdcmScanner.h	1075

gdcmscanner.man	1076
gdcmscu.man	1076
gdcmSegment.h	1076
gdcmSegmentedPaletteColorLookupTable.h	1078
gdcmSegmentHelper.h	1078
gdcmSegmentReader.h	1080
gdcmSegmentWriter.h	1081
gdcmSequenceOfFragments.h	1082
gdcmSequenceOfItems.h	1083
gdcmSerieHelper.h	1083
gdcmSeries.h	1085
gdcmServiceClassApplicationInformation.h	1086
gdcmServiceClassUser.h	1087
gdcmSHA1.h	1087
gdcmSimpleSubjectWatcher.h	1088
gdcmSmartPointer.h	1089
gdcmSOPClassExtendedNegociationSub.h	1090
gdcmSOPClassUIDToIOD.h	1091
gdcmSorter.h	1092
gdcmSpacing.h	1094
gdcmSpectroscopy.h	1094
gdcmSplitMosaicFilter.h	1095
gdcmStaticAssert.h	1096
gdcmStreamImageReader.h	1097
gdcmStreamImageWriter.h	1097
gdcmString.h	1098
gdcmStringFilter.h	1099
gdcmStudy.h	1100
gdcmSubject.h	1101
gdcmSurface.h	1102
gdcmSurfaceHelper.h	1103
gdcmSurfaceReader.h	1104
gdcmSurfaceWriter.h	1105
gdcmSwapCode.h	1106
gdcmSwapper.h	1107
gdcmSystem.h	1108
gdcmTable.h	1109
gdcmTableEntry.h	1110
gdcmTableReader.h	1111
gdcmTag.h	1113
gdcmTagPath.h	1114
gdcmTagToVR.h	1114
gdcm.tar.man	1114
gdcmTerminal.h	1115
gdcmTestDriver.h	1116
gdcmTesting.h	1116
gdcmTrace.h	1117
gdcmTransferSyntax.h	1120
gdcmTransferSyntaxSub.h	1121
gdcmType.h	1122
gdcmTypes.h	1124
gdcmUIDGenerator.h	1124
gdcmUIDs.h	1125
gdcmULAction.h	1126

gdcmULActionAA.h	1127
gdcmULActionAE.h	1128
gdcmULActionAR.h	1129
gdcmULActionDT.h	1129
gdcmULBasicCallback.h	1130
gdcmULConnection.h	1131
gdcmULConnectionCallback.h	1132
gdcmULConnectionInfo.h	1133
gdcmULConnectionManager.h	1134
gdcmULEvent.h	1135
gdcmULTransitionTable.h	1136
gdcmULWritingCallback.h	1137
gdcmUNExplicitDataElement.h	1137
gdcmUNExplicitImplicitDataElement.h	1138
gdcmUnpacker12Bits.h	1139
gdcmUsage.h	1139
gdcmUserInformation.h	1142
gdcmValidate.h	1143
gdcmValue.h	1143
gdcmValueIO.h	1144
gdcmVersion.h	1145
gdcmviewer.man	1146
gdcmVL.h	1146
gdcmVM.h	1147
gdcmVR.h	1148
gdcmVR16ExplicitDataElement.h	1151
gdcmWaveform.h	1151
gdcmWin32.h	1152
gdcmWriter.h	1152
gdcmXMLDictReader.h	1153
gdcmXMLPrivateDictReader.h	1154
vtkGDCMImageReader.h	1155
vtkGDCMImageWriter.h	1156
vtkGDCMMedicalImageProperties.h	1156
vtkGDCMPolyDataReader.h	1157
vtkGDCMPolyDataWriter.h	1158
vtkGDCMTesting.h	1158
vtkGDCMThreadedImageReader.h	1159
vtkGDCMThreadedImageReader2.h	1160
vtkImageColorViewer.h	1160
vtkImageMapToColors16.h	1161
vtkImageMapToWindowLevelColors2.h	1161
vtkImagePlanarComponentsToComponents.h	1162
vtkImageRGBToYBR.h	1162
vtkImageYBRToRGB.h	1163
vtkLookupTable16.h	1163
vtkRTStructSetProperties.h	1164

Chapter 24

Namespace Documentation

24.1 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 [Codec](#)
 - [Codec](#) class.*
- class [Coder](#)
 - [Coder](#).*
- class [CodeString](#)
 - [CodeString](#) This is an implementation of DICOM [VR](#): CS The ctor 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 [CryptographicMessageSyntax](#)
 - Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.*
- 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.
- class [EncodingImplementation](#)
EncodingImplementation.
- class [EncodingImplementation< VR::VRASCII >](#)
- class [EncodingImplementation< VR::VRBINARY >](#)
- class [EndEvent](#)
- class [EnumeratedValues](#)
Element. A Data Element with Enumerated Values that does not have a Value equivalent to one of the Values specified in this standard has an invalid value within the scope of a specific Information Object/SOP Class definition. Note:
- class [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 be written, read and/or deleted.
- class [FileAnonymizer](#)
 - [FileAnonymizer](#).
- 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 representation, 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 [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 [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 [gdcm::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 [gdcm::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 Interpretation* 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 representation: *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 Standard Data Element* with a specific *Data Element Tag* of *Value (FFFE,E000)*. The *Item Tag* is followed by a 4 byte *Item* Length field encoded in one of the following two ways *Explicit/ Implicit*.
- 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 redispach in between the different codec implementation: [gdcm::JPEG8Codec](#), [gdcm::JPEG12Codec](#) & [gdcm::JPEG16Codec](#) It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case.*
- class [JPEGLSCCodec](#)
 - JPEG-LS.*
- 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 [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 minimum 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 disappear soon, you've been warned.
- class [Series](#)
[Series](#).
- class [ServiceClassUser](#)
[ServiceClassUser](#).
- class [SHA1](#)
Class for [SHA1](#).
- class [SimpleMemberCommand](#)
[Command](#) subclass that calls a pointer to a member function.
- class [SimpleSubjectWatcher](#)
[SimpleSubjectWatcher](#) 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::Sort-Function](#).
- 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 [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](#)
- class [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 [XMLPrivateDictReader](#)
 - Class for representing a [XMLPrivateDictReader](#).*

Typedefs

- typedef [String](#)<"\", 16 > [AECComp](#)
- typedef [String](#)<"\", 64 > [ASComp](#)
- typedef bool(* [BOOL_FUNCTION_PFILE_PFILE_POINTER](#))(File *, [File](#) *)
- typedef [String](#)<"\", 16 > [CSCComp](#)
- typedef [String](#)<"\", 64 > [DACComp](#)
- 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_LESSEQUAL](#) }
- 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 PDElement &val)`
- `std::ostream & operator<< (std::ostream &os, const CommandDataSet &val)`
- `std::ostream & operator<< (std::ostream &os, const PrivateTag &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 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 Dicts &d)`
- `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 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 CSAElement &val)`
- `std::ostream & operator<< (std::ostream &os, const CSAHeader &d)`
- `std::ostream & operator<< (std::ostream &os, const VL &val)`
- `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 PixelFormat &pf)`
- `std::ostream & operator<< (std::ostream &os, const Fragment &val)`
- `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`

24.1.1 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.

24.1.2 Typedef Documentation

24.1.2.1 `typedef String<'\\',16> gdcm::AECComp`

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

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

24.1.2.4 `typedef String<'\\',16> gdcm::CSCComp`

24.1.2.5 `typedef String<'\\',64> gdcm::DACComp`

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

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

- 24.1.2.8 `typedef Bitmap gdcm::IconImage`
- 24.1.2.9 `typedef String<'\',64> gdcm::LOComp`
- 24.1.2.10 `typedef String<'\',64> gdcm::LTComp`
- 24.1.2.11 `typedef ModuleEntry gdcm::MacroEntry`
- 24.1.2.12 `typedef NestedModuleEntries gdcm::NestedMacroEntries`
- 24.1.2.13 `typedef String<'\',64> gdcm::PNComp`
- 24.1.2.14 `typedef String<'\',64> gdcm::SHComp`
- 24.1.2.15 `typedef String<'\',64> gdcm::STComp`
- 24.1.2.16 `typedef String<'\',16> gdcm::TMComp`
- 24.1.2.17 `typedef String<'\',64,0> gdcm::UIComp`
- 24.1.2.18 `typedef String<'\',64> gdcm::UTComp`

24.1.3 Enumeration Type Documentation

- 24.1.3.1 `enum gdcm::CompOperators`

Enumerator

GDCM_EQUAL
GDCM_DIFFERENT
GDCM_GREATER
GDCM_GREATEROREQUAL
GDCM_LESS
GDCM_LESOREQUAL

- 24.1.3.2 `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

24.1.3.3 enum gdcm::EQueryLevel

Enumerator

ePatient
eStudy
eSeries
eImage

24.1.3.4 enum gdcm::EQueryType

Enumerator

eFind
eMove

24.1.3.5 enum gdcm::ERootType

Enumerator

ePatientRootType
eStudyRootType

24.1.3.6 enum gdcm::LodModeType

Enumerator

LD_ALL
LD_NOSEQ
LD_NOSHADOW
LD_NOSHADOWSEQ

24.1.4 Function Documentation

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

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

24.1.4.2 `VR::VRType gdcmm::GetVRFromTag (Tag const & tag)`

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

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

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

References `gdcmm::Version::Print()`.

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

References `gdcmm::ModuleEntry::DataElementType`, `gdcmm::ModuleEntry::DescriptionField`, and `gdcmm::ModuleEntry::Name`.

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

References `gdcmm::SwapCode::GetSwapCodeString()`.

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

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

References `gdcmm::Region::Print()`.

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

Generic inserter operator for [Event](#) and its subclasses.

References `gdcmm::Event::Print()`.

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

References `gdcmm::PDBElement::NameField`, and `gdcmm::PDBElement::ValueField`.

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

References `gdcmm::DataSet::Print()`.

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

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

References `gdcmm::Orientation::Print()`.

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

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

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

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

References `gdcm::Type::GetTypeString()`.

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

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

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

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

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

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

References `gdcm::File::GetHeader()`.

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

References `gdcm::Usage::GetUsageString()`.

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

References `gdcm::Sorter::Print()`.

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

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

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

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

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

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

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

References `gdcm::PDBHeader::Print()`.

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

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

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

References `gdcmm::PhotometricInterpretation::GetPIString()`.

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

References `gdcmm::Directory::Print()`.

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

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

References `gdcmm::Object::Print()`.

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

References `gdcmm::DataElement::GetByteValue()`, `gdcmm::DataElement::ValueField`, and `gdcmm::DataElement::ValueLengthField`.

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

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

References `gdcmm::CSAElement::DataField`, `gdcmm::ByteValue::GetLength()`, `gdcmm::ByteValue::GetPointer()`, `gdcmm::CSAElement::KeyField`, `gdcmm::CSAElement::NameField`, `gdcmm::CSAElement::NoOfItemsField`, `gdcmm::CSAElement::SyngoDTField`, `gdcmm::CSAElement::ValueMultiplicityField`, `gdcmm::VM::VM1`, and `gdcmm::CSAElement::VRField`.

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

References `gdcmm::CSAHeader::Print()`.

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

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

References `gdcmm::TransferSyntax::GetTSSString()`.

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

References `gdcmm::FileMetaInformation::GetPreamble()`, and `gdcmm::DataSet::Print()`.

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

References `gdcmm::VM::GetVMString()`.

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

References `gdcm::Scanner::Print()`.

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

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

References `gdcm::MediaStorage::GetMSString()`.

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

References `gdcm::VR::GetVRString()`.

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

References `gdcm::PixelFormat::Print()`.

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

References `gdcm::DataElement::TagField`, `gdcm::DataElement::ValueField`, and `gdcm::DataElement::ValueLengthField`.

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

References `gdcm::UI::Internal`.

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

References `gdcm::Object::Print()`, `gdcm::DataElement::TagField`, `gdcm::DataElement::ValueField`, `gdcm::DataElement::ValueLengthField`, and `gdcm::DataElement::VRField`.

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

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

References `gdcm::DataSet::Print()`.

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

References `gdcm::DataSet::Print()`, `gdcm::DataElement::TagField`, and `gdcm::DataElement::ValueLengthField`.

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

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

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

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

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

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

References `gdcmm::ignore_char::m_char`.

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

References `gdcmm::Tag::SetElement()`, and `gdcmm::Tag::SetGroup()`.

24.1.4.63 `template<typename Float > std::string gdcmm::to_string (Float data)`

Referenced by `gdcmm::EncodingImplementation< VR::VRASCII >::Write()`.

24.1.4.64 `gdcmm::TYPETOENCODING (SQ , VRBINARY , unsigned char)`

24.1.5 Variable Documentation

24.1.5.1 `Global gdcmm::GlobalInstance` `[static]`

24.1.5.2 `gdcmm::VRBINARY`

24.2 gdcmm::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 Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)
- 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 FIELDS \(A-ASSOCIATE-RQ\)](#)
- 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. Data-Sets 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

24.2.1 Enumeration Type Documentation

24.2.1.1 enum [gdcmm::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
eUnrecognizedPDURceived
eEventDoesNotExist

24.2.1.2 enum gdcmm::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

24.2.2 Function Documentation

24.2.2.1 int gdcmm::network::GetStateIndex (EStateID inState) [inline]

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

24.2.3 Variable Documentation

24.2.3.1 const int gdcmm::network::cMaxEventID = eEventDoesNotExist

24.2.3.2 const int gdcmm::network::cMaxStateID = 13

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

24.3 gdcmm::SegmentHelper Namespace Reference

Classes

- struct [BasicCodedEntry](#)

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

24.4 gdcmm::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)

24.4.1 Detailed Description

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

- support VT100 compatible shell
- win32 console

24.4.2 Enumeration Type Documentation

24.4.2.1 enum gdcmm::terminal::Attribute

Enumerator

reset
bright
dim
underline
blink
reverse
hidden

24.4.2.2 enum gdcmm::terminal::Color

Enumerator

black
red
green
yellow
blue
magenta
cyan
white

24.4.2.3 enum gdcmm::terminal::Mode

Enumerator

CONSOLE
VT100

24.4.3 Function Documentation

24.4.3.1 GDCM_EXPORT std::string gdcmm::terminal::setattribute (Attribute *att*)

24.4.3.2 GDCM_EXPORT std::string gdcmm::terminal::setbgcolor (Color *c*)

24.4.3.3 GDCM_EXPORT std::string gdcmm::terminal::setfgcolor (Color *c*)

24.4.3.4 GDCM_EXPORT void gdcmm::terminal::setmode (Mode *m*)

Chapter 25

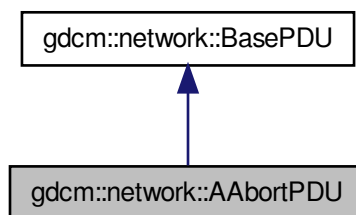
Class Documentation

25.1 gdcm::network::AAabortPDU Class Reference

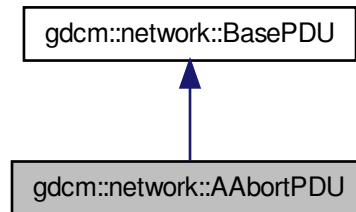
[AAabortPDU](#) [Table 9-26](#) A-ABORT PDU FIELDS.

```
#include <gdcmAAabortPDU.h>
```

Inheritance diagram for `gdcm::network::AAabortPDU`:



Collaboration diagram for `gdcn::network::AAabortPDU`:



Public Member Functions

- [AAabortPDU](#) ()
- 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

25.1.1 Detailed Description

[AAabortPDU Table](#) 9-26 A-ABORT PDU FIELDS.

25.1.2 Constructor & Destructor Documentation

25.1.2.1 `gdcn::network::AAabortPDU::AAabortPDU ()`

25.1.3 Member Function Documentation

25.1.3.1 `bool gdcn::network::AAabortPDU::IsLastFragment () const` `[inline],[virtual]`

Implements [gdcn::network::BasePDU](#).

25.1.3.2 `void gdcn::network::AAabortPDU::Print (std::ostream & os) const` `[virtual]`

Implements [gdcn::network::BasePDU](#).

25.1.3.3 `std::istream& gdcn::network::AAabortPDU::Read (std::istream & is)` `[virtual]`

Implements [gdcn::network::BasePDU](#).

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

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

25.1.3.6 size_t gdcm::network::AAbortPDU::Size () const [virtual]

Implements [gdcm::network::BasePDU](#).

25.1.3.7 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:

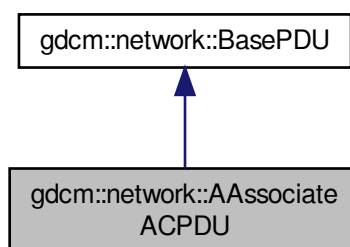
- [gdcmAAbortPDU.h](#)

25.2 gdcm::network::AAssociateACPDU Class Reference

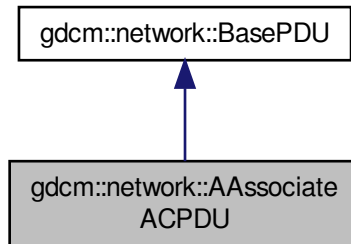
[AAssociateACPDU](#) Table 9-17 ASSOCIATE-AC PDU fields.

```
#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 UserInfo & GetUserInfo () 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](#)

25.2.1 Detailed Description

[AAssociateACPDU Table 9-17](#) ASSOCIATE-AC PDU fields.

25.2.2 Member Typedef Documentation

25.2.2.1 `typedef std::vector<PresentationContextAC>::size_type gdcmm::network::AAAssociateACPDU::SizeType`

25.2.3 Constructor & Destructor Documentation

25.2.3.1 `gdcmm::network::AAAssociateACPDU::AAAssociateACPDU ()`

25.2.4 Member Function Documentation

25.2.4.1 `void gdcmm::network::AAAssociateACPDU::AddPresentationContextAC (PresentationContextAC const & pcac)`

25.2.4.2 `SizeType gdcmm::network::AAAssociateACPDU::GetNumberOfPresentationContextAC () const [inline]`

25.2.4.3 `const PresentationContextAC& gdcmm::network::AAAssociateACPDU::GetPresentationContextAC (SizeType i) [inline]`

25.2.4.4 `const UserInformation& gdcmm::network::AAAssociateACPDU::GetUserInformation () const [inline]`

25.2.4.5 `void gdcmm::network::AAAssociateACPDU::InitFromRQ (AAAssociateRQPDU const & rqpdu)`

25.2.4.6 `bool gdcmm::network::AAAssociateACPDU::IsLastFragment () const [inline],[virtual]`

Implements [gdcmm::network::BasePDU](#).

25.2.4.7 `void gdcmm::network::AAAssociateACPDU::Print (std::ostream & os) const [virtual]`

Implements [gdcmm::network::BasePDU](#).

25.2.4.8 `std::istream& gdcmm::network::AAAssociateACPDU::Read (std::istream & is) [virtual]`

Implements [gdcmm::network::BasePDU](#).

25.2.4.9 `void gdcmm::network::AAAssociateACPDU::SetCalledAETitle (const char calledaetitle[16]) [protected]`

25.2.4.10 `void gdcmm::network::AAAssociateACPDU::SetCallingAETitle (const char callingaetitle[16]) [protected]`

25.2.4.11 `SizeType gdcmm::network::AAAssociateACPDU::Size () const [virtual]`

Implements [gdcmm::network::BasePDU](#).

25.2.4.12 `const std::ostream& gdcmm::network::AAAssociateACPDU::Write (std::ostream & os) const [virtual]`

Implements [gdcmm::network::BasePDU](#).

25.2.5 Friends And Related Function Documentation

25.2.5.1 friend class **AAssociateRQPDU** [friend]

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

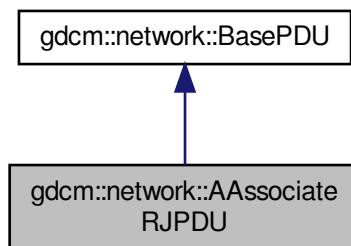
- [gdcmAAssociateACPDU.h](#)

25.3 gdcm::network::AAssociateRJPDU Class Reference

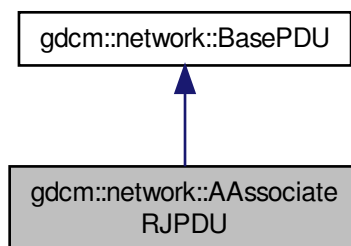
[AAssociateRJPDU](#) Table 9-21 ASSOCIATE-RJ PDU FIELDS.

```
#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

25.3.1 Detailed Description

[AAssociateRJPDU Table](#) 9-21 ASSOCIATE-RJ PDU FIELDS.

25.3.2 Constructor & Destructor Documentation

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

25.3.3 Member Function Documentation

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

Implements [gdcm::network::BasePDU](#).

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

Implements [gdcm::network::BasePDU](#).

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

Implements [gdcm::network::BasePDU](#).

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

Implements [gdcm::network::BasePDU](#).

25.3.3.5 `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:

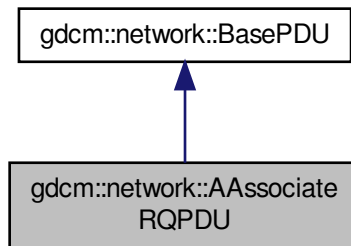
- [gdcmAAssociateRJPDU.h](#)

25.4 gdcm::network::AAssociateRQPDU Class Reference

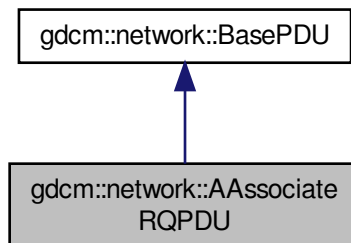
[AAssociateRQPDU Table](#) 9-11 ASSOCIATE-RQ PDU fields.

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

Inheritance diagram for `gdc::network::AAssociateRQPDU`:



Collaboration diagram for `gdc::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 [UserInfo](#) & [GetUserInfo](#) () 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 [SetUserInfo](#) ([UserInfo](#) 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 title is a valid AE title.

Protected Member Functions

- std::string [GetReserved43_74](#) () const

Friends

- class [AAssociateACPDU](#)

25.4.1 Detailed Description

[AAssociateRQPDU](#) Table 9-11 ASSOCIATE-RQ PDU fields.

25.4.2 Member Typedef Documentation

25.4.2.1 typedef std::vector<[PresentationContextRQ](#)> gdcm::network::AAssociateRQPDU::PresentationContext-
ArrayType

25.4.2.2 typedef std::vector<[PresentationContextRQ](#)>::size_type gdcm::network::AAssociateRQPDU::SizeType

25.4.3 Constructor & Destructor Documentation

25.4.3.1 gdcm::network::AAssociateRQPDU::AAssociateRQPDU ()

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

25.4.4 Member Function Documentation

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

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

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

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

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

25.4.4.6 `const PresentationContextRQ* gdcn::network::AAssociateRQPDU::GetPresentationContextByAbstractSyntax (AbstractSyntax const & as) const`

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

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

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

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

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

Check whether or not the title is a valid AE title.

25.4.4.12 `bool gdcn::network::AAssociateRQPDU::IsLastFragment () const [inline],[virtual]`

Implements [gdcn::network::BasePDU](#).

25.4.4.13 `void gdcn::network::AAssociateRQPDU::Print (std::ostream & os) const [virtual]`

This function will initialize an [AAssociateACPDU](#) from the fields in the [AAssociateRQPDU](#) structure

Implements [gdcn::network::BasePDU](#).

25.4.4.14 `std::istream& gdcn::network::AAssociateRQPDU::Read (std::istream & is) [virtual]`

Implements [gdcn::network::BasePDU](#).

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

Set the Called AE Title.

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

Set the Calling AE Title.

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

25.4.4.18 size_t gdcm::network::AAssociateRQPDU::Size () const [virtual]

Implements [gdcm::network::BasePDU](#).

25.4.4.19 const std::ostream& gdcm::network::AAssociateRQPDU::Write (std::ostream & *os*) const [virtual]

Implements [gdcm::network::BasePDU](#).

25.4.5 Friends And Related Function Documentation

25.4.5.1 friend class AAssociateACPDU [friend]

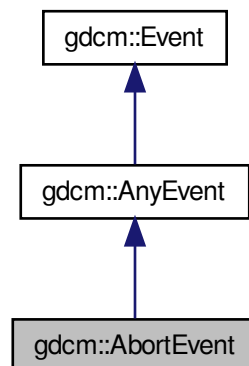
The documentation for this class was generated from the following file:

- [gdcmAAssociateRQPDU.h](#)

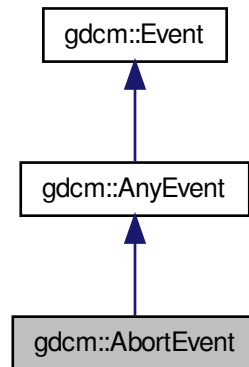
25.5 gdcm::AbortEvent Class Reference

```
#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](#)

25.6 `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

25.6.1 Detailed Description

[AbstractSyntax](#) Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.

25.6.2 Constructor & Destructor Documentation

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

25.6.3 Member Function Documentation

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

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

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

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

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

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

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

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

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

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

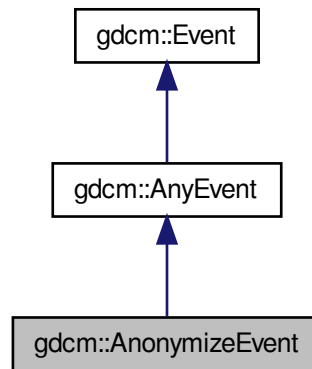
- [gdcmAbstractSyntax.h](#)

25.7 gdcm::AnonymizeEvent Class Reference

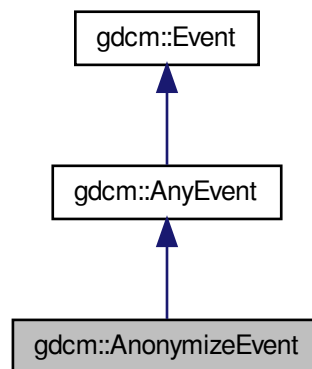
[AnonymizeEvent](#) Special type of event triggered during the Anonymization process.

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

Inheritance diagram for `gdc::AnonymizeEvent`:



Collaboration diagram for `gdc::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)

25.7.1 Detailed Description

[AnonymizeEvent](#) Special type of event triggered during the Anonymization process.

See Also

[Anonymizer](#)

25.7.2 Member Typedef Documentation

25.7.2.1 typedef [AnonymizeEvent](#) [gdcm::AnonymizeEvent::Self](#)

25.7.2.2 typedef [AnyEvent](#) [gdcm::AnonymizeEvent::Superclass](#)

25.7.3 Constructor & Destructor Documentation

25.7.3.1 [gdcm::AnonymizeEvent::AnonymizeEvent](#) ([Tag](#) const & *tag* = 0) [\[inline\]](#)

25.7.3.2 virtual [gdcm::AnonymizeEvent::~~AnonymizeEvent](#) () [\[inline\]](#),[\[virtual\]](#)

25.7.3.3 [gdcm::AnonymizeEvent::AnonymizeEvent](#) ([const Self](#) & *s*) [\[inline\]](#)

25.7.4 Member Function Documentation

25.7.4.1 virtual bool [gdcm::AnonymizeEvent::CheckEvent](#) ([const ::gdcm::Event](#) * *e*) const [\[inline\]](#),[\[virtual\]](#)

25.7.4.2 virtual const char* [gdcm::AnonymizeEvent::GetEventName](#) () const [\[inline\]](#),[\[virtual\]](#)

Return the StringName associated with the event.

Implements [gdcm::Event](#).

25.7.4.3 [Tag](#) const& [gdcm::AnonymizeEvent::GetTag](#) () const [\[inline\]](#)

25.7.4.4 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](#).

25.7.4.5 void [gdcm::AnonymizeEvent::SetTag](#) ([const Tag](#) & *t*) [\[inline\]](#)

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

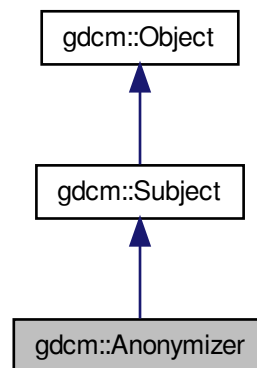
- [gdcmAnonymizeEvent.h](#)

25.8 gdcm::Anonymizer Class Reference

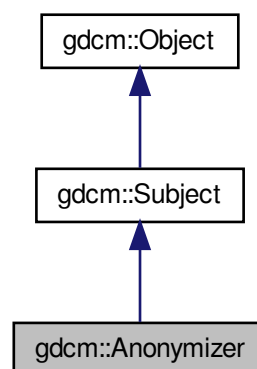
[Anonymizer](#) This class is a multi purpose anonymizer. It can work in 2 mode:

```
#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 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 [BALCPPProtect](#) ([DataSet](#) &ds, [Tag](#) const &tag, const [IOD](#) &iod)
- bool [CanEmptyTag](#) ([Tag](#) const &tag, const [IOD](#) &iod) const
- void [RecurseDataSet](#) ([DataSet](#) &ds)

25.8.1 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 $O(m \cdot \log(n))$ 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 [gdcm::Anonymizer](#) class when anonymizing a [FileSet](#). Once the [gdcm::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](#).

25.8.2 Constructor & Destructor Documentation

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

25.8.2.2 `gdcm::Anonymizer::~~Anonymizer ()`

25.8.3 Member Function Documentation

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

25.8.3.2 `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

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

25.8.3.4 `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](#).

25.8.3.5 `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](#).

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

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

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

for wrapped language: instantiate a reference counted object

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

25.8.3.10 `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

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

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

Examples:

[ClinicalTrialAnnotate.cxx](#).

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

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

Examples:

[ClinicalTrialAnnotate.cxx](#).

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

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

25.8.3.14 `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 VRASCI

Examples:

[ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), and [EncapsulateFileInRawData.cxx](#).

25.8.3.15 `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

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

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

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

Set/Get [File](#).

Examples:

[ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), and [EncapsulateFileInRawData.cxx](#).

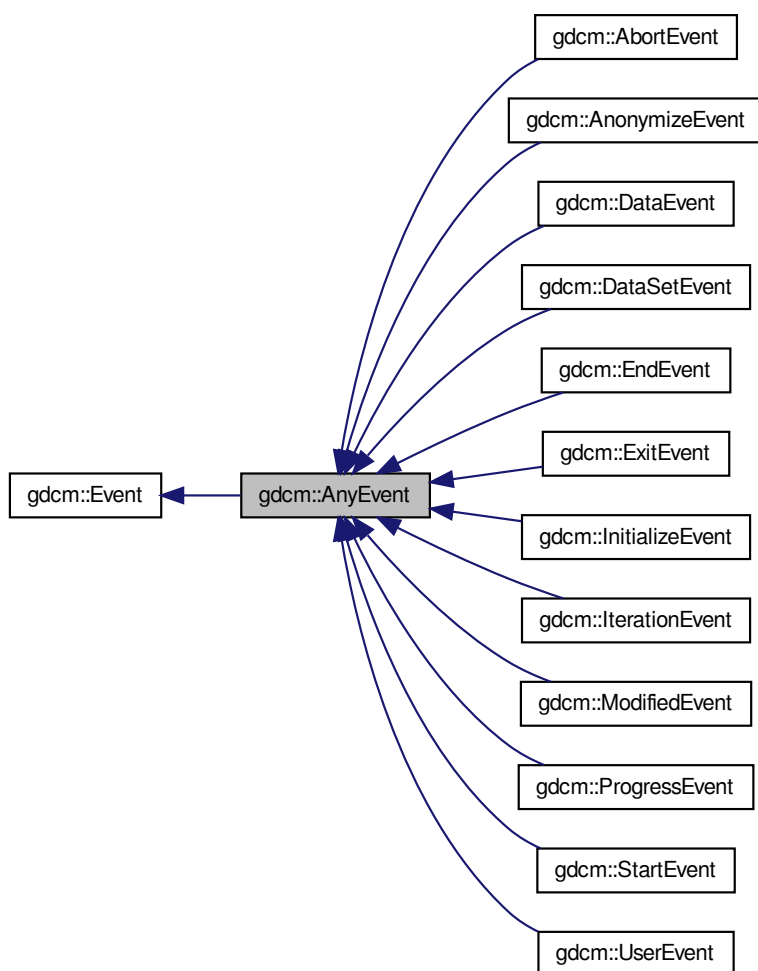
The documentation for this class was generated from the following file:

- [gdcmAnonymizer.h](#)

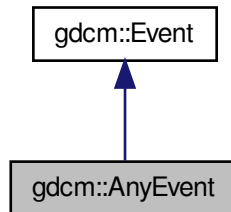
25.9 `gdcm::AnyEvent` Class Reference

```
#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](#)

25.10 gdcm::network::ApplicationContext Class Reference

[ApplicationContext](#) Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

```
#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`

25.10.1 Detailed Description

[ApplicationContext](#) Table 9-12 APPLICATION CONTEXT ITEM FIELDS Looks like Application Context can only be 64 bytes at max (see Figure 9-1 / PS 3.8 - 2009)

25.10.2 Constructor & Destructor Documentation

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

25.10.3 Member Function Documentation

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

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

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

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

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

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

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

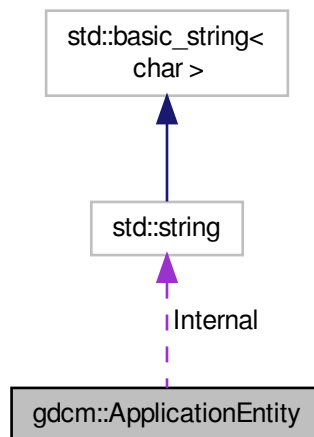
- [gdcmApplicationContext.h](#)

25.11 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](#) = ''

25.11.1 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

25.11.2 Member Function Documentation

25.11.2.1 bool gdcM::ApplicationEntity::IsValid () const [inline]

25.11.2.2 void gdcM::ApplicationEntity::Print (std::ostream & os) const [inline]

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

25.11.2.4 void gdcM::ApplicationEntity::Squeeze () [inline]

25.11.3 Member Data Documentation

25.11.3.1 std::string gdcM::ApplicationEntity::Internal

25.11.3.2 const unsigned int gdcM::ApplicationEntity::MaxLength = 16 [static]

25.11.3.3 const unsigned int gdcM::ApplicationEntity::MaxNumberOfComponents = 1 [static]

25.11.3.4 const char gdcM::ApplicationEntity::Padding = '' [static]

25.11.3.5 const char gdcM::ApplicationEntity::Separator = '' [static]

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

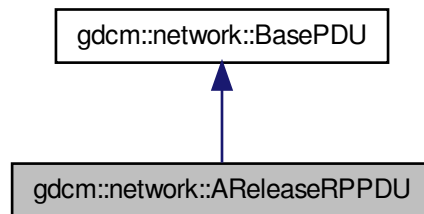
- [gdcmmApplicationEntity.h](#)

25.12 gdcmm::network::AReleaseRPPDU Class Reference

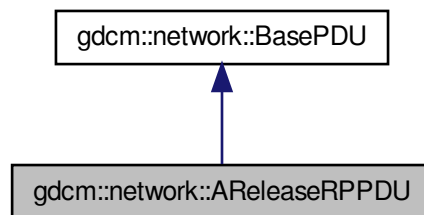
[AReleaseRPPDU](#) Table 9-25 A-RELEASE-RP PDU fields.

```
#include <gdcmmAReleaseRPPDU.h>
```

Inheritance diagram for gdcmm::network::AReleaseRPPDU:



Collaboration diagram for gdcmm::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

25.12.1 Detailed Description

[AReleaseRPPDU Table](#) 9-25 A-RELEASE-RP PDU fields.

25.12.2 Constructor & Destructor Documentation

25.12.2.1 `gdcn::network::AReleaseRPPDU::AReleaseRPPDU ()`

25.12.3 Member Function Documentation

25.12.3.1 `bool gdcn::network::AReleaseRPPDU::IsLastFragment () const` `[inline],[virtual]`

Implements [gdcn::network::BasePDU](#).

25.12.3.2 `void gdcn::network::AReleaseRPPDU::Print (std::ostream & os) const` `[virtual]`

Implements [gdcn::network::BasePDU](#).

25.12.3.3 `std::istream& gdcn::network::AReleaseRPPDU::Read (std::istream & is)` `[virtual]`

Implements [gdcn::network::BasePDU](#).

25.12.3.4 `size_t gdcn::network::AReleaseRPPDU::Size () const` `[virtual]`

Implements [gdcn::network::BasePDU](#).

25.12.3.5 `const std::ostream& gdcn::network::AReleaseRPPDU::Write (std::ostream & os) const` `[virtual]`

Implements [gdcn::network::BasePDU](#).

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

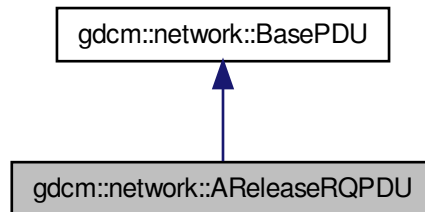
- [gdcnAReleaseRPPDU.h](#)

25.13 gdcn::network::AReleaseRQPDU Class Reference

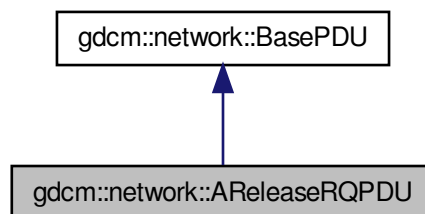
[AReleaseRQPDU Table](#) 9-24 A-RELEASE-RQ PDU FIELDS.

```
#include <gdcnAReleaseRQPDU.h>
```


Inheritance diagram for gdcmm::network::AReleaseRQPDU:



Collaboration diagram for gdcmm::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

25.13.1 Detailed Description

[AReleaseRQPDU](#) Table 9-24 A-RELEASE-RQ PDU FIELDS.

25.13.2 Constructor & Destructor Documentation

25.13.2.1 `gdcmm::network::AReleaseRQPDU::AReleaseRQPDU ()`

25.13.3 Member Function Documentation

25.13.3.1 `bool gdcmm::network::AReleaseRQPDU::IsLastFragment () const` `[inline],[virtual]`

Implements [gdcmm::network::BasePDU](#).

25.13.3.2 `void gdcmm::network::AReleaseRQPDU::Print (std::ostream & os) const` `[virtual]`

Implements [gdcmm::network::BasePDU](#).

25.13.3.3 `std::istream& gdcmm::network::AReleaseRQPDU::Read (std::istream & is)` `[virtual]`

Implements [gdcmm::network::BasePDU](#).

25.13.3.4 `size_t gdcmm::network::AReleaseRQPDU::Size () const` `[virtual]`

Implements [gdcmm::network::BasePDU](#).

25.13.3.5 `const std::ostream& gdcmm::network::AReleaseRQPDU::Write (std::ostream & os) const` `[virtual]`

Implements [gdcmm::network::BasePDU](#).

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

- [gdcmmAReleaseRQPDU.h](#)

25.14 gdcmm::network::ARTIMTimer Class Reference

[ARTIMTimer](#) This file contains the code for the ARTIM timer.

```
#include <gdcmmARTIMTimer.h>
```

Public Member Functions

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

25.14.1 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.

25.14.2 Constructor & Destructor Documentation

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

25.14.3 Member Function Documentation

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

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

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

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

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

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

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

- [gdcmARTIMTimer.h](#)

25.15 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](#) ()

25.15.1 Detailed Description

Class for [ASN1](#).

25.15.2 Constructor & Destructor Documentation

25.15.2.1 `gdcmm::ASN1::ASN1 ()`

25.15.2.2 `gdcmm::ASN1::~~ASN1 ()`

25.15.3 Member Function Documentation

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

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

25.15.3.3 `int gdcmm::ASN1::TestPBKDF2 ()` [protected]

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

- [gdcmmASN1.h](#)

25.16 gdcmm::network::AsynchronousOperationsWindowSub Class Reference

[AsynchronousOperationsWindowSub](#) PS 3.7 [Table D.3-7](#) ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELD (A-ASSOCIATE-RQ)

```
#include <gdcmmAsynchronousOperationsWindowSub.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

25.16.1 Detailed Description

[AsynchronousOperationsWindowSub](#) PS 3.7 [Table D.3-7](#) ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELD (A-ASSOCIATE-RQ)

25.16.2 Constructor & Destructor Documentation

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

25.16.3 Member Function Documentation

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

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

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

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

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

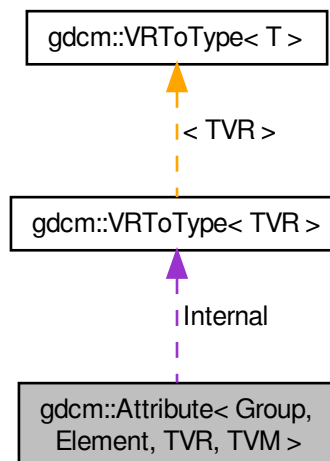
- [gdcmAsynchronousOperationsWindowSub.h](#)

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

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

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

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)

25.17.1 Detailed Description

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

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

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

Examples that will NOT compile are:

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

Examples:

[CreateJIPIDataSet.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSequenceUltrasound.cxx](#), [HelloWorld.cxx](#), [LargeVRDSEExplicit.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndPrintAttributes.cxx](#), [rle2img.cxx](#), [SortImage.cxx](#), [StreamImage-ReaderTest.cxx](#), and [VolumeSorter.cxx](#).

25.17.2 Member Typedef Documentation

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

25.17.3 Member Enumeration Documentation

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

Enumerator

VMType

25.17.4 Member Function Documentation

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

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

25.17.4.3 `template<uint16_t Group, uint16_t Element, int TVR = TagToType<Group, Element>::VRType, int TVM = TagToType<Group, Element>::VMType> gdcmm::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)))`

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

References gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcmm::DataElement::GetVR(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcmm::Attribute< Group, Element, TVR, TVM >::Internal, gdcmm::DataElement::SetByteValue(), gdcmm::DataElement::SetVR(), gdcmm::VR::SQ, gdcmm::VR::UI, and gdcmm::VR::VRASCII.

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

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

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

Referenced by gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::operator!(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!(), gdcmm::Attribute< Group, Element, TVR, TVM >::operator<(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator<(), gdcmm::Attribute< Group, Element, TVR, TVM >::operator==(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==(), gdcmm::Attribute< Group, Element, TVR, TVM >::Print(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues().

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

Referenced by gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, TVM >::Print(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print(), gdcmm::Attribute< Group, Element, TVR, TVM >::Set(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Set(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet().

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::operator[]()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[]()`.

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator<()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator<()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator==()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==()`.

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

Referenced by `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print()`.

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

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.17.4.15 `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()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetValues()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

25.17.4.16 `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()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetValues()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

25.17.4.17 `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()`.

25.17.4.18 `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()`.

25.17.4.19 `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()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetTag()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

25.17.4.20 `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()`.

25.17.4.21 `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()`, `gdcM::ByteValue::GetPointer()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

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()`.

```
25.17.4.22 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(), gdcm::ByteValue::GetPointer(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), and gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement().

```
25.17.4.23 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().

```
25.17.4.24 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().

```
25.17.4.25 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(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

```
25.17.4.26 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(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue(), and gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues().

25.17.5 Member Data Documentation

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

Referenced by `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator<()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator<()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator==()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::Print()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::~~Attribute()`.

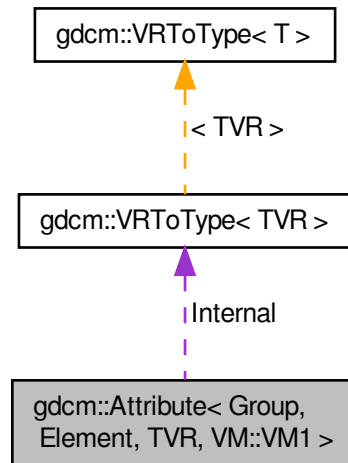
The documentation for this class was generated from the following file:

- [gdcmmAttribute.h](#)

25.18 `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >` Class Template Reference

```
#include <gdcmmAttribute.h>
```

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



Public Types

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

Public Member Functions

- `GDCM_STATIC_ASSERT (VMToLength< VM::VM1 >::Length==1)`
- `GDCM_STATIC_ASSERT (((VR::VRTType) TVR &(VR::VRTType)(TagToType< Group, Element >::VRTType)))`
- `GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))`
- `GDCM_STATIC_ASSERT (((((VR::VRTType) TVR &VR::VR_VM1)&&((VM::VMType) VM::VM1==VM::VM1))||!((VR::VRTType) 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)

25.18.1 Member Typedef Documentation

25.18.1.1 `template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::ArrayType`

25.18.2 Member Enumeration Documentation

25.18.2.1 `template<uint16_t Group, uint16_t Element, int TVR> anonymous enum`

Enumerator

VMType

25.18.3 Member Function Documentation

25.18.3.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (VMTToLength< VM::VM1 >::Length ==1)`

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

25.18.3.3 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GDCM_STATIC_ASSERT (((VM::VMType) VM::VM1 &(VM::VMType)(TagToType< Group, Element >::VMType)))`

25.18.3.4 `template<uint16_t Group, uint16_t Element, int TVR> gdcmm::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)))`

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcmm::DataElement::GetVR()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR()`,

gdcm::Attribute< Group, Element, TVR, TVM >::Internal, gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VRASCII.

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

25.18.3.7 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetDictVR () [inline],[static]`

25.18.3.8 `template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetNumberOfValues () const [inline]`

25.18.3.9 `template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetTag () [inline],[static]`

25.18.3.10 `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.

25.18.3.11 `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.

25.18.3.12 `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.

25.18.3.13 `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.

25.18.3.14 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1 >::GetVR () [inline],[static]`

25.18.3.15 `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(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.16 `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(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

25.18.3.17 `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()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetValues()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

25.18.3.18 `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`.

25.18.3.19 `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()`.

25.18.3.20 `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()`, `gdcM::ByteValue::GetPointer()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

25.18.3.21 `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()`, `gdcM::ByteValue::GetPointer()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

25.18.3.22 `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`.

25.18.3.23 `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()`.

25.18.3.24 `template<uint16_t Group, uint16_t Element, int TVR> void gdcM::Attribute< Group, Element, TVR, VM::VM1 >::SetValue (ArrayType v) [inline]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

25.18.4 Member Data Documentation

25.18.4.1 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType gdcm::Attribute< Group, Element, TVR, VM::VM1 >::Internal`

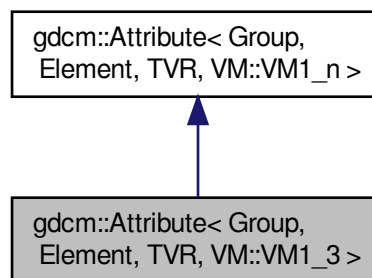
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

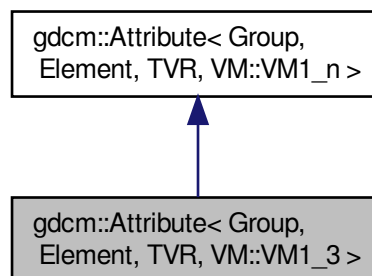
25.19 gdcm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference

```
#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

25.19.1 Member Function Documentation

25.19.1.1 `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`.

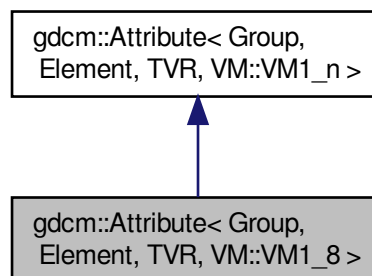
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

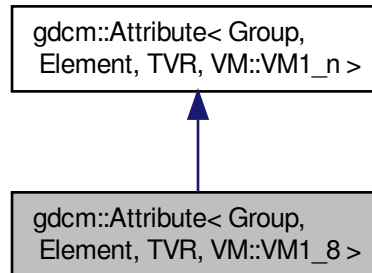
25.20 `gdcM::Attribute< Group, Element, TVR, VM::VM1_8 >` Class Template Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for `gdcM::Attribute< Group, Element, TVR, VM::VM1_8 >`:



Collaboration diagram for gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >:



Public Member Functions

- [VM GetVM](#) () const

Additional Inherited Members

25.20.1 Member Function Documentation

25.20.1.1 `template<uint16_t Group, uint16_t Element, int TVR> VM gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >::GetVM () const [inline]`

References gdcmm::VM::VM1_8.

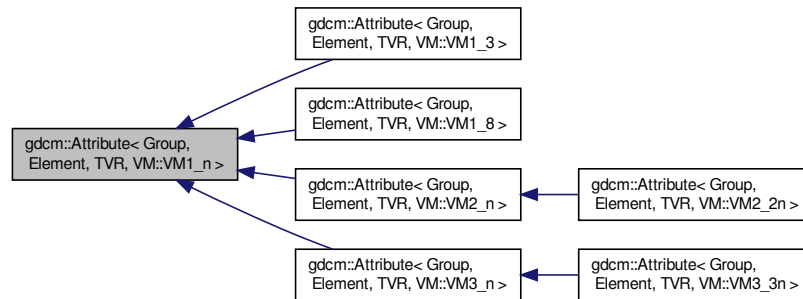
The documentation for this class was generated from the following file:

- [gdcmmAttribute.h](#)

25.21 gdcmm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmmAttribute.h>
```

Inheritance diagram for `gdc::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)

25.21.1 Member Typedef Documentation

25.21.1.1 `template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::ArrayType`

25.21.2 Constructor & Destructor Documentation

25.21.2.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute () [inline],[explicit]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.2.2 `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`.

25.21.3 Member Function Documentation

25.21.3.1 `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)))`

25.21.3.2 `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)))`

25.21.3.3 `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)))`

25.21.3.4 `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::Attribute< Group, Element, TVR, TVM >::Internal`, `gdcm::DataElement::SetByteValue()`, `gdcm::DataElement::SetVR()`, `gdcm::VR::SQ`, `gdcm::VR::UI`, and `gdcm::VR::VRASCII`.

25.21.3.5 `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()`.

25.21.3.6 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR () [inline],[static]`

25.21.3.7 `template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues () const [inline]`

25.21.3.8 `template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag () [inline], [static]`

25.21.3.9 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (unsigned int idx = 0) [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.10 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue (unsigned int idx = 0) const [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.11 `template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues () const [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.12 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM () [inline], [static]`

References `gdcmm::VM::VM1_n`.

25.21.3.13 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR () [inline], [static]`

25.21.3.14 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (unsigned int idx) [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue()`.

25.21.3.15 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType const& gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] (unsigned int idx) const [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue()`.

25.21.3.16 `template<uint16_t Group, uint16_t Element, int TVR> void gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print (std::ostream & os) const [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetVM()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.17 `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()`.

25.21.3.18 `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()`.

25.21.3.19 `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()`.

25.21.3.20 `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()`.

25.21.3.21 `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()`.

25.21.3.22 `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()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

25.21.3.23 `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()`.

25.21.3.24 `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`.

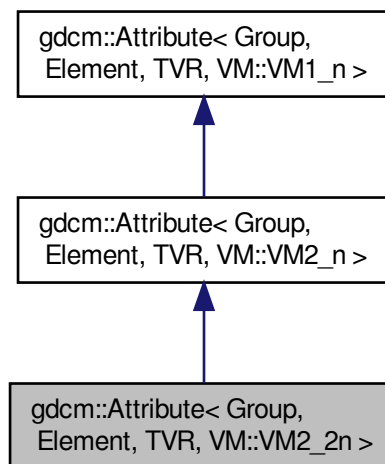
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

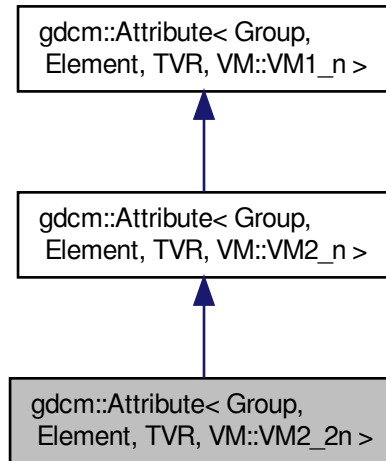
25.22 `gdcM::Attribute< Group, Element, TVR, VM::VM2_2n >` Class Template Reference

```
#include <gdcMAttribute.h>
```

Inheritance diagram for `gdcM::Attribute< Group, Element, TVR, VM::VM2_2n >`:



Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2_n >:



Static Public Member Functions

- static [VM GetVM](#) ()

Additional Inherited Members

25.22.1 Member Function Documentation

25.22.1.1 `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`.

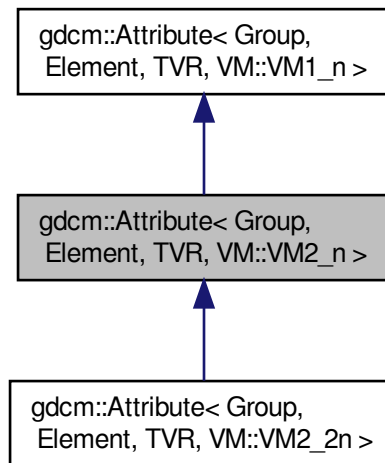
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

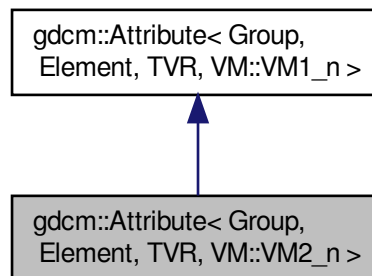
25.23 gdcm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference

```
#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

25.23.1 Member Function Documentation

```
25.23.1.1  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.

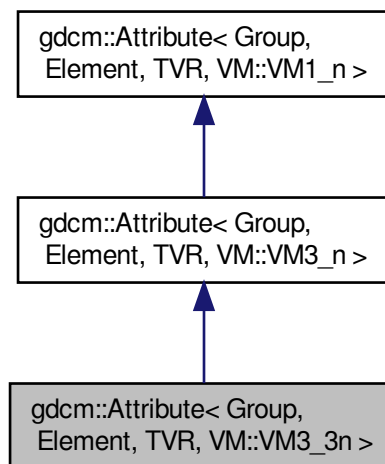
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

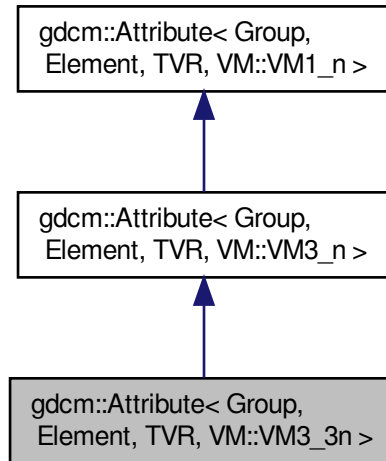
25.24 gdcM::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference

```
#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

25.24.1 Member Function Documentation

25.24.1.1 `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`.

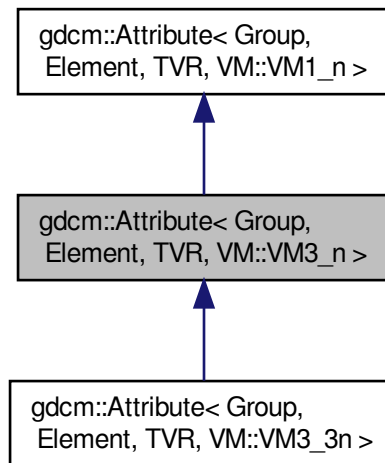
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

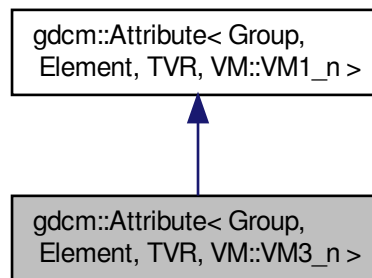
25.25 `gdcM::Attribute< Group, Element, TVR, VM::VM3_n >` Class Template Reference

```
#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

25.25.1 Member Function Documentation

25.25.1.1 `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:

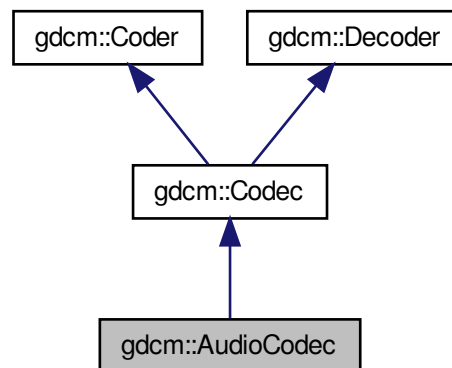
- [gdcMAttribute.h](#)

25.26 gdcM::AudioCodec Class Reference

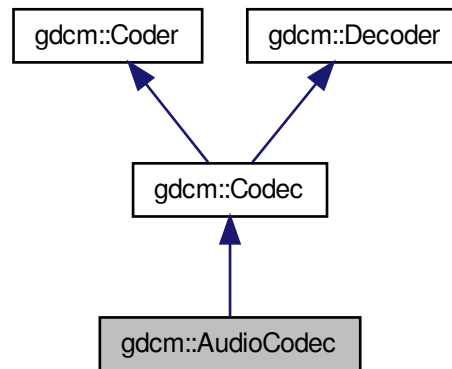
[AudioCodec](#).

```
#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

25.26.1 Detailed Description

[AudioCodec](#).

25.26.2 Constructor & Destructor Documentation

25.26.2.1 `gdcm::AudioCodec::AudioCodec ()`

25.26.2.2 `gdcm::AudioCodec::~~AudioCodec ()`

25.26.3 Member Function Documentation

25.26.3.1 `bool gdcm::AudioCodec::CanCode (TransferSyntax const &) const` `[inline], [virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

25.26.3.2 `bool gdcm::AudioCodec::CanDecode (TransferSyntax const &) const` `[inline],[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

25.26.3.3 `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](#)

25.27 gdcm::Base64 Class Reference

Class for [Base64](#).

```
#include <gdcmBase64.h>
```

Public Member Functions

- [Base64](#) ()
- [~Base64](#) ()

Static Public Member Functions

- static int [Decode](#) (char *dst, int dlen, const char *src, int slen)
Decode a base64-formatted buffer.
- static int [Encode](#) (char *dst, int dlen, const char *src, int slen)
Encode a buffer into base64 format.
- static int [GetDecodeLength](#) (const char *src, int slen)
- static int [GetEncodeLength](#) (const char *src, int slen)

25.27.1 Detailed Description

Class for [Base64](#).

25.27.2 Constructor & Destructor Documentation

25.27.2.1 `gdcm::Base64::Base64 ()`

25.27.2.2 `gdcm::Base64::~~Base64 ()`

25.27.3 Member Function Documentation

25.27.3.1 `static int gdcm::Base64::Decode (char * dst, int dlen, const char * src, int slen)` `[static]`

Decode a base64-formatted buffer.

Parameters

<i>dst</i>	destination buffer
<i>dlen</i>	size of the buffer
<i>src</i>	source buffer
<i>slen</i>	amount of data to be decoded

Returns

0 if successful

25.27.3.2 `static int gdcM::Base64::Encode (char * dst, int dlen, const char * src, int slen)` `[static]`

Encode a buffer into base64 format.

Parameters

<i>dst</i>	destination buffer
<i>dlen</i>	size of the buffer
<i>src</i>	source buffer
<i>slen</i>	amount of data to be encoded

Returns

0 if successful

25.27.3.3 `static int gdcM::Base64::GetDecodeLength (const char * src, int slen)` `[static]`

Call this function with *dlen = 0 to obtain the required buffer size in *dlen

25.27.3.4 `static int gdcM::Base64::GetEncodeLength (const char * src, int slen)` `[static]`

Call this function with dlen = 0 to obtain the required buffer size in dlen

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

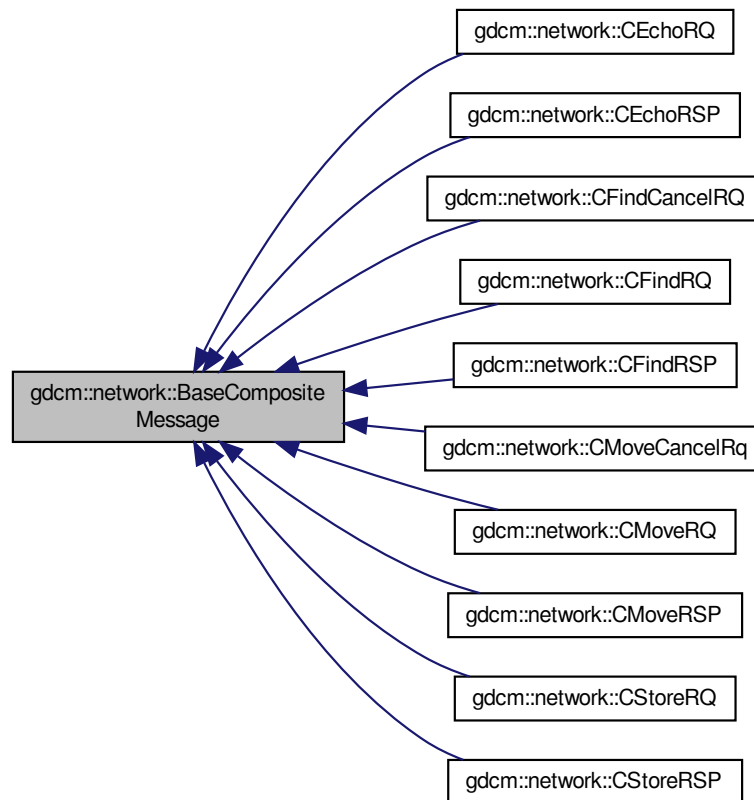
- [gdcMBase64.h](#)

25.28 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 gdcmm::network::BaseCompositeMessage:



Public Member Functions

- virtual std::vector
< [PresentationDataValue](#) > [ConstructPDV](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)=0

25.28.1 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, `gdcmCompositePDUFactory`.

This is an abstract class. It cannot be instantiated on its own.

25.28.2 Member Function Documentation

25.28.2.1 `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:

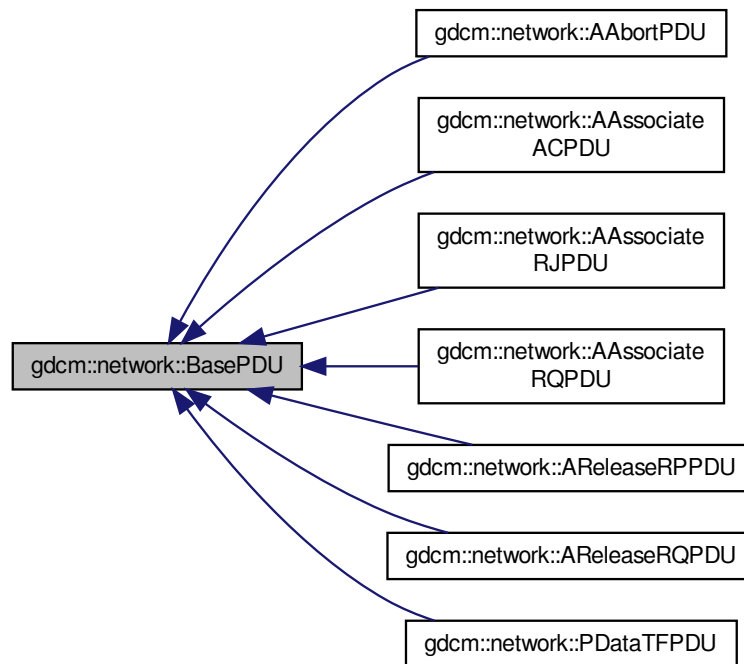
- [gdcmBaseCompositeMessage.h](#)

25.29 gdcm::network::BasePDU Class Reference

[BasePDU](#) base class for PDUs.

```
#include <gdcmBasePDU.h>
```

Inheritance diagram for gdcmm::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

25.29.1 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

25.29.2 Constructor & Destructor Documentation

25.29.2.1 `virtual gdcm::network::BasePDU::~~BasePDU () [inline],[virtual]`

25.29.3 Member Function Documentation

25.29.3.1 `virtual bool gdcm::network::BasePDU::IsLastFragment () const [pure virtual]`

Implemented in [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AAssociateACPDU](#), [gdcm::network::PDataTFPDU](#), [gdcm::network::AAabortPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AReleaseRPPDU](#), and [gdcm::network::AReleaseRQPDU](#).

25.29.3.2 `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::AAabortPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AReleaseRQPDU](#), and [gdcm::network::AAssociateRJPDU](#).

25.29.3.3 `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::AAabortPDU](#).

25.29.3.4 `virtual size_t gdcm::network::BasePDU::Size () const [pure virtual]`

Implemented in [gdcm::network::AAssociateACPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::PDataTFPDU](#), [gdcm::network::AAabortPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AReleaseRPPDU](#), and [gdcm::network::AReleaseRQPDU](#).

25.29.3.5 `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::AAabortPDU](#).

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

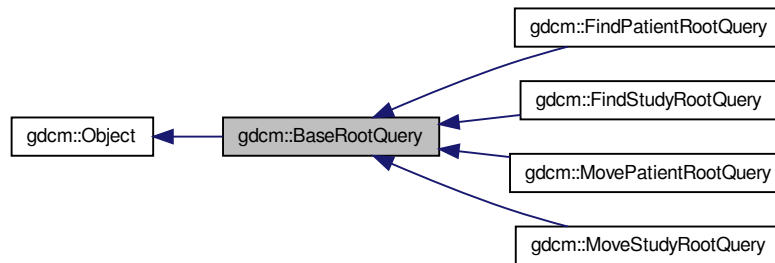
- [gdcmBasePDU.h](#)

25.30 gdcm::BaseRootQuery Class Reference

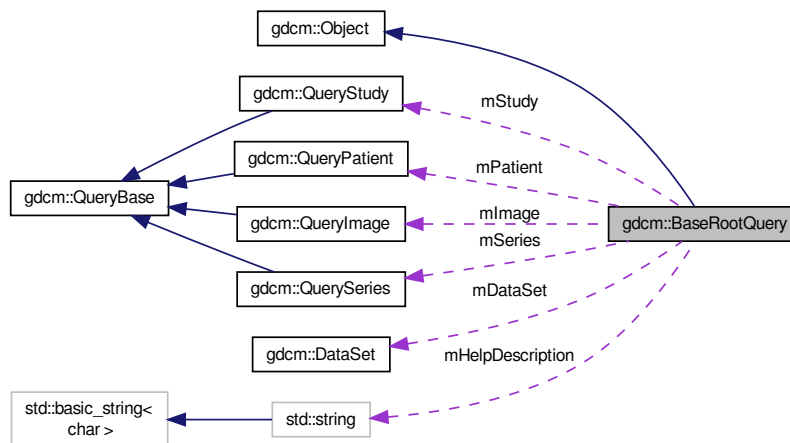
[BaseRootQuery](#) contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.

```
#include <gdcmBaseRootQuery.h>
```

Inheritance diagram for gdcmm::BaseRootQuery:



Collaboration diagram for gdcmm::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](#) qllevel)
- 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](#)

25.30.1 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 derivatives) should be passed to a c-find or c-move query.

25.30.2 Constructor & Destructor Documentation

25.30.2.1 `gdcm::BaseRootQuery::BaseRootQuery ()` [protected]

25.30.2.2 `virtual gdcm::BaseRootQuery::~BaseRootQuery ()` [virtual]

25.30.3 Member Function Documentation

25.30.3.1 void gdcm::BaseRootQuery::AddQueryDataSet (const DataSet & ds)

25.30.3.2 static QueryBase* gdcm::BaseRootQuery::Construct (ERootType inRootType, EQueryLevel qllevel)
[static]

25.30.3.3 virtual UIDs::TSName gdcm::BaseRootQuery::GetAbstractSyntaxUID () const [pure virtual]

Implemented in [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::FindPatientRootQuery](#).

25.30.3.4 DataSet const& gdcm::BaseRootQuery::GetQueryDataSet () const

Set/Get the internal representation of the query as a [DataSet](#).

25.30.3.5 DataSet& gdcm::BaseRootQuery::GetQueryDataSet ()

25.30.3.6 EQueryLevel gdcm::BaseRootQuery::GetQueryLevelFromQueryRoot (ERootType roottype)

25.30.3.7 static int gdcm::BaseRootQuery::GetQueryLevelFromString (const char * str) [static]

25.30.3.8 static const char* gdcm::BaseRootQuery::GetQueryLevelString (EQueryLevel ql) [static]

25.30.3.9 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](#).

25.30.3.10 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](#).

25.30.3.11 void gdcm::BaseRootQuery::Print (std::ostream & os) const [virtual]

Reimplemented from [gdcm::Object](#).

25.30.3.12 void gdcm::BaseRootQuery::SetSearchParameter (const Tag & inTag, const DictEntry & inDictEntry, const std::string & inValue) [protected]

25.30.3.13 void gdcm::BaseRootQuery::SetSearchParameter (const Tag & inTag, const std::string & inValue)

25.30.3.14 `void gdcm::BaseRootQuery::SetSearchParameter (const std::string & inKeyword, const std::string & inValue)`

25.30.3.15 `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](#).

25.30.3.16 `const std::ostream& gdcm::BaseRootQuery::WriteHelpFile (std::ostream & os)`

25.30.3.17 `bool gdcm::BaseRootQuery::WriteQuery (const std::string & inFileName)`

25.30.4 Friends And Related Function Documentation

25.30.4.1 `friend class QueryFactory` `[friend]`

25.30.5 Member Data Documentation

25.30.5.1 `DataSet gdcm::BaseRootQuery::mDataSet` `[protected]`

25.30.5.2 `std::string gdcm::BaseRootQuery::mHelpDescription` `[protected]`

25.30.5.3 `QueryImage gdcm::BaseRootQuery::mImage` `[protected]`

25.30.5.4 `QueryPatient gdcm::BaseRootQuery::mPatient` `[protected]`

25.30.5.5 `ERootType gdcm::BaseRootQuery::mRootType` `[protected]`

25.30.5.6 `QuerySeries gdcm::BaseRootQuery::mSeries` `[protected]`

25.30.5.7 `QueryStudy gdcm::BaseRootQuery::mStudy` `[protected]`

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

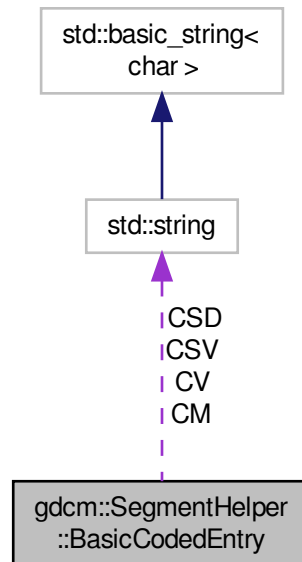
- [gdcmBaseRootQuery.h](#)

25.31 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 attributes of the basic coded entry is defined.

Public Attributes

- std::string [CM](#)
Coding Scheme [Version](#) attribute.
- std::string [CSD](#)
Code [Value](#) attribute.
- std::string [CSV](#)
Coding Scheme Designator attribute.
- std::string [CV](#)

25.31.1 Detailed Description

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

See Also

PS 3.3 section 8.8.

25.31.2 Constructor & Destructor Documentation

25.31.2.1 `gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry () [inline]`

Constructor.

25.31.2.2 `gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry (const char * a_CV, const char * a_CSD, const char * a_CM) [inline]`

constructor which defines type 1 attributes.

25.31.2.3 `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.

25.31.3 Member Function Documentation

25.31.3.1 `bool gdcm::SegmentHelper::BasicCodedEntry::IsEmpty (const bool checkOptionalAttributes = false) const`

Check if each attributes of the basic coded entry is defined.

Parameters

<i>checkOptional-Attributes</i>	Check also type 1C attributes.
---------------------------------	--------------------------------

25.31.4 Member Data Documentation

25.31.4.1 `std::string gdcm::SegmentHelper::BasicCodedEntry::CM`

Coding Scheme [Version](#) attribute.

25.31.4.2 `std::string gdcm::SegmentHelper::BasicCodedEntry::CSD`

Code [Value](#) attribute.

25.31.4.3 `std::string gdcm::SegmentHelper::BasicCodedEntry::CSV`

Coding Scheme Designator attribute.

25.31.4.4 std::string gdcm::SegmentHelper::BasicCodedEntry::CV

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

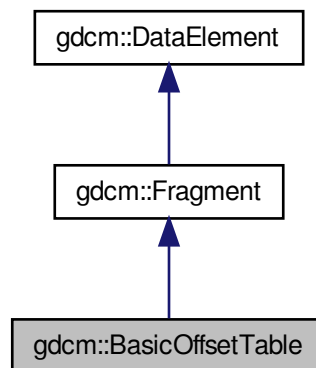
- [gdcmSegmentHelper.h](#)

25.32 gdcm::BasicOffsetTable Class Reference

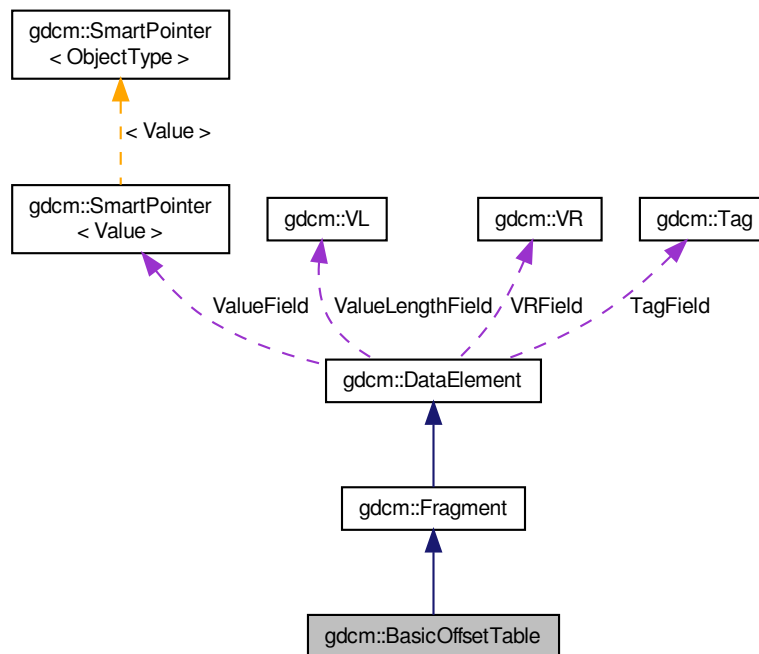
Class to represent a [BasicOffsetTable](#).

```
#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

25.32.1 Detailed Description

Class to represent a [BasicOffsetTable](#).

25.32.2 Constructor & Destructor Documentation

25.32.2.1 `gdcm::BasicOffsetTable::BasicOffsetTable ()` `[inline]`

25.32.3 Member Function Documentation

25.32.3.1 `template<typename TSwap> std::istream& gdcm::BasicOffsetTable::Read (std::istream & is)` `[inline]`

25.32.4 Friends And Related Function Documentation

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

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

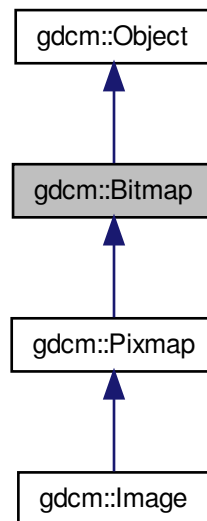
- [gdcmBasicOffsetTable.h](#)

25.33 gdcm::Bitmap Class Reference

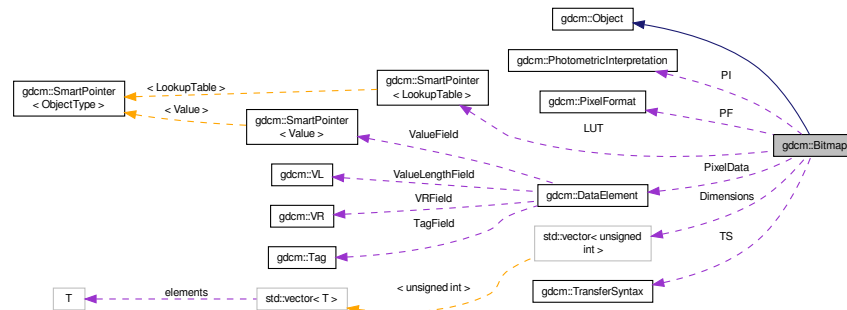
Bitmap class A bitmap based image. Used as parent for both IconImage and the main Pixel Data **Image** It does not contains any World Space information (IPP, IOP)

```
#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 [TryKAKADUCoec](#) (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](#)

25.33.1 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](#).

25.33.2 Member Typedef Documentation

25.33.2.1 `typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr` `[protected]`

25.33.3 Constructor & Destructor Documentation

25.33.3.1 `gdcm::Bitmap::Bitmap ()`

25.33.3.2 `gdcm::Bitmap::~~Bitmap ()`

25.33.4 Member Function Documentation

25.33.4.1 `virtual bool gdcm::Bitmap::AreOverlaysInPixelData () const` `[inline],[virtual]`

Reimplemented in [gdcm::Pixmap](#).

25.33.4.2 `void gdcm::Bitmap::Clear ()`

25.33.4.3 `bool gdcm::Bitmap::ComputeLossyFlag ()` `[protected]`

25.33.4.4 `bool gdcm::Bitmap::GetBuffer (char * buffer) const`

Acces the raw data.

Examples:

[ConvertToQImage.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

25.33.4.5 `bool gdcm::Bitmap::GetBuffer2 (std::ostream & os) const` `[protected]`

25.33.4.6 `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](#), [GenFakelImage.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

25.33.4.7 unsigned int gdcm::Bitmap::GetColumns () const [inline]

25.33.4.8 const DataElement& gdcm::Bitmap::GetDataElement () const [inline]

Examples:

[ExtractIconFromFile.cxx](#).

25.33.4.9 DataElement& gdcm::Bitmap::GetDataElement () [inline]

25.33.4.10 unsigned int gdcm::Bitmap::GetDimension (unsigned int *idx*) const

25.33.4.11 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](#).

25.33.4.12 const LookupTable& gdcm::Bitmap::GetLUT () const [inline]

Examples:

[ExtractIconFromFile.cxx](#).

25.33.4.13 LookupTable& gdcm::Bitmap::GetLUT () [inline]

25.33.4.14 bool gdcm::Bitmap::GetNeedByteSwap () const [inline]

25.33.4.15 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](#).

25.33.4.16 const PhotometricInterpretation& gdcm::Bitmap::GetPhotometricInterpretation () const

return the photometric interpretation

Examples:

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), and [HelloVizWorld.cxx](#).

25.33.4.17 `const PixelFormat& gdcm::Bitmap::GetPixelFormat () const` `[inline]`

Get/Set [PixelFormat](#).

Examples:

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAI BugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), and [threadgdcm.cxx](#).

25.33.4.18 `PixelFormat& gdcm::Bitmap::GetPixelFormat ()` `[inline]`

25.33.4.19 `unsigned int gdcm::Bitmap::GetPlanarConfiguration () const`

return the planar configuration

25.33.4.20 `unsigned int gdcm::Bitmap::GetRows () const` `[inline]`

25.33.4.21 `const TransferSyntax& gdcm::Bitmap::GetTransferSyntax () const` `[inline]`

Examples:

[ExtractIconFromFile.cxx](#).

25.33.4.22 `bool gdcm::Bitmap::IsEmpty () const` `[inline]`

25.33.4.23 `bool gdcm::Bitmap::IsLossy () const`

Return whether or not the image was compressed using a lossy compressor or not.

25.33.4.24 `bool gdcm::Bitmap::IsTransferSyntaxCompatible (TransferSyntax const & ts) const`

25.33.4.25 `void gdcm::Bitmap::Print (std::ostream &) const` `[virtual]`

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::Image](#), and [gdcm::Pixmap](#).

Examples:

[ExtractIconFromFile.cxx](#).

25.33.4.26 `void gdcm::Bitmap::SetColumns (unsigned int col)` `[inline]`

25.33.4.27 `void gdcm::Bitmap::SetDataElement (DataElement const & de)` `[inline]`

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakeImage.cxx](#), and [iU22tomultisc.cxx](#).

25.33.4.28 void gdcm::Bitmap::SetDimension (unsigned int *idx*, unsigned int *dim*)

Examples:

[csa2img.cxx](#), [GenFakelImage.cxx](#), and [iU22tomultisc.cxx](#).

25.33.4.29 void gdcm::Bitmap::SetDimensions (const unsigned int *dims*[3])

Examples:

[CreateARGBImage.cxx](#), and [CreateCMYKImage.cxx](#).

25.33.4.30 void gdcm::Bitmap::SetLossyFlag (bool *f*) [inline]

Specifically set that the image was compressed using a lossy compression mechanism.

25.33.4.31 void gdcm::Bitmap::SetLUT (LookupTable const & *lut*) [inline]

Set/Get LUT.

25.33.4.32 void gdcm::Bitmap::SetNeedByteSwap (bool *b*) [inline]

25.33.4.33 void gdcm::Bitmap::SetNumberOfDimensions (unsigned int *dim*)

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

25.33.4.34 void gdcm::Bitmap::SetPhotometricInterpretation (PhotometricInterpretation const & *pi*)

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), and [iU22tomultisc.cxx](#).

25.33.4.35 void gdcm::Bitmap::SetPixelFormat (PixelFormat const & *pf*) [inline]

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

References gdcm::PixelFormat::Validate().

25.33.4.36 void gdcm::Bitmap::SetPlanarConfiguration (unsigned int *pc*)

Warning

you need to call SetPixelFormat first (before SetPlanarConfiguration) for consistency checking

25.33.4.37 void gdcmm::Bitmap::SetRows (unsigned int *rows*) [inline]

25.33.4.38 void gdcmm::Bitmap::SetTransferSyntax (TransferSyntax const & *ts*) [inline]

Transfer syntax.

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [MergeTwoFiles.cxx](#).

25.33.4.39 bool gdcmm::Bitmap::TryJPEG2000Codec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.4.40 bool gdcmm::Bitmap::TryJPEG2000Codec2 (std::ostream & *os*) const [protected]

25.33.4.41 bool gdcmm::Bitmap::TryJPEGCodec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.4.42 bool gdcmm::Bitmap::TryJPEGCodec2 (std::ostream & *os*) const [protected]

25.33.4.43 bool gdcmm::Bitmap::TryJPEGLSCodec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.4.44 bool gdcmm::Bitmap::TryKAKADUCoec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.4.45 bool gdcmm::Bitmap::TryPVRGCodec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.4.46 bool gdcmm::Bitmap::TryRAWCodec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.4.47 bool gdcmm::Bitmap::TryRLECodec (char * *buffer*, bool & *lossyflag*) const [protected]

25.33.5 Friends And Related Function Documentation

25.33.5.1 friend class ImageChangeTransferSyntax [friend]

25.33.5.2 friend class PixmapReader [friend]

25.33.6 Member Data Documentation

25.33.6.1 std::vector<unsigned int> gdcmm::Bitmap::Dimensions [protected]

25.33.6.2 bool gdcmm::Bitmap::LossyFlag [protected]

25.33.6.3 LUTPtr gdcmm::Bitmap::LUT [protected]

25.33.6.4 bool gdcmm::Bitmap::NeedByteSwap [protected]

25.33.6.5 unsigned int gdcmm::Bitmap::NumberOfDimensions [protected]

25.33.6.6 PixelFormat gdcmm::Bitmap::PF [protected]

25.33.6.7 PhotometricInterpretation gdcmm::Bitmap::PI [protected]

25.33.6.8 DataElement gdcmm::Bitmap::PixelData [protected]

25.33.6.9 `unsigned int gdcm::Bitmap::PlanarConfiguration` `[protected]`

25.33.6.10 `TransferSyntax gdcm::Bitmap::TS` `[protected]`

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

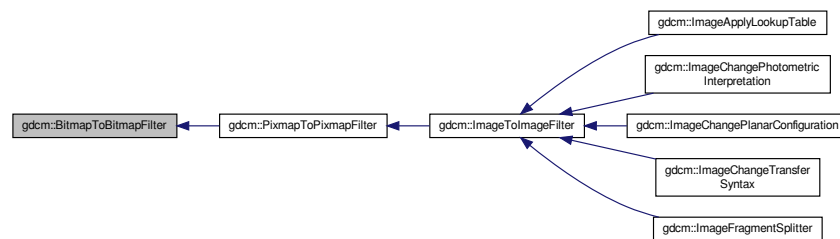
- [gdcmBitmap.h](#)

25.34 gdcm::BitmapToBitmapFilter Class Reference

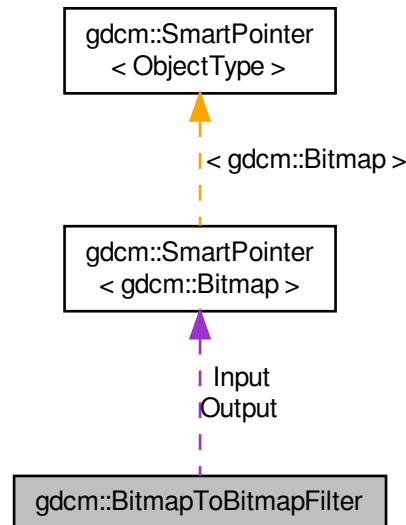
[BitmapToBitmapFilter](#) class Super class for all filter taking an image and producing an output image.

```
#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](#)

25.34.1 Detailed Description

[BitmapToBitmapFilter](#) class Super class for all filter taking an image and producing an output image.

25.34.2 Constructor & Destructor Documentation

25.34.2.1 `gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ()`

25.34.2.2 `gdcm::BitmapToBitmapFilter::~~BitmapToBitmapFilter ()` `[inline]`

25.34.3 Member Function Documentation

25.34.3.1 `const Bitmap& gdcm::BitmapToBitmapFilter::GetOutput () const` `[inline]`

Get Output image.

25.34.3.2 `const Bitmap& gdcm::BitmapToBitmapFilter::GetOutputAsBitmap () const`

25.34.3.3 `void gdcm::BitmapToBitmapFilter::SetInput (const Bitmap & image)`

Set input image.

Examples:

[CompressImage.cxx](#).

25.34.4 Member Data Documentation

25.34.4.1 `SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input` `[protected]`

25.34.4.2 `SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output` `[protected]`

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

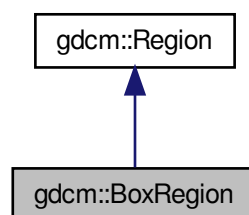
- [gdcmBitmapToBitmapFilter.h](#)

25.35 gdcm::BoxRegion Class Reference

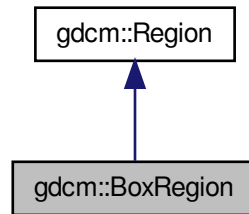
Class for manipulation box region This is a very simple implementation of the [Region](#) class. It only support 3D box type region. It assumes the 3D Box does not have a tilt Origin is as (0,0,0)

```
#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](#).

25.35.1 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)

25.35.2 Constructor & Destructor Documentation

25.35.2.1 `gdcm::BoxRegion::BoxRegion ()`

25.35.2.2 `gdcm::BoxRegion::~~BoxRegion ()`

25.35.2.3 `gdcm::BoxRegion::BoxRegion (const BoxRegion &)`

copy/cstor and al.

25.35.3 Member Function Documentation

25.35.3.1 `size_t gdcm::BoxRegion::Area () const` [virtual]

compute the area

Implements [gdcm::Region](#).

25.35.3.2 `static BoxRegion gdcm::BoxRegion::BoundingBox (BoxRegion const & b1, BoxRegion const & b2)`
[static]

Helper class to compute the bounding box of two [BoxRegion](#).

25.35.3.3 `Region* gdcm::BoxRegion::Clone () const` [virtual]

Implements [gdcm::Region](#).

25.35.3.4 `BoxRegion gdcm::BoxRegion::ComputeBoundingBox ()` [virtual]

Return the Axis-Aligned minimum bounding box for all regions.

Implements [gdcm::Region](#).

25.35.3.5 `bool gdcm::BoxRegion::Empty () const` [virtual]

return whether this domain is empty:

Implements [gdcm::Region](#).

25.35.3.6 `unsigned int gdcm::BoxRegion::GetXMax () const`

25.35.3.7 `unsigned int gdcm::BoxRegion::GetXMin () const`

Get domain.

25.35.3.8 unsigned int `gdcm::BoxRegion::GetYMax ()` const

25.35.3.9 unsigned int `gdcm::BoxRegion::GetYMin ()` const

25.35.3.10 unsigned int `gdcm::BoxRegion::GetZMax ()` const

25.35.3.11 unsigned int `gdcm::BoxRegion::GetZMin ()` const

25.35.3.12 bool `gdcm::BoxRegion::IsValid ()` const [virtual]

return whether this is valid domain

Implements [gdcm::Region](#).

25.35.3.13 void `gdcm::BoxRegion::operator= (const BoxRegion &)`

25.35.3.14 void `gdcm::BoxRegion::Print (std::ostream & os = std::cout)` const [virtual]

Print.

Reimplemented from [gdcm::Region](#).

25.35.3.15 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](#)

25.36 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](#) ()

25.36.1 Detailed Description

[ByteBuffer](#).

Detailed description here

Note

looks like a std::streambuf or std::filebuf class with the get and peek pointer

25.36.2 Constructor & Destructor Documentation

25.36.2.1 gdcmm::ByteBuffer::ByteBuffer () [inline]

25.36.3 Member Function Documentation

25.36.3.1 char* gdcmm::ByteBuffer::Get (int *len*) [inline]

25.36.3.2 const char* gdcmm::ByteBuffer::GetStart () const [inline]

25.36.3.3 void gdcmm::ByteBuffer::ShiftEnd (int *len*) [inline]

25.36.3.4 void gdcmm::ByteBuffer::UpdatePosition () [inline]

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

- [gdcmmByteBuffer.h](#)

25.37 gdcmm::ByteSwap< T > Class Template Reference

[ByteSwap.](#)

```
#include <gdcmmByteSwap.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](#) ()

25.37.1 Detailed Description

```
template<class T>class gdcmm::ByteSwap< T >
```

[ByteSwap.](#)

Perform machine dependent byte swapping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian). TODO: bswap_32 / bswap_64 ...

Examples:

[TestByteSwap.cxx.](#)

25.37.2 Member Function Documentation

25.37.2.1 `template<class T> static void gdcm::ByteSwap<T>::Swap (T & p) [static]`

25.37.2.2 `template<class T> static void gdcm::ByteSwap<T>::SwapFromSwapCodeIntoSystem (T & p, SwapCode const & sc) [static]`

Examples:

[TestByteSwap.cxx](#).

25.37.2.3 `template<class T> static void gdcm::ByteSwap<T>::SwapRange (T * p, unsigned int num) [static]`

25.37.2.4 `template<class T> static void gdcm::ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem (T * p, SwapCode const & sc, std::streamoff num) [static]`

Examples:

[TestByteSwap.cxx](#).

25.37.2.5 `template<class T> static bool gdcm::ByteSwap<T>::SystemIsBigEndian () [static]`

Query the machine Endian-ness.

25.37.2.6 `template<class T> static bool gdcm::ByteSwap<T>::SystemIsLittleEndian () [static]`

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

- [gdcmByteSwap.h](#)

25.38 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)

25.38.1 Detailed Description

[ByteSwapFilter](#) In place byte-swapping of a dataset FIXME: FL status ??

25.38.2 Constructor & Destructor Documentation

25.38.2.1 `gdcm::ByteSwapFilter::ByteSwapFilter (DataSet & ds)` `[inline]`

25.38.2.2 `gdcm::ByteSwapFilter::~~ByteSwapFilter ()`

25.38.3 Member Function Documentation

25.38.3.1 `bool gdcm::ByteSwapFilter::ByteSwap ()`

25.38.3.2 `void gdcm::ByteSwapFilter::SetByteSwapTag (bool b)` `[inline]`

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

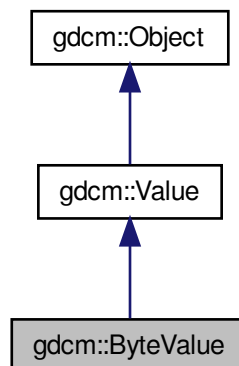
- [gdcmByteSwapFilter.h](#)

25.39 gdcm::ByteValue Class Reference

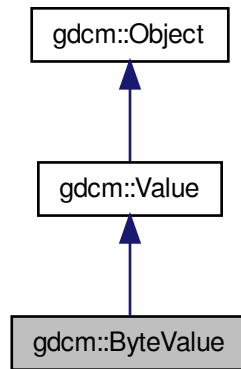
Class to represent binary value (array of bytes)

```
#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 Clear` ()
- `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) / dont 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 PrintGroupLength` (`std::ostream &os`)
- `void PrintHex` (`std::ostream &os, VL maxlength`) `const`
- `template<typename TSwap, typename TType >`
`std::istream & Read` (`std::istream &is`)
- `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

25.39.1 Detailed Description

Class to represent binary value (array of bytes)

Note

Examples:

[DumpADAC.cxx](#), [DumpImageHeaderInfo.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](#).

25.39.2 Constructor & Destructor Documentation

25.39.2.1 `gdcm::ByteValue::ByteValue (const char * array = 0, VL const & vl = 0)` `[inline]`

References `gdcmDebugMacro`.

25.39.2.2 `gdcm::ByteValue::ByteValue (std::vector< char > & v)` `[inline]`

Warning

casting to `uint32_t`

25.39.2.3 `gdcm::ByteValue::~~ByteValue ()` `[inline]`

25.39.3 Member Function Documentation

25.39.3.1 `void gdcm::ByteValue::Clear ()` `[inline]`, `[virtual]`

Implements `gdcm::Value`.

25.39.3.2 `void gdcm::ByteValue::Fill (char c)` `[inline]`

Examples:

[DuplicatePCDE.cxx](#).

25.39.3.3 `bool gdcm::ByteValue::GetBuffer (char * buffer, unsigned long length) const`

Examples:

[FixJAIBugJPEGLS.cxx](#).

25.39.3.4 `VL gdcmm::ByteValue::GetLength () const [inline],[virtual]`

Implements [gdcmm::Value](#).

Examples:

[DumpADAC.cxx](#), [DumpImageHeaderInfo.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAI BugJPEGLS.cxx](#), [GetSubSequenceData.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

Referenced by `gdcmm::operator<<()`, `gdcmm::SequenceOfFragments::ReadValue()`, `gdcmm::Element< VR::OB, VM::VM1_n >::Set()`, `gdcmm::Element< TVR, VM::VM1_n >::Set()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap()`, `gdcmm::Element< VR::OB, VM::VM1_n >::SetNoSwap()`, `gdcmm::Element< TVR, VM::VM1_n >::SetNoSwap()`, and `gdcmm::Fragment::Write()`.

25.39.3.5 `const char* gdcmm::ByteValue::GetPointer () const [inline]`

Examples:

[DumpADAC.cxx](#), [DumpImageHeaderInfo.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 `gdcmm::operator<<()`, `gdcmm::SequenceOfFragments::ReadValue()`, `gdcmm::Element< VR::OB, VM::VM1_n >::Set()`, `gdcmm::Element< TVR, VM::VM1_n >::Set()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap()`, `gdcmm::Element< VR::OB, VM::VM1_n >::SetNoSwap()`, and `gdcmm::Element< TVR, VM::VM1_n >::SetNoSwap()`.

25.39.3.6 `bool gdcmm::ByteValue::IsEmpty () const [inline]`

25.39.3.7 `bool gdcmm::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 dont think this function is working since it does not handle UNICODE or character set...

25.39.3.8 `gdcmm::ByteValue::operator const std::vector< char > & () const [inline]`

25.39.3.9 `ByteValue& gdcmm::ByteValue::operator= (const ByteValue & val) [inline]`

25.39.3.10 `bool gdcmm::ByteValue::operator== (const ByteValue & val) const [inline]`

25.39.3.11 `bool gdcmm::ByteValue::operator== (const Value & val) const [inline],[virtual]`

Implements [gdcmm::Value](#).

25.39.3.12 `void gdcmm::ByteValue::Print (std::ostream & os) const [inline],[protected],[virtual]`

Reimplemented from [gdcmm::Object](#).

25.39.3.13 void gdcm::ByteValue::PrintASCII (std::ostream & *os*, VL *maxlength*) const

25.39.3.14 void gdcm::ByteValue::PrintGroupLength (std::ostream & *os*) [inline]

25.39.3.15 void gdcm::ByteValue::PrintHex (std::ostream & *os*, VL *maxlength*) const

25.39.3.16 template<typename TSwap, typename TType > std::istream& gdcm::ByteValue::Read (std::istream & *is*)
[inline]

25.39.3.17 template<typename TSwap > std::istream& gdcm::ByteValue::Read (std::istream & *is*) [inline]

25.39.3.18 void gdcm::ByteValue::SetLength (VL *vl*) [inline],[virtual]

Implements [gdcm::Value](#).

References [gdcmDebugMacro](#), [gdcm::VL::IsOdd\(\)](#), and [gdcm::VL::IsUndefined\(\)](#).

25.39.3.19 template<typename TSwap, typename TType > std::ostream const& gdcm::ByteValue::Write (std::ostream & *os*)
const [inline]

Referenced by [gdcm::Fragment::Write\(\)](#).

25.39.3.20 template<typename TSwap > std::ostream const& gdcm::ByteValue::Write (std::ostream & *os*) const [inline]

25.39.3.21 bool gdcm::ByteValue::WriteBuffer (std::ostream & *os*) const [inline]

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

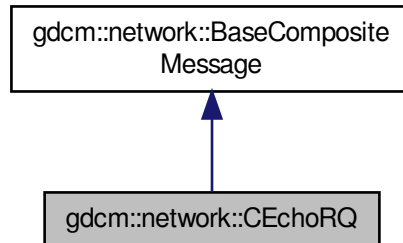
- [gdcmByteValue.h](#)

25.40 gdcm::network::CEchoRQ Class Reference

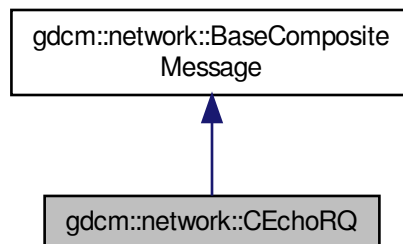
[CEchoRQ](#) this file defines the messages for the cecho action.

```
#include <gdcmCEchoMessages.h>
```

Inheritance diagram for `gdcmm::network::CEchoRQ`:



Collaboration diagram for `gdcmm::network::CEchoRQ`:



Public Member Functions

- `std::vector< PresentationDataValue > ConstructPDV` (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)

Public Attributes

- [UIComp AffectedSOPClassUID](#)
- `uint16_t` [MessageID](#)

25.40.1 Detailed Description

[CEchoRQ](#) this file defines the messages for the cecho action.

25.40.2 Member Function Documentation

25.40.2.1 `std::vector<PresentationDataValue> gdcm::network::CEchoRQ::ConstructPDV (const ULConnection & inConnection, const BaseRootQuery * inRootQuery) [virtual]`

Implements [gdcm::network::BaseCompositeMessage](#).

25.40.3 Member Data Documentation

25.40.3.1 `UIComp` `gdcm::network::CEchoRQ::AffectedSOPClassUID`

25.40.3.2 `uint16_t` `gdcm::network::CEchoRQ::MessageID`

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

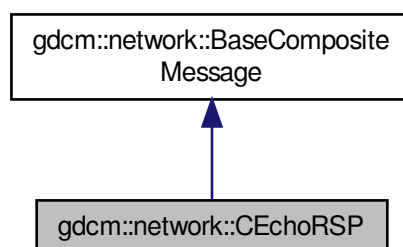
- [gdcmCEchoMessages.h](#)
- [gdcmDIMSE.h](#)

25.41 gdcm::network::CEchoRSP Class Reference

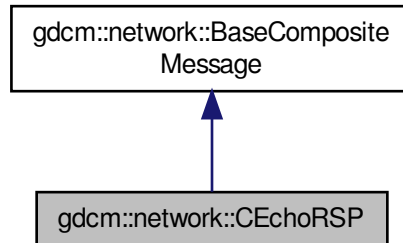
[CEchoRSP](#) this file defines the messages for the cecho action.

```
#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)`

25.41.1 Detailed Description

`CEchoRSP` this file defines the messages for the cecho action.

25.41.2 Member Function Documentation

25.41.2.1 `std::vector<PresentationDataValue> gdcm::network::CEchoRSP::ConstructPDVByDataSet (const DataSet *inDataSet)`

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

- `gdcmCEchoMessages.h`

25.42 gdcm::network::CFind Class Reference

```
#include <gdcmDIMSE.h>
```

25.42.1 Detailed Description

PS 3.4 - 2009 [Table B.2-1](#) C-STORE STATUS

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

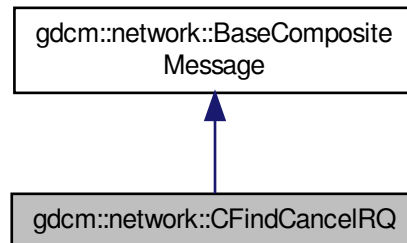
- `gdcmDIMSE.h`

25.43 gdcmm::network::CFindCancelRQ Class Reference

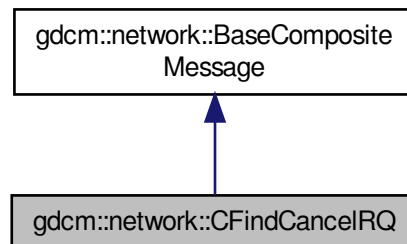
[CFindCancelRQ](#) this file defines the messages for the cfind action.

```
#include <gdcmmCFindMessages.h>
```

Inheritance diagram for gdcmm::network::CFindCancelRQ:



Collaboration diagram for gdcmm::network::CFindCancelRQ:



Public Member Functions

- `std::vector`
 < [PresentationDataValue](#) > [ConstructPDVByDataSet](#) (const [DataSet](#) *inDataSet)

25.43.1 Detailed Description

[CFindCancelRQ](#) this file defines the messages for the cfind action.

25.43.2 Member Function Documentation

25.43.2.1 `std::vector<PresentationDataValue> gdcm::network::CFindCancelRQ::ConstructPDVByDataSet (const DataSet * inDataSet)`

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

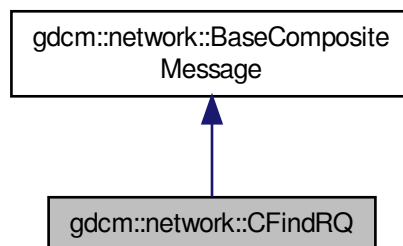
- [gdcmCFindMessages.h](#)

25.44 gdcm::network::CFindRQ Class Reference

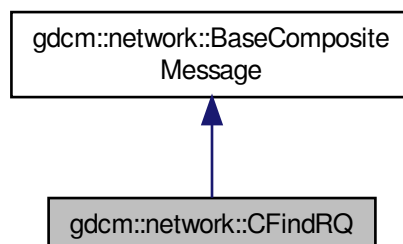
[CFindRQ](#) this file defines the messages for the cfind action.

```
#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 *inRootQuery)`

25.44.1 Detailed Description

[CFindRQ](#) this file defines the messages for the cfind action.

25.44.2 Member Function Documentation

25.44.2.1 `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:

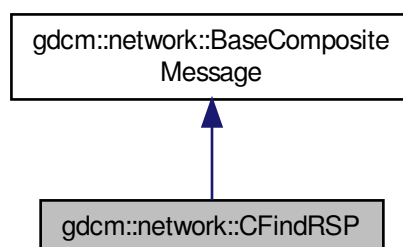
- [gdcmCFindMessages.h](#)

25.45 gdcm::network::CFindRSP Class Reference

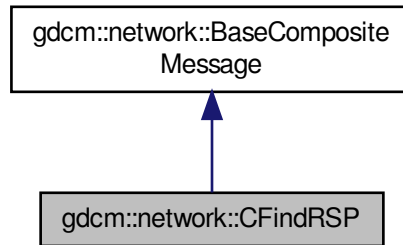
[CFindRSP](#) this file defines the messages for the cfind action.

```
#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)`

25.45.1 Detailed Description

[CFindRSP](#) this file defines the messages for the cfind action.

25.45.2 Member Function Documentation

25.45.2.1 `std::vector<PresentationDataValue> gdcm::network::CFindRSP::ConstructPDVByDataSet (const DataSet * inDataSet)`

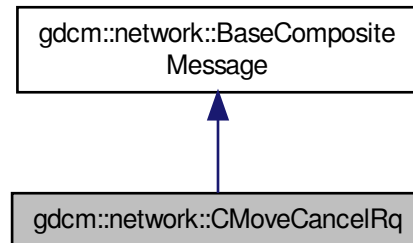
The documentation for this class was generated from the following file:

- [gdcmCFindMessages.h](#)

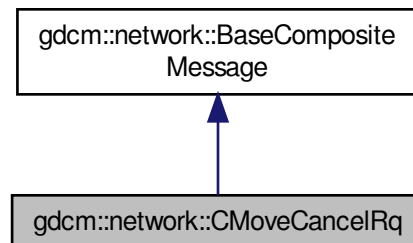
25.46 gdcm::network::CMoveCancelRq Class Reference

```
#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)`

25.46.1 Member Function Documentation

25.46.1.1 `std::vector<PresentationDataValue> gdcm::network::CMoveCancelRq::ConstructPDVByDataSet (const DataSet * inDataSet)`

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

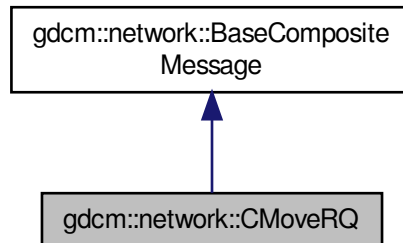
- [gdcmCMoveMessages.h](#)

25.47 gdcmm::network::CMoveRQ Class Reference

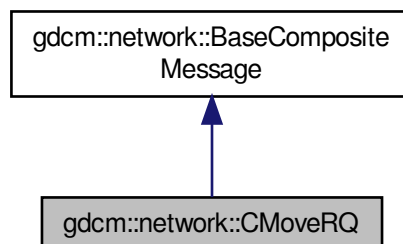
[CMoveRQ](#) this file defines the messages for the cmove action.

```
#include <gdcmmCMoveMessages.h>
```

Inheritance diagram for gdcmm::network::CMoveRQ:



Collaboration diagram for gdcmm::network::CMoveRQ:



Public Member Functions

- std::vector< [PresentationDataValue](#) > [ConstructPDV](#) (const [ULConnection](#) &inConnection, const [BaseRootQuery](#) *inRootQuery)

25.47.1 Detailed Description

[CMoveRQ](#) this file defines the messages for the cmove action.

25.47.2 Member Function Documentation

25.47.2.1 `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:

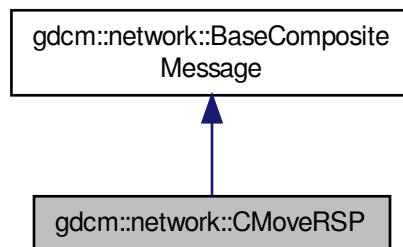
- [gdcmCMoveMessages.h](#)

25.48 gdcm::network::CMoveRSP Class Reference

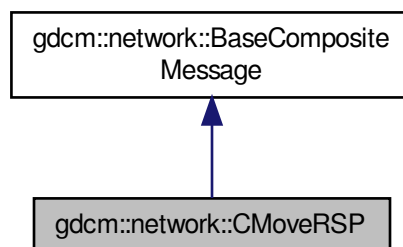
[CMoveRSP](#) this file defines the messages for the cmove action.

```
#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)`

25.48.1 Detailed Description

[CMoveRSP](#) this file defines the messages for the cmove action.

25.48.2 Member Function Documentation

25.48.2.1 `std::vector<PresentationDataValue> gdcm::network::CMoveRSP::ConstructPDVByDataSet (const DataSet * inDataSet)`

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

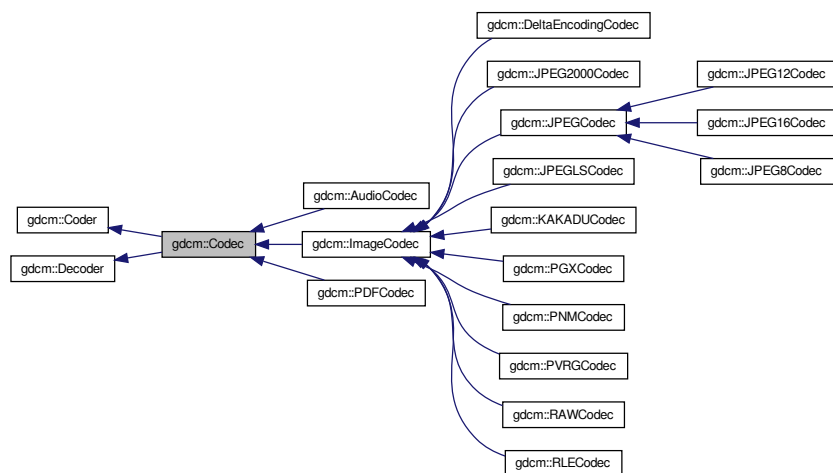
- [gdcmCMoveMessages.h](#)

25.49 gdcm::Codec Class Reference

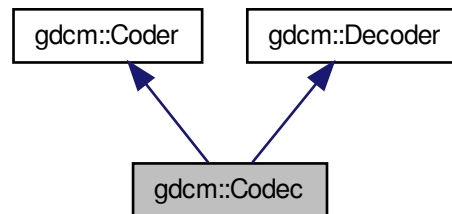
[Codec](#) class.

```
#include <gdcmCodec.h>
```

Inheritance diagram for `gdcm::Codec`:



Collaboration diagram for gdcm::Codec:



Additional Inherited Members

25.49.1 Detailed Description

[Codec](#) class.

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

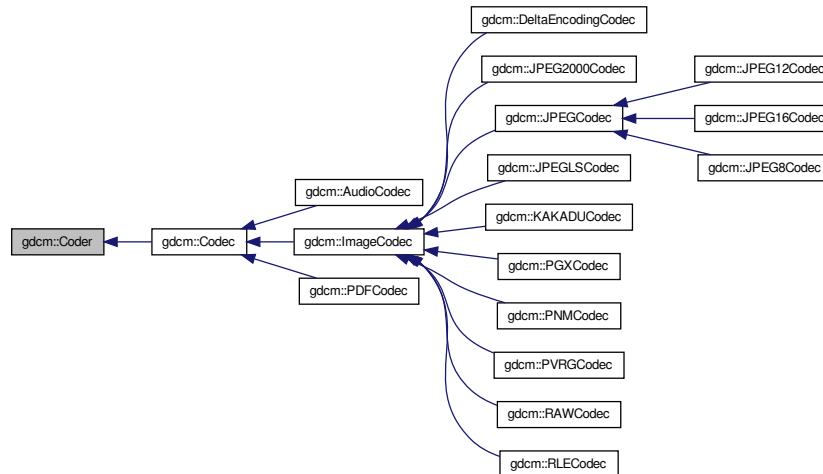
- [gdcmCodec.h](#)

25.50 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)

25.50.1 Detailed Description

`Coder`.

25.50.2 Constructor & Destructor Documentation

25.50.2.1 virtual `gdcm::Coder::~Coder` () [inline], [virtual]

25.50.3 Member Function Documentation

25.50.3.1 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::JPEGLSCodec`, `gdcm::ImageCodec`, `gdcm::PNMCodec`, `gdcm::PGXCodec`, `gdcm::KAKADUCODEC`, `gdcm::RAWCodec`, `gdcm::AudioCodec`, and `gdcm::PDFCodec`.

25.50.3.2 `virtual bool gdcm::Coder::Code (DataElement const & in_, DataElement & out_) [inline], [virtual]`

Code.

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::JPEGLSCodec](#), [gdcm::PVRGCodec](#), [gdcm::JPE-G2000Codec](#), [gdcm::KAKADUCodec](#), and [gdcm::RAWCodec](#).

25.50.3.3 `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](#)

25.51 gdcm::CodeString Class Reference

[CodeString](#) This is an implementation of DICOM VR: CS The ctor 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)`

25.51.1 Detailed Description

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

Note

the ctor 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.

25.51.2 Member Typedef Documentation

25.51.2.1 `typedef InternalClass::const_iterator gdcm::CodeString::const_iterator`

25.51.2.2 `typedef InternalClass::const_reference gdcm::CodeString::const_reference`

25.51.2.3 `typedef InternalClass::const_reverse_iterator gdcm::CodeString::const_reverse_iterator`

25.51.2.4 `typedef InternalClass::difference_type gdcm::CodeString::difference_type`

25.51.2.5 `typedef InternalClass::iterator gdcm::CodeString::iterator`

25.51.2.6 `typedef InternalClass::pointer gdcm::CodeString::pointer`

25.51.2.7 `typedef InternalClass::reference gdcm::CodeString::reference`

25.51.2.8 `typedef InternalClass::reverse_iterator gdcm::CodeString::reverse_iterator`

25.51.2.9 `typedef InternalClass::size_type gdcm::CodeString::size_type`

25.51.2.10 `typedef InternalClass::value_type gdcm::CodeString::value_type`

25.51.3 Constructor & Destructor Documentation

25.51.3.1 `gdcm::CodeString::CodeString () [inline]`

[CodeString](#) constructors.

25.51.3.2 `gdcm::CodeString::CodeString (const value_type * s) [inline]`

25.51.3.3 `gdcm::CodeString::CodeString (const value_type * s, size_type n) [inline]`

25.51.3.4 `gdcm::CodeString::CodeString (const InternalClass & s, size_type pos = 0, size_type n = InternalClass::npos) [inline]`

25.51.4 Member Function Documentation

25.51.4.1 `std::string gdcm::CodeString::GetAsString () const [inline]`

Return the full code string as std::string.

25.51.4.2 `bool gdcm::CodeString::IsValid () const`

Check if [CodeString](#) obj is correct..

25.51.4.3 `size_type gdcm::CodeString::Size () const [inline]`

Return the size of the string.

25.51.4.4 `std::string gdcm::CodeString::TrimInternal () const [inline],[protected]`

25.51.5 Friends And Related Function Documentation

25.51.5.1 `bool operator!= (const CodeString & ref, const CodeString & cs) [friend]`

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

25.51.5.3 `bool operator== (const CodeString & ref, const CodeString & cs) [friend]`

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

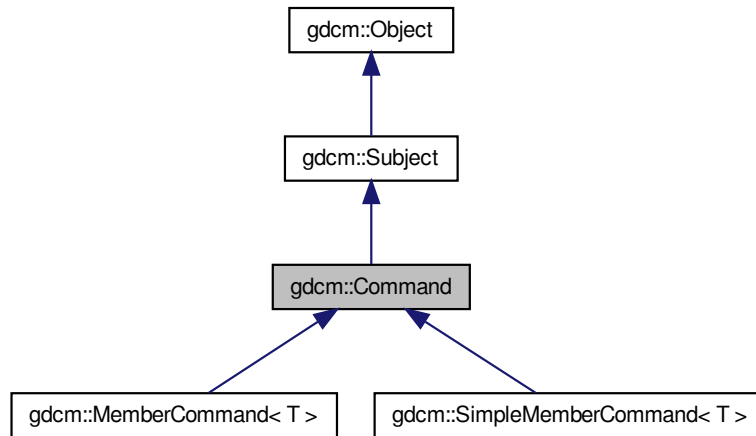
- [gdcmCodeString.h](#)

25.52 gdcm::Command Class Reference

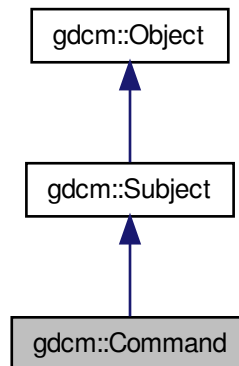
[Command](#) superclass for callback/observer methods.

```
#include <gdcCommand.h>
```

Inheritance diagram for gdc::Command:



Collaboration diagram for gdc::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\(\)](#)

25.52.1 Detailed Description

[Command](#) superclass for callback/observer methods.

See Also

[Subject](#)

25.52.2 Constructor & Destructor Documentation

25.52.2.1 `gdcm::Command::Command()` [protected]

25.52.2.2 `gdcm::Command::~~Command()` [protected]

25.52.3 Member Function Documentation

25.52.3.1 `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 >](#).

25.52.3.2 `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:

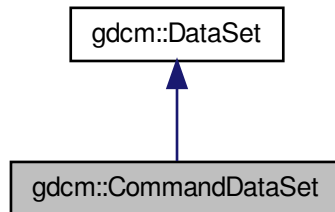
- [gdcmCommand.h](#)

25.53 gdcm::CommandDataSet Class Reference

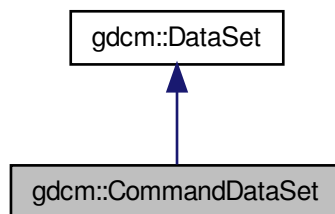
Class to represent a [Command DataSet](#).

```
#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

25.53.1 Detailed Description

Class to represent a [Command DataSet](#).

See Also

[DataSet](#)

25.53.2 Constructor & Destructor Documentation

25.53.2.1 `gdcm::CommandDataSet::CommandDataSet ()` `[inline]`

25.53.2.2 `gdcm::CommandDataSet::~~CommandDataSet ()` `[inline]`

25.53.3 Member Function Documentation

25.53.3.1 `void gdcm::CommandDataSet::Insert (const DataElement & de)` `[inline]`

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

25.53.3.2 `std::istream& gdcm::CommandDataSet::Read (std::istream & is)`

Read.

25.53.3.3 `void gdcm::CommandDataSet::Replace (const DataElement & de)` `[inline]`

References `gdcm::DataElement::GetTag()`.

25.53.3.4 `std::ostream& gdcm::CommandDataSet::Write (std::ostream & os) const`

Write.

25.53.4 Friends And Related Function Documentation

25.53.4.1 `std::ostream& operator<< (std::ostream & _os, const CommandDataSet & _val)` `[friend]`

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

- [gdcmCommandDataSet.h](#)

25.54 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)

25.54.1 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).

25.54.2 Member Function Documentation

- 25.54.2.1 static std::vector<[PresentationDataValue](#)> [gdcm::network::CompositeMessageFactory::ConstructCEchoRQ](#) (const [ULConnection](#) & *inConnection*) [static]
- 25.54.2.2 static std::vector<[PresentationDataValue](#)> [gdcm::network::CompositeMessageFactory::ConstructCFindRQ](#) (const [ULConnection](#) & *inConnection*, const [BaseRootQuery](#) * *inRootQuery*) [static]
- 25.54.2.3 static std::vector<[PresentationDataValue](#)> [gdcm::network::CompositeMessageFactory::ConstructCMoveRQ](#) (const [ULConnection](#) & *inConnection*, const [BaseRootQuery](#) * *inRootQuery*) [static]
- 25.54.2.4 static std::vector<[PresentationDataValue](#)> [gdcm::network::CompositeMessageFactory::ConstructCStoreRQ](#) (const [ULConnection](#) & *inConnection*, const [File](#) & *file*) [static]
- 25.54.2.5 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](#)

25.55 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)

25.55.1 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)

25.55.2 Member Typedef Documentation

25.55.2.1 typedef std::vector< [KeyValuePairType](#) > gdcm::CompositeNetworkFunctions::KeyValuePairArrayType

25.55.2.2 `typedef std::pair<Tag, std::string> gdcmm::CompositeNetworkFunctions::KeyValuePairType`

25.55.3 Member Function Documentation

25.55.3.1 `static bool gdcmm::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

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

25.55.3.2 `static bool gdcmm::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

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0

Returns

true if it worked.

25.55.3.3 `static bool gdcmm::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

<i>aetitle</i>	when not set will default to 'GDCMSCU'
----------------	--

<i>call</i>	when not set will default to 'ANY-SCP' This is an error to set remote to NULL or portno to 0 when
<i>outputdir</i>	is not set default to current dir ('.')

Returns

true if it worked.

25.55.3.4 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

<i>inMove</i>).	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)
------------------	--

25.55.3.5 static BaseRootQuery* gdcm::CompositeNetworkFunctions::ConstructQuery (ERootType *inRootType*, EQueryLevel *inQueryLevel*, const KeyValuePairArrayType & *keys*, bool *inMove* = false) [static]

Deprecated

25.55.3.6 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

<i>aetitle</i>	when not set will default to 'GDCMSCU'
<i>call</i>	when not set will default to 'ANY-SCP' 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](#)

25.56 gdcm::ConstCharWrapper Class Reference

Do not use me.

```
#include <gdcmConstCharWrapper.h>
```

Public Member Functions

- [ConstCharWrapper](#) (const char *i=0)
- [operator const char * \(\)](#) const

25.56.1 Detailed Description

Do not use me.

25.56.2 Constructor & Destructor Documentation

25.56.2.1 `gdcm::ConstCharWrapper::ConstCharWrapper (const char * i = 0) [inline]`

25.56.3 Member Function Documentation

25.56.3.1 `gdcm::ConstCharWrapper::operator const char * () const [inline]`

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

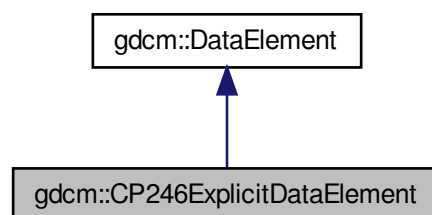
- [gdcmConstCharWrapper.h](#)

25.57 gdcm::CP246ExplicitDataElement Class Reference

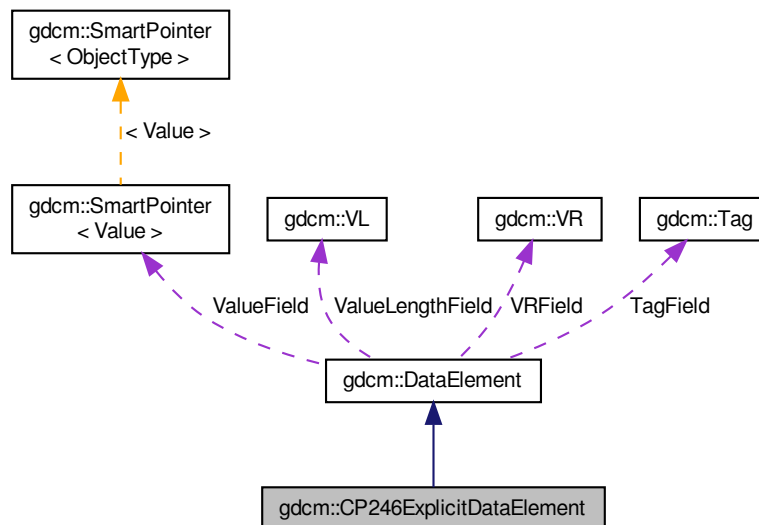
Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).

```
#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)
- template<typename TSwap >
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

Additional Inherited Members

25.57.1 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

25.57.2 Member Function Documentation

25.57.2.1 VL gdcm::CP246ExplicitDataElement::GetLength () const

25.57.2.2 `template<typename TSwap> std::istream& gdcmm::CP246ExplicitDataElement::Read (std::istream & is)`

25.57.2.3 `template<typename TSwap> std::istream& gdcmm::CP246ExplicitDataElement::ReadPreValue (std::istream & is)`

25.57.2.4 `template<typename TSwap> std::istream& gdcmm::CP246ExplicitDataElement::ReadValue (std::istream & is)`

25.57.2.5 `template<typename TSwap> std::istream& gdcmm::CP246ExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

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

- [gdcmmCP246ExplicitDataElement.h](#)

25.58 gdcmm::CryptographicMessageSyntax Class Reference

Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

```
#include <gdcmmCryptographicMessageSyntax.h>
```

Public Types

- enum [CipherTypes](#) {
[DES_CIPHER](#),
[DES3_CIPHER](#),
[AES128_CIPHER](#),
[AES192_CIPHER](#),
[AES256_CIPHER](#) }

Public Member Functions

- [CryptographicMessageSyntax](#) ()
- [~CryptographicMessageSyntax](#) ()
- 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)

25.58.1 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

Examples:

[BasicAnonymizer.cs](#), and [ClinicalTrialIdentificationWorkflow.cs](#).

25.58.2 Member Enumeration Documentation

25.58.2.1 enum gdcM::CryptographicMessageSyntax::CipherTypes

Enumerator

DES_CIPHER
DES3_CIPHER
AES128_CIPHER
AES192_CIPHER
AES256_CIPHER

25.58.3 Constructor & Destructor Documentation

25.58.3.1 gdcM::CryptographicMessageSyntax::CryptographicMessageSyntax ()

25.58.3.2 gdcM::CryptographicMessageSyntax::~~CryptographicMessageSyntax ()

25.58.4 Member Function Documentation

25.58.4.1 bool gdcM::CryptographicMessageSyntax::Decrypt (char * *output*, size_t & *outlen*, const char * *array*, size_t *len*) const

decrypt content from a PKCS#7 envelopedData structure

25.58.4.2 bool gdcM::CryptographicMessageSyntax::Encrypt (char * *output*, size_t & *outlen*, const char * *array*, size_t *len*) const

create a PKCS#7 envelopedData structure

25.58.4.3 CipherTypes gdcM::CryptographicMessageSyntax::GetCipherType () const

25.58.4.4 bool gdcM::CryptographicMessageSyntax::ParseCertificateFile (const char * *filename*)

25.58.4.5 bool gdcM::CryptographicMessageSyntax::ParseKeyFile (const char * *filename*)

25.58.4.6 void gdcM::CryptographicMessageSyntax::SetCipherType (CipherTypes *type*)

Set Cipher [Type](#). Default is: AES256_CIPHER

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

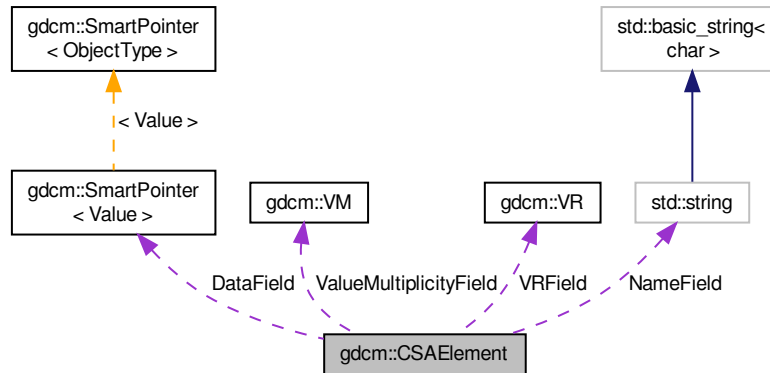
- [gdcMCryptographicMessageSyntax.h](#)

25.59 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)

25.59.1 Detailed Description

Class to represent a CSA [Element](#).

See Also

[CSAHeader](#)

Examples:

[csa2img.cxx](#), and [MrProtocol.cxx](#).

25.59.2 Member Typedef Documentation

25.59.2.1 typedef [SmartPointer](#)<[Value](#)> [gdcm::CSAElement::DataPtr](#) [protected]

25.59.3 Constructor & Destructor Documentation

25.59.3.1 [gdcm::CSAElement::CSAElement](#) (unsigned int *kf* = 0) [inline]

25.59.3.2 [gdcm::CSAElement::CSAElement](#) (const [CSAElement](#) &*_val*) [inline]

25.59.4 Member Function Documentation

25.59.4.1 `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](#).

25.59.4.2 `unsigned int gdcm::CSAElement::GetKey () const` [inline]

Set/Get Key.

Referenced by operator<().

25.59.4.3 `const char* gdcm::CSAElement::GetName () const` [inline]

Set/Get Name.

25.59.4.4 `unsigned int gdcm::CSAElement::GetNoOfItems () const` [inline]

Set/Get NoOfItems.

25.59.4.5 `unsigned int gdcm::CSAElement::GetSyngoDT () const` [inline]

Set/Get SyngoDT.

25.59.4.6 `Value const& gdcm::CSAElement::GetValue () const` [inline]

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

Examples:

[csa2img.cxx](#).

25.59.4.7 `Value& gdcm::CSAElement::GetValue ()` [inline]

25.59.4.8 `const VM& gdcm::CSAElement::GetVM () const` [inline]

Set/Get [VM](#).

25.59.4.9 `VR const& gdcm::CSAElement::GetVR () const` [inline]

Set/Get [VR](#).

25.59.4.10 `bool gdcm::CSAElement::IsEmpty () const [inline]`

Check if CSA [Element](#) is empty.

Examples:

[csa2img.cxx](#).

25.59.4.11 `bool gdcm::CSAElement::operator< (const CSAElement & de) const [inline]`

References [GetKey\(\)](#).

25.59.4.12 `CSAElement& gdcm::CSAElement::operator= (const CSAElement & de) [inline]`

References [DataField](#), [KeyField](#), [NameField](#), [NoOfItemsField](#), [SyngoDTField](#), [ValueMultiplicityField](#), and [VRField](#).

25.59.4.13 `bool gdcm::CSAElement::operator== (const CSAElement & de) const [inline]`

References [KeyField](#), [NameField](#), [SyngoDTField](#), [ValueMultiplicityField](#), and [VRField](#).

25.59.4.14 `void gdcm::CSAElement::SetByteValue (const char * array, VL length) [inline]`

Set.

25.59.4.15 `void gdcm::CSAElement::SetKey (unsigned int key) [inline]`

25.59.4.16 `void gdcm::CSAElement::SetName (const char * name) [inline]`

25.59.4.17 `void gdcm::CSAElement::SetNoOfItems (unsigned int items) [inline]`

25.59.4.18 `void gdcm::CSAElement::SetSyngoDT (unsigned int syngodt) [inline]`

25.59.4.19 `void gdcm::CSAElement::SetValue (Value const & vl) [inline]`

25.59.4.20 `void gdcm::CSAElement::SetVM (const VM & vm) [inline]`

25.59.4.21 `void gdcm::CSAElement::SetVR (VR const & vr) [inline]`

25.59.5 Friends And Related Function Documentation

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

25.59.6 Member Data Documentation

25.59.6.1 `DataPtr gdcm::CSAElement::DataField [protected]`

Referenced by [gdcm::operator<<\(\)](#), and [operator=\(\)](#).

25.59.6.2 `unsigned int gdcmm::CSAElement::KeyField` [protected]

Referenced by `gdcmm::operator<<()`, `operator=()`, and `operator==()`.

25.59.6.3 `std::string gdcmm::CSAElement::NameField` [protected]

Referenced by `gdcmm::operator<<()`, `operator=()`, and `operator==()`.

25.59.6.4 `unsigned int gdcmm::CSAElement::NoOfItemsField` [protected]

Referenced by `gdcmm::operator<<()`, and `operator=()`.

25.59.6.5 `unsigned int gdcmm::CSAElement::SyngoDTField` [protected]

Referenced by `gdcmm::operator<<()`, `operator=()`, and `operator==()`.

25.59.6.6 `VM gdcmm::CSAElement::ValueMultiplicityField` [protected]

Referenced by `gdcmm::operator<<()`, `operator=()`, and `operator==()`.

25.59.6.7 `VR gdcmm::CSAElement::VRField` [protected]

Referenced by `gdcmm::operator<<()`, `operator=()`, and `operator==()`.

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

- [gdcmmCSAElement.h](#)

25.60 gdcmm::CSAHeader Class Reference

Class for [CSAHeader](#).

```
#include <gdcmmCSAHeader.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](#) & [GetCSAEnd](#) () const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [CSAHeader](#) &d)

25.60.1 Detailed Description

Class for [CSAHeader](#).

SIEMENS store private information in tag (0x0029,0x10,"SIEMENS CSA HEADER") this class is meant for user wishing to access values stored within this private attribute. There are basically two main 'format' for this attribute : SV10/NOMAGIC and DATASET_FORMAT SV10 and NOMAGIC are from a user prospective identical, see CSAHeader.xml for possible name / value stored in this format. DATASET_FORMAT is in fact simply just another DICOM dataset (implicit) with -currently unknown- value. This can be only be printed for now.

Warning

Everything you do with this code is at your own risk, since decoding process was not written from specification documents.
the API of this class might change.

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

See Also

[PDBHeader](#)

External references: 5.1.3.2.4.1 MEDCOM History Information and 5.1.4.3 CSA Non-Image [Module](#) in http://tamsinfo.toshiba.com/docrequest/pdf/E.Soft_v2.0.pdf

Examples:

[csa2img.cxx](#), and [MrProtocol.cxx](#).

25.60.2 Member Enumeration Documentation

25.60.2.1 enum gdcm::CSAHeader::CSAHeaderType

Divers format of [CSAHeader](#) as found 'in the wild'.

Enumerator

UNKNOWN
SV10
NOMAGIC
DATASET_FORMAT
INTERFILE
ZEROED_OUT

25.60.3 Constructor & Destructor Documentation

25.60.3.1 gdcm::CSAHeader::CSAHeader () [inline]

25.60.3.2 gdcm::CSAHeader::~~CSAHeader () [inline]

25.60.4 Member Function Documentation

25.60.4.1 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](#).

25.60.4.2 static const PrivateTag& gdcm::CSAHeader::GetCSDataInfo () [static]

Return the private tag used by SIEMENS to store the CSA Data Info This is: PrivateTag(0x0029,0x0010,"SIEMENS CSA NON-IMAGE");

25.60.4.3 const CSAElement& gdcm::CSAHeader::GetCSAEnd () const [protected]

25.60.4.4 const CSAElement& gdcm::CSAHeader::GetCSAElementByName (const char * name)

Return the CSAElement corresponding to name 'name'

Warning

Case Sensitive

Examples:

[csa2img.cxx](#), and [MrProtocol.cxx](#).

25.60.4.5 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](#).

25.60.4.6 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,"SIEMENS CSA HEADER");

Examples:

[MrProtocol.cxx](#).

25.60.4.7 const DataSet& gdcm::CSAHeader::GetDataSet () const [inline]

Return the DataSet output (use only if Format == DATASET_FORMAT)

25.60.4.8 CSAHeaderType gdcm::CSAHeader::GetFormat () const

return the format of the CSAHeader SV10 and NOMAGIC are equivalent.

25.60.4.9 const char* gdcm::CSAHeader::GetInterfile () const [inline]

Return the string output (use only if Format == Interfile)

25.60.4.10 `bool gdcM::CSAHeader::LoadFromDataElement (DataElement const & de)`

Decode the [CSAHeader](#) from element 'de'.

Examples:

[csa2img.cxx](#), and [MrProtocol.cxx](#).

25.60.4.11 `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<<()`.

25.60.4.12 `template<typename TSwap > std::istream& gdcM::CSAHeader::Read (std::istream & is)`

25.60.4.13 `template<typename TSwap > const std::ostream& gdcM::CSAHeader::Write (std::ostream & os) const`

25.60.5 Friends And Related Function Documentation

25.60.5.1 `std::ostream& operator<< (std::ostream & _os, const CSAHeader & d)` [*friend*]

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

- [gdcMCSAHeader.h](#)

25.61 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)

25.61.1 Detailed Description

Class to represent a map of [CSAHeaderDictEntry](#).

Examples:

[MrProtocol.cxx](#).

25.61.2 Member Typedef Documentation

25.61.2.1 typedef MapCSAHeaderDictEntry::const_iterator gdcm::CSAHeaderDict::ConstIterator

25.61.2.2 typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator

25.61.2.3 typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry

25.61.3 Constructor & Destructor Documentation

25.61.3.1 gdcm::CSAHeaderDict::CSAHeaderDict () [\[inline\]](#)

25.61.4 Member Function Documentation

25.61.4.1 void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry (const CSAHeaderDictEntry & de) [\[inline\]](#)

25.61.4.2 ConstIterator gdcm::CSAHeaderDict::Begin () const [\[inline\]](#)

25.61.4.3 ConstIterator gdcm::CSAHeaderDict::End () const [\[inline\]](#)

25.61.4.4 const CSAHeaderDictEntry& gdcm::CSAHeaderDict::GetCSAHeaderDictEntry (const char * name) const [\[inline\]](#)

Examples:

[MrProtocol.cxx](#).

25.61.4.5 `bool gdcm::CSAHeaderDict::IsEmpty () const` `[inline]`

25.61.4.6 `void gdcm::CSAHeaderDict::LoadDefault ()` `[protected]`

25.61.5 Friends And Related Function Documentation

25.61.5.1 `friend class Dicts` `[friend]`

25.61.5.2 `std::ostream& operator<< (std::ostream &_os, const CSAHeaderDict &_val)` `[friend]`

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

- [gdcmCSAHeaderDict.h](#)

25.62 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)

25.62.1 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

[gdcmm::Dict](#)

Examples:

[MrProtocol.cxx](#).

25.62.2 Constructor & Destructor Documentation

25.62.2.1 `gdcmm::CSAHeaderDictEntry::CSAHeaderDictEntry (const char * name = " ", VR const & vr = VR::INVALID, VM const & vm = VM::VMO, const char * desc = " ") [inline]`

25.62.3 Member Function Documentation

25.62.3.1 `const char* gdcmm::CSAHeaderDictEntry::GetDescription () const [inline]`

Set/Get Description.

25.62.3.2 `const char* gdcmm::CSAHeaderDictEntry::GetName () const [inline]`

Set/Get Name.

Referenced by operator<().

25.62.3.3 `const VM& gdcmm::CSAHeaderDictEntry::GetVM () const [inline]`

Set/Get [VM](#).

25.62.3.4 `const VR& gdcmm::CSAHeaderDictEntry::GetVR () const [inline]`

Set/Get [VR](#).

25.62.3.5 `bool gdcmm::CSAHeaderDictEntry::operator< (const CSAHeaderDictEntry & entry) const [inline]`

References [GetName\(\)](#).

25.62.3.6 `void gdcmm::CSAHeaderDictEntry::SetDescription (const char * desc) [inline]`

25.62.3.7 `void gdcmm::CSAHeaderDictEntry::SetName (const char * name) [inline]`

25.62.3.8 `void gdcmm::CSAHeaderDictEntry::SetVM (VM const & vm) [inline]`

25.62.3.9 `void gdcmm::CSAHeaderDictEntry::SetVR (const VR & vr) [inline]`

25.62.4 Friends And Related Function Documentation

25.62.4.1 `std::ostream& operator<< (std::ostream & _os, const CSAHeaderDictEntry & _val)` [*friend*]

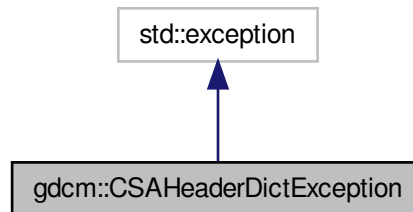
The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDictEntry.h](#)

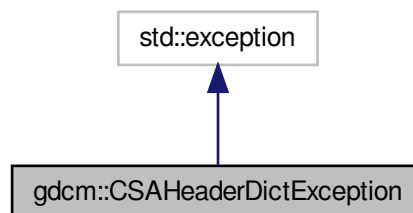
25.63 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for `gdcm::CSAHeaderDictException`:



Collaboration diagram for `gdcm::CSAHeaderDictException`:



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

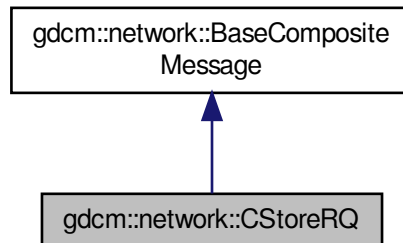
- [gdcmCSAHeaderDict.h](#)

25.64 gdcm::network::CStoreRQ Class Reference

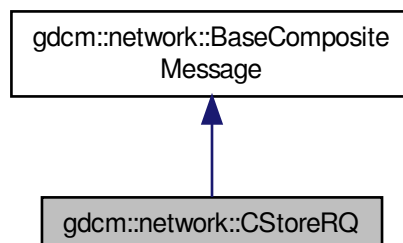
[CStoreRQ](#) this file defines the messages for the cecho action.

```
#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)

25.64.1 Detailed Description

[CStoreRQ](#) this file defines the messages for the cecho action.

25.64.2 Member Function Documentation

25.64.2.1 `std::vector<PresentationDataValue> gdcm::network::CStoreRQ::ConstructPDV (const ULConnection & inConnection, const File & file)`

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

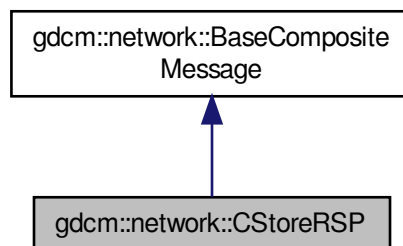
- [gdcmCStoreMessages.h](#)

25.65 gdcm::network::CStoreRSP Class Reference

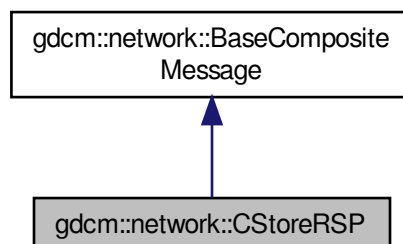
[CStoreRSP](#) this file defines the messages for the cecho action.

```
#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)

25.65.1 Detailed Description

[CStoreRSP](#) this file defines the messages for the cecho action.

25.65.2 Member Function Documentation

25.65.2.1 `std::vector<PresentationDataValue> gdcm::network::CStoreRSP::ConstructPDV (const DataSet * inDataSet, const BasePDU * inPC)`

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

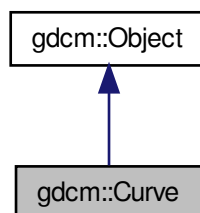
- [gdcmCStoreMessages.h](#)

25.66 gdcm::Curve Class Reference

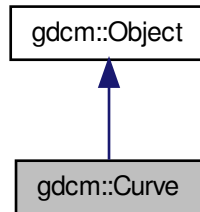
[Curve](#) class to handle element 50xx,3000 [Curve](#) Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

```
#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

25.66.1 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/integriss_HV_5000/xa_integriss.dcm
- TOSHIBA-CurveData[1-3].dcm

25.66.2 Constructor & Destructor Documentation

25.66.2.1 `gdcm::Curve::Curve ()`

25.66.2.2 `gdcm::Curve::~~Curve ()`

25.66.2.3 `gdcm::Curve::Curve (Curve const & ov)`

25.66.3 Member Function Documentation

25.66.3.1 `void gdcm::Curve::Decode (std::istream & is, std::ostream & os)`

25.66.3.2 `void gdcm::Curve::GetAsPoints (float * array) const`

25.66.3.3 `std::vector<unsigned short> const& gdcm::Curve::GetCurveDataDescriptor () const`

25.66.3.4 `unsigned short gdcm::Curve::GetDataValueRepresentation () const`

25.66.3.5 `unsigned short gdcm::Curve::GetDimensions () const`

25.66.3.6 `unsigned short gdcm::Curve::GetGroup () const`

25.66.3.7 `static unsigned int gdcm::Curve::GetNumberOfCurves (DataSet const & ds) [static]`

25.66.3.8 `unsigned short gdcm::Curve::GetNumberOfPoints () const`

25.66.3.9 `const char* gdcm::Curve::GetTypeOfData () const`

25.66.3.10 `const char* gdcm::Curve::GetTypeOfDataDescription () const`

25.66.3.11 `bool gdcm::Curve::IsEmpty () const`

25.66.3.12 `void gdcm::Curve::Print (std::ostream &) const [virtual]`

Reimplemented from [gdcm::Object](#).

25.66.3.13 void gdcM::Curve::SetCoordinateStartValue (unsigned short *v*)

25.66.3.14 void gdcM::Curve::SetCoordinateStepValue (unsigned short *v*)

25.66.3.15 void gdcM::Curve::SetCurve (const char * *array*, unsigned int *length*)

25.66.3.16 void gdcM::Curve::SetCurveDataDescriptor (const uint16_t * *values*, size_t *num*)

25.66.3.17 void gdcM::Curve::SetCurveDescription (const char * *curvedescription*)

25.66.3.18 void gdcM::Curve::SetDataValueRepresentation (unsigned short *datavaluerepresentation*)

25.66.3.19 void gdcM::Curve::SetDimensions (unsigned short *dimensions*)

25.66.3.20 void gdcM::Curve::SetGroup (unsigned short *group*)

25.66.3.21 void gdcM::Curve::SetNumberOfPoints (unsigned short *numberofpoints*)

25.66.3.22 void gdcM::Curve::SetTypeOfData (const char * *typeofdata*)

25.66.3.23 void gdcM::Curve::Update (const DataElement & *de*)

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

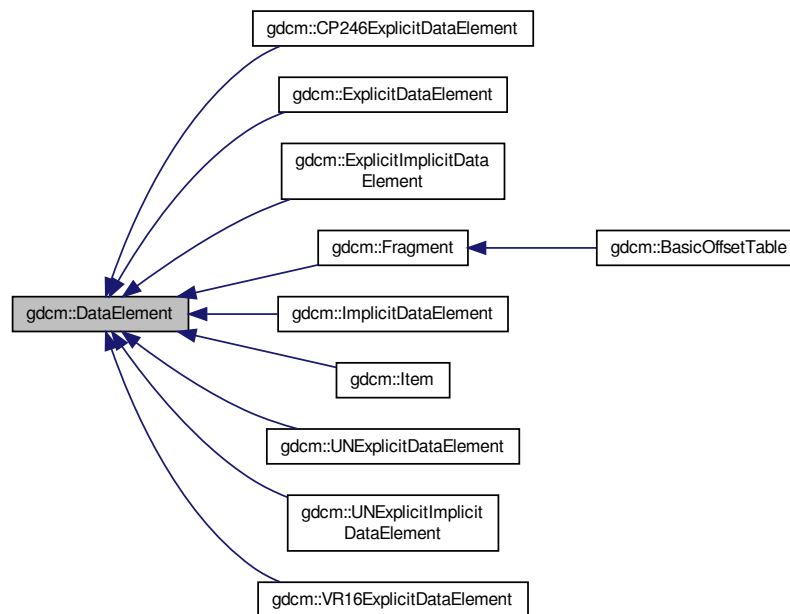
- [gdcMCurve.h](#)

25.67 gdcM::DataElement Class Reference

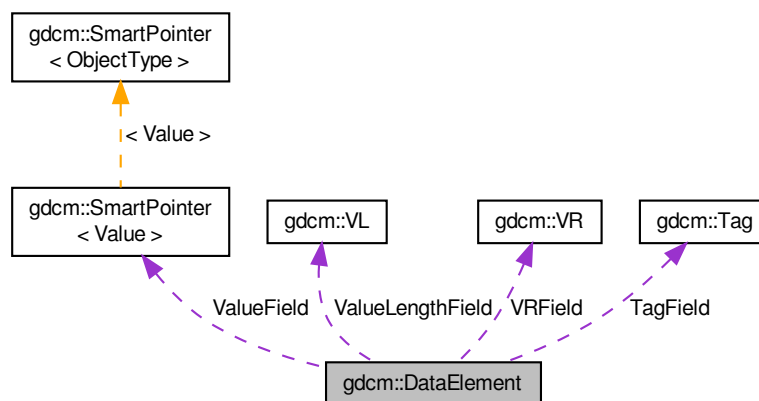
Class to represent a Data [Element](#) either Implicit or Explicit.

```
#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
- const **SequenceOfItems** * **GetSequenceOfItems** () const
- **SequenceOfItems** * **GetSequenceOfItems** ()
- 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 & **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 Attributes

- [Tag](#) [TagField](#)
- [ValuePtr](#) [ValueField](#)
- [VL](#) [ValueLengthField](#)
- [VR](#) [VRField](#)

Friends

- `std::ostream & operator<< (std::ostream &_os, const DataElement &_val)`

25.67.1 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:

[ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.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](#), [NewSequence.cs](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

25.67.2 Member Typedef Documentation

25.67.2.1 `typedef SmartPointer<Value> gdcm::DataElement::ValuePtr` `[protected]`

25.67.3 Constructor & Destructor Documentation

25.67.3.1 `gdcm::DataElement::DataElement (const Tag & t = Tag (0), const VL & vl = 0, const VR & vr = VR::INVALID)`
`[inline]`

25.67.3.2 `gdcm::DataElement::DataElement (const DataElement & _val) [inline]`

25.67.4 Member Function Documentation

25.67.4.1 `void gdcm::DataElement::Clear () [inline]`

Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))

References `gdcm::VR::INVALID`.

Referenced by `gdcm::Item::Clear()`.

25.67.4.2 `void gdcm::DataElement::Empty () [inline]`

Make Data [Element](#) empty (no [Value](#))

25.67.4.3 `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](#), [DumpImageHeaderInfo.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_n >::SetFromDataElement()`, `gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

25.67.4.4 `template<typename TDE> VL gdcm::DataElement::GetLength () const [inline]`

25.67.4.5 `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](#).

25.67.4.6 `const SequenceOfItems* gdcm::DataElement::GetSequenceOfItems () const`

Return the [Value](#) of [DataElement](#) as a Sequence Of Items (if possible)

Warning

: You need to check for NULL return value
 : In some case a [Value](#) could not have been recognized as a [SequenceOfItems](#) in those case the return of the function will be NULL, while the [Value](#) would be a valid [SequenceOfItems](#), in those case prefer GetValueAsSQ. In which case the code internally trigger an assert to warn developer. When in doubt do not use this function and prefer [GetValueAsSQ\(\)](#)

Deprecated Replaced by [DataElement::GetValueAsSQ\(\)](#) as of GDCM 2.2.

25.67.4.7 **SequenceOfItems*** [gdcm::DataElement::GetSequenceOfItems \(\)](#)

25.67.4.8 **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\(\)](#).

25.67.4.9 **Tag&** [gdcm::DataElement::GetTag \(\)](#) `[inline]`

25.67.4.10 **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\(\)](#).

25.67.4.11 **Value&** [gdcm::DataElement::GetValue \(\)](#) `[inline]`

25.67.4.12 **SmartPointer<SequenceOfItems>** [gdcm::DataElement::GetValueAsSQ \(\) const](#)

Interpret the [Value](#) stored in the [DataElement](#). This is more robust (but also more expensive) to call this function rather than the simplest form: [GetSequenceOfItems\(\)](#) It also return NULL when the [Value](#) is NOT of type [SequenceOfItems](#)

Warning

in case [GetSequenceOfItems\(\)](#) succeed the function return this value, otherwise it creates a new [SequenceOfItems](#), you should handle that in your case, for instance: `SmartPointer<SequenceOfItems> sqi = de.GetValueAsSQ();`

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmrptionplan.cxx](#), [gdcmrtpplan.cxx](#), [GetSequenceUltrasound.cxx](#), [LargeVRDSExplicit.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

25.67.4.13 `const VL& gdcm::DataElement::GetVL () const` `[inline]`

Get [VL](#).

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::SequenceOfItems::Read()`, and `gdcm::SequenceOfFragments::ReadValue()`.

25.67.4.14 `VL& gdcm::DataElement::GetVL ()` `[inline]`

25.67.4.15 `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()`.

25.67.4.16 `bool gdcm::DataElement::IsEmpty () const` `[inline]`

Check if Data [Element](#) is empty.

Examples:

[DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [ELSCINT1WaveToText.cxx](#), [FixJAI-BugJPEGLS.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()`.

25.67.4.17 `bool gdcm::DataElement::IsUndefinedLength () const` `[inline]`

return if [Value](#) Length if of undefined length

25.67.4.18 `bool gdcm::DataElement::operator< (const DataElement & de) const` `[inline]`

References `GetTag()`.

25.67.4.19 **DataElement& gdcm::DataElement::operator= (const DataElement & de)** `[inline]`

References TagField, ValueField, ValueLengthField, and VRField.

25.67.4.20 **bool gdcm::DataElement::operator== (const DataElement & de) const** `[inline]`

References TagField, ValueField, ValueLengthField, and VRField.

25.67.4.21 **template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::Read (std::istream & is)**
`[inline]`

25.67.4.22 **template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadOrSkip (std::istream & is, std::set< Tag > const & skiptags)** `[inline]`

25.67.4.23 **template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadPreValue (std::istream & is, std::set< Tag > const & skiptags)** `[inline]`

25.67.4.24 **template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadValue (std::istream & is, std::set< Tag > const & skiptags)** `[inline]`

25.67.4.25 **template<typename TDE , typename TSwap > std::istream& gdcm::DataElement::ReadWithLength (std::istream & is, VL & length)** `[inline]`

25.67.4.26 **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:

[ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), and [StreamImageReaderTest.cxx](#).

Referenced by `gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::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()`, and `gdcm::SequenceOfFragments::ReadPreValue()`.

25.67.4.27 **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](#).

25.67.4.28 void `gdcm::DataElement::SetValue (Value const & v/)` [`inline`]

Warning

you need to set the `ValueLengthField` explicitly

Examples:

[DuplicatePCDE.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

References `gdcm::Value::GetLength()`.

25.67.4.29 void `gdcm::DataElement::SetVL (const VL & v/)` [`inline`]

Set [VL](#) Use with cautious (need to match Part 6), advanced user only

See Also

[SetByteValue](#)

25.67.4.30 void `gdcm::DataElement::SetVLToUndefined ()`

Examples:

[Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

25.67.4.31 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](#), and [StreamImageReader-Test.cxx](#).

References `gdcm::VR::IsVRFile()`.

Referenced by `gdcm::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::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()`.

25.67.4.32 `template<typename TDE , typename TSwap > const std::ostream& gdcm::DataElement::Write (std::ostream & os)
const [inline]`

25.67.5 Friends And Related Function Documentation

25.67.5.1 `std::ostream& operator<< (std::ostream & _os, const DataElement & _val) [friend]`

25.67.6 Member Data Documentation

25.67.6.1 `Tag gdcm::DataElement::TagField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

25.67.6.2 `ValuePtr gdcm::DataElement::ValueField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

25.67.6.3 `VL gdcm::DataElement::ValueLengthField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

25.67.6.4 `VR gdcm::DataElement::VRField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

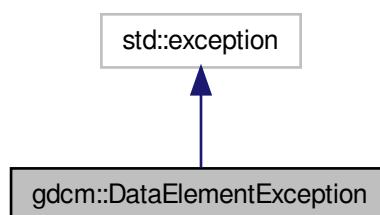
The documentation for this class was generated from the following file:

- [gdcmDataElement.h](#)

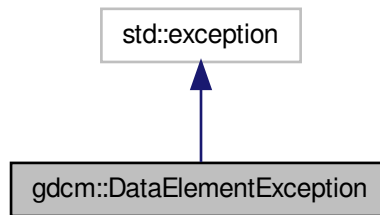
25.68 gdcm::DataElementException Class Reference

```
#include <gdcmDataSet.h>
```

Inheritance diagram for `gdcm::DataElementException`:



Collaboration diagram for `gdcm::DataElementException`:



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

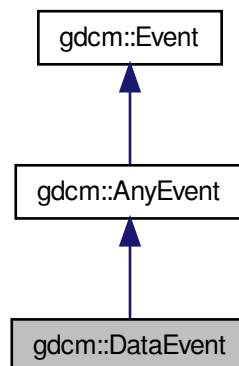
- [gdcmDataSet.h](#)

25.69 `gdcm::DataEvent` Class Reference

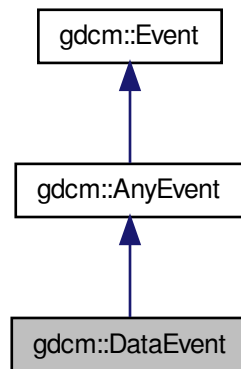
[DataEvent](#).

```
#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`)

25.69.1 Detailed Description

[DataEvent](#).

25.69.2 Member Typedef Documentation

25.69.2.1 typedef `DataEvent` `gdcm::DataEvent::Self`

25.69.2.2 typedef `AnyEvent` `gdcm::DataEvent::Superclass`

25.69.3 Constructor & Destructor Documentation

25.69.3.1 `gdcm::DataEvent::DataEvent (const char * bytes = 0, size_t len = 0)` `[inline]`

25.69.3.2 `virtual gdcm::DataEvent::~DataEvent ()` `[inline],[virtual]`

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

25.69.4 Member Function Documentation

25.69.4.1 `virtual bool gdcm::DataEvent::CheckEvent (const ::gdcm::Event * e) const` `[inline],[virtual]`

25.69.4.2 `const char* gdcm::DataEvent::GetData () const` `[inline]`

25.69.4.3 `size_t gdcm::DataEvent::GetDataLength () const` `[inline]`

25.69.4.4 `virtual const char* gdcm::DataEvent::GetEventName () const` `[inline],[virtual]`

Return the StringName associated with the event.

Implements [gdcm::Event](#).

25.69.4.5 `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](#).

25.69.4.6 `void gdcm::DataEvent::SetData (const char * bytes, size_t len)` `[inline]`

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

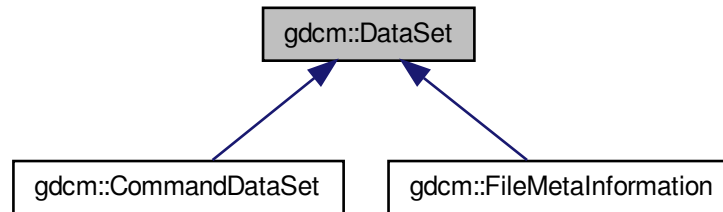
- [gdcmDataEvent.h](#)

25.70 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 & [ReadSelectedTags](#) (std::istream &is, const std::set< [Tag](#) > &tags)
- template<typename TDE , typename TSwap >
std::istream & [ReadSelectedTagsWithLength](#) (std::istream &is, const std::set< [Tag](#) > &tags, [VL](#) &length)
- 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, [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)

25.70.1 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:

[ChangeSequenceUltrasound.cxx](#), [CreateJIPIDataset.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.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](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), and [VolumeSorter.cxx](#).

25.70.2 Member Typedef Documentation

25.70.2.1 `typedef DataSet::const_iterator gdcm::DataSet::ConstIterator`

25.70.2.2 `typedef std::set<DataElement> gdcm::DataSet::DataElementSet`

25.70.2.3 `typedef DataSet::iterator gdcm::DataSet::Iterator`

25.70.2.4 `typedef DataSet::size_type gdcm::DataSet::SizeType`

25.70.3 Member Function Documentation

25.70.3.1 `ConstIterator gdcm::DataSet::Begin () const [inline]`

Examples:

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [DuplicatePCDE.cxx](#).

25.70.3.2 `Iterator gdcm::DataSet::Begin () [inline]`

25.70.3.3 `void gdcm::DataSet::Clear () [inline]`

Referenced by `gdcm::Item::Read()`.

25.70.3.4 `Tag gdcm::DataSet::ComputeDataElement (const PrivateTag & t) const [protected]`

25.70.3.5 `template<typename TDE > unsigned int gdcm::DataSet::ComputeGroupLength (Tag const & tag) const` `[inline]`

References `gdcm::Tag::GetElement()`, and `gdcm::Tag::GetGroup()`.

25.70.3.6 `ConstIterator gdcm::DataSet::End () const` `[inline]`

Examples:

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [DuplicatePCDE.cxx](#).

25.70.3.7 `Iterator gdcm::DataSet::End ()` `[inline]`

25.70.3.8 `bool gdcm::DataSet::FindDataElement (const PrivateTag & t) const`

Look up if private tag 't' is present in the dataset:

Examples:

[ChangeSequenceUltrasound.cxx](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImage-HeaderInfo.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()`.

25.70.3.9 `bool gdcm::DataSet::FindDataElement (const Tag & t) const` `[inline]`

25.70.3.10 `const DataElement& gdcm::DataSet::FindNextDataElement (const Tag & t) const` `[inline]`

Examples:

[DuplicatePCDE.cxx](#).

25.70.3.11 `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:

[ChangeSequenceUltrasound.cxx](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImage-HeaderInfo.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, TVR, TVM >::SetFromDataSet()`, `gdcm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet()`.

25.70.3.12 `const DataElement& gdcm::DataSet::GetDataElement (const PrivateTag & t) const`

Return the dataelement.

25.70.3.13 `const DataElement& gdcm::DataSet::GetDEEnd () const` `[protected]`

25.70.3.14 `const DataElementSet& gdcm::DataSet::GetDES () const` `[inline]`

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

25.70.3.15 `DataElementSet& gdcm::DataSet::GetDES ()` `[inline]`

25.70.3.16 `template<typename TDE > VL gdcm::DataSet::GetLength () const` `[inline]`

25.70.3.17 `MediaStorage gdcm::DataSet::GetMediaStorage () const`

25.70.3.18 `std::string gdcm::DataSet::GetPrivateCreator (const Tag & t) const`

Return the private creator of the private tag 't':

Examples:

[DuplicatePCDE.cxx](#).

25.70.3.19 `void gdcm::DataSet::Insert (const DataElement & de)` `[inline]`

Insert a [DataElement](#) in the [DataSet](#).

Warning

: [Tag](#) need to be $\geq 0x8$ 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](#), and [StreamImageReader-Test.cxx](#).

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

25.70.3.20 `void gdcm::DataSet::InsertDataElement (const DataElement & de)` `[inline]`, `[protected]`

References `gdcmWarningMacro`, `gdcm::Value::GetLength()`, `gdcm::DataElement::GetValue()`, `gdcm::DataElement::GetVL()`, and `gdcm::DataElement::IsEmpty()`.

25.70.3.21 `bool gdcmm::DataSet::IsEmpty () const [inline]`

Returns if the dataset is empty.

Referenced by `gdcmm::Item::Read()`.

25.70.3.22 `const DataElement& gdcmm::DataSet::operator() (uint16_t group, uint16_t element) const [inline]`

25.70.3.23 `DataSet& gdcmm::DataSet::operator= (DataSet const & val) [inline]`

25.70.3.24 `const DataElement& gdcmm::DataSet::operator[] (const Tag & t) const [inline]`

25.70.3.25 `void gdcmm::DataSet::Print (std::ostream & os, std::string const & indent = " ") const [inline]`

Referenced by `gdcmm::operator<<()`.

25.70.3.26 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataSet::Read (std::istream & is)`

25.70.3.27 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataSet::ReadNested (std::istream & is)`

25.70.3.28 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataSet::ReadSelectedTags (std::istream & is, const std::set< Tag > & tags)`

25.70.3.29 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataSet::ReadSelectedTagsWithLength (std::istream & is, const std::set< Tag > & tags, VL & length)`

25.70.3.30 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataSet::ReadUpToTag (std::istream & is, const Tag & t, std::set< Tag > const & skiptags)`

25.70.3.31 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataSet::ReadUpToTagWithLength (std::istream & is, const Tag & t, VL & length)`

25.70.3.32 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataSet::ReadWithLength (std::istream & is, VL & length)`

25.70.3.33 `SizeType gdcmm::DataSet::Remove (const Tag & tag) [inline]`

Completely remove a dataelement from the dataset.

Examples:

[GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

25.70.3.34 `void gdcmm::DataSet::Replace (const DataElement & de) [inline]`

Replace a dataelement with another one.

Examples:

[ChangeSequenceUltrasound.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [Hello-World.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

25.70.3.35 void gdcm::DataSet::ReplaceEmpty (const DataElement & de) [inline]

Only replace a DICOM attribute when it is missing or empty.

25.70.3.36 SizeType gdcm::DataSet::Size () const [inline]

Examples:

[DumpGEMSMovieGroup.cxx](#).

Referenced by gdcm::SequenceOfItems::Read().

25.70.3.37 template<typename TDE , typename TSwap > std::ostream const& gdcm::DataSet::Write (std::ostream & os) const

25.70.4 Friends And Related Function Documentation

25.70.4.1 friend class CSAHeader [friend]

25.70.4.2 std::ostream& operator<< (std::ostream & _os, const DataSet & val) [friend]

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

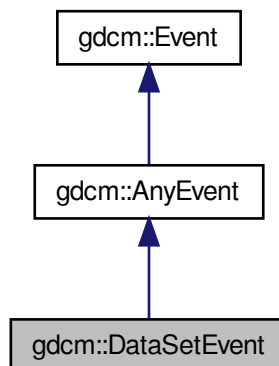
- [gdcmDataSet.h](#)

25.71 gdcm::DataSetEvent Class Reference

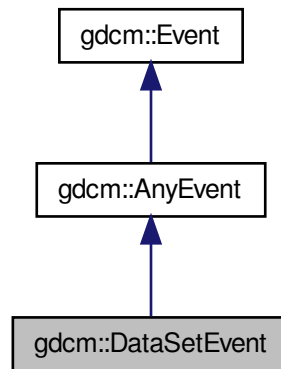
[DataSetEvent](#) Special type of event triggered during the [DataSet](#) store/move process.

```
#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

25.71.1 Detailed Description

[DataSetEvent](#) Special type of event triggered during the [DataSet](#) store/move process.

See Also

25.71.2 Member Typedef Documentation

25.71.2.1 typedef `DataSetEvent` `gdcm::DataSetEvent::Self`

25.71.2.2 typedef `AnyEvent` `gdcm::DataSetEvent::Superclass`

25.71.3 Constructor & Destructor Documentation

25.71.3.1 `gdcm::DataSetEvent::DataSetEvent (DataSet const * ds = NULL)` `[inline]`

25.71.3.2 `virtual gdcm::DataSetEvent::~~DataSetEvent ()` `[inline]`, `[virtual]`

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

25.71.4 Member Function Documentation

25.71.4.1 `virtual bool gdcm::DataSetEvent::CheckEvent (const ::gdcm::Event * e) const` `[inline]`, `[virtual]`

25.71.4.2 `DataSet const& gdcm::DataSetEvent::GetDataSet () const` `[inline]`

25.71.4.3 `virtual const char* gdcm::DataSetEvent::GetEventName () const` `[inline]`, `[virtual]`

Return the StringName associated with the event.

Implements [gdcm::Event](#).

25.71.4.4 `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](#)

25.72 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)

25.72.1 Detailed Description

[DataSetHelper](#) (internal class, not intended for user level)

25.72.2 Member Function Documentation

25.72.2.1 `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:

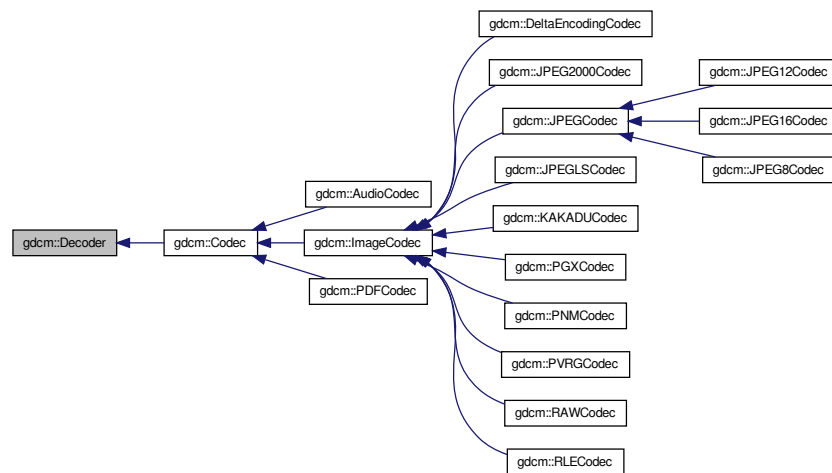
- [gdcmDataSetHelper.h](#)

25.73 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 &)

25.73.1 Detailed Description

[Decoder.](#)

25.73.2 Constructor & Destructor Documentation

25.73.2.1 virtual gdcm::Decoder::~Decoder () [inline], [virtual]

25.73.3 Member Function Documentation

25.73.3.1 `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](#).

25.73.3.2 `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](#).

25.73.3.3 `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](#)

25.74 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](#) ()

25.74.1 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.

25.74.2 Constructor & Destructor Documentation

25.74.2.1 `gdcm::DefinedTerms::DefinedTerms ()` `[inline]`

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

- [gdcmDefinedTerms.h](#)

25.75 `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](#)

25.75.1 Detailed Description

FIXME I do not like the name '[Defs](#)'.

Note

bla

Examples:

[GenerateStandardSOPClasses.cxx](#), and [TraverseModules.cxx](#).

25.75.2 Constructor & Destructor Documentation

25.75.2.1 `gdcm::Defs::Defs ()`

25.75.2.2 `gdcm::Defs::~~Defs ()`

25.75.3 Member Function Documentation

25.75.3.1 `const IOD& gdcm::Defs::GetIODFromFile (const File & file) const`

25.75.3.2 `static const char* gdcm::Defs::GetIODNameFromMediaStorage (MediaStorage const & ms) [static]`

Examples:

[GenerateStandardSOPClasses.cxx](#).

25.75.3.3 `const IODs& gdcm::Defs::GetIODs () const [inline]`

25.75.3.4 `IODs& gdcm::Defs::GetIODs () [inline]`

25.75.3.5 `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 conveniently wrapped within [Modules](#). See [gdcm::Module](#) API directly

25.75.3.6 `Macros& gdcm::Defs::GetMacros () [inline]`

25.75.3.7 `const Modules& gdcm::Defs::GetModules () const [inline]`

25.75.3.8 `Modules& gdcm::Defs::GetModules () [inline]`

25.75.3.9 `Type gdcm::Defs::GetTypeFromTag (const File & file, const Tag & tag) const`

25.75.3.10 `bool gdcm::Defs::IsEmpty () const [inline]`

25.75.3.11 `void gdcm::Defs::LoadDefaults () [protected]`

25.75.3.12 `void gdcm::Defs::LoadFromFile (const char * filename) [protected]`

25.75.3.13 `bool gdcM::Defs::Verify (const File & file) const`

25.75.3.14 `bool gdcM::Defs::Verify (const DataSet & ds) const`

25.75.4 Friends And Related Function Documentation

25.75.4.1 `friend class Global` [*friend*]

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

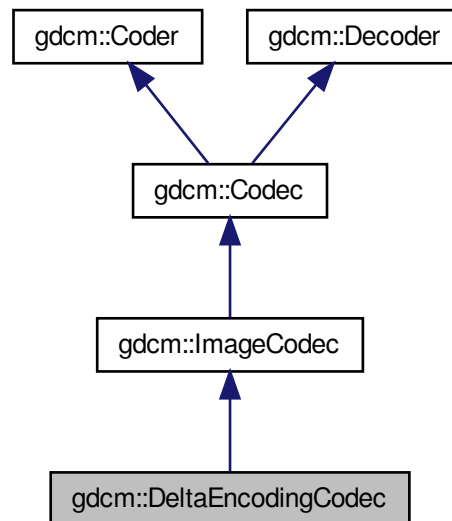
- [gdcMDefs.h](#)

25.76 gdcM::DeltaEncodingCodec Class Reference

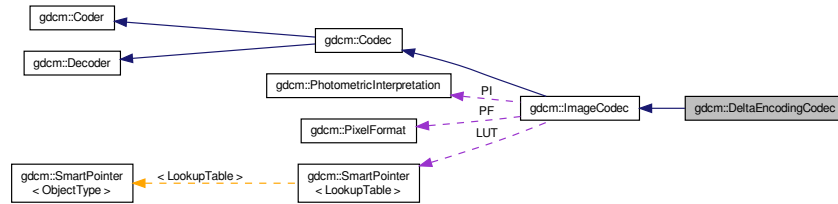
[DeltaEncodingCodec](#) compression used by some private vendor.

```
#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

25.76.1 Detailed Description

[DeltaEncodingCodec](#) compression used by some private vendor.

25.76.2 Constructor & Destructor Documentation

25.76.2.1 [gdcm::DeltaEncodingCodec::DeltaEncodingCodec](#) ()

25.76.2.2 [gdcm::DeltaEncodingCodec::~~DeltaEncodingCodec](#) ()

25.76.3 Member Function Documentation

25.76.3.1 bool [gdcm::DeltaEncodingCodec::CanDecode](#) ([TransferSyntax](#) const & ts)

25.76.3.2 bool [gdcm::DeltaEncodingCodec::Decode](#) ([DataElement](#) const & , [DataElement](#) &) [virtual]

Decode.

Reimplemented from [gdcm::Decoder](#).

25.76.3.3 `bool gdcm::DeltaEncodingCodec::Decode (std::istream & is, std::ostream & os)` `[protected]`

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

- [gdcmDeltaEncodingCodec.h](#)

25.77 gdcm::DICOMDIR Class Reference

[DICOMDIR](#) class.

```
#include <gdcmDICOMDIR.h>
```

Public Member Functions

- [DICOMDIR](#) ()
- [DICOMDIR](#) (const [FileSet](#) &fs)

25.77.1 Detailed Description

[DICOMDIR](#) class.

Structured for handling [DICOMDIR](#)

25.77.2 Constructor & Destructor Documentation

25.77.2.1 `gdcm::DICOMDIR::DICOMDIR ()` `[inline]`

25.77.2.2 `gdcm::DICOMDIR::DICOMDIR (const FileSet & fs)` `[inline]`

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

- [gdcmDICOMDIR.h](#)

25.78 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](#) ()

25.78.1 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 [gdcm::Scanner](#) does not allow us See PS 3.11 / [Table](#) D.3-2 STD-GEN Additional [DICOMDIR](#) Keys

Examples:

[GenerateDICOMDIR.cs](#).

25.78.2 Member Typedef Documentation

25.78.2.1 `typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType`

25.78.2.2 `typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType`

25.78.3 Constructor & Destructor Documentation

25.78.3.1 `gdcm::DICOMDIRGenerator::DICOMDIRGenerator ()`

25.78.3.2 `gdcm::DICOMDIRGenerator::~~DICOMDIRGenerator ()`

25.78.4 Member Function Documentation

25.78.4.1 `bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ()` [protected]

25.78.4.2 `bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ()` [protected]

25.78.4.3 `bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ()` [protected]

25.78.4.4 `bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ()` [protected]

25.78.4.5 `bool gdcm::DICOMDIRGenerator::Generate ()`

Main function to generate the [DICOMDIR](#).

25.78.4.6 `File& gdcm::DICOMDIRGenerator::GetFile ()`

25.78.4.7 `Scanner& gdcm::DICOMDIRGenerator::GetScanner ()` [protected]

25.78.4.8 `void gdcm::DICOMDIRGenerator::SetDescriptor (const char * d)`

Set the [File](#) Set ID.

Warning

 this need to be a valid [VR::CS](#) value

25.78.4.9 `void gdcm::DICOMDIRGenerator::SetFile (const File & f)`

Set/Get file. The [DICOMDIR](#) file will be valid once a call to Generate has been done.

25.78.4.10 `void gdcm::DICOMDIRGenerator::SetFilenames (FilenameType const & fns)`

Set the list of filenames from which the [DICOMDIR](#) should be generated from.

25.78.4.11 `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](#)

25.79 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)

25.79.1 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](#).

25.79.2 Member Typedef Documentation

25.79.2.1 `typedef MapDictEntry::const_iterator gdcm::Dict::ConstIterator`

25.79.2.2 `typedef MapDictEntry::iterator gdcm::Dict::Iterator`

25.79.2.3 `typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry`

25.79.3 Constructor & Destructor Documentation

25.79.3.1 `gdcm::Dict::Dict () [inline]`

25.79.4 Member Function Documentation

25.79.4.1 `void gdcm::Dict::AddDictEntry (const Tag & tag, const DictEntry & de) [inline]`

25.79.4.2 `ConstIterator gdcm::Dict::Begin () const [inline]`

Examples:

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

25.79.4.3 `ConstIterator gdcm::Dict::End () const [inline]`

Examples:

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

25.79.4.4 `const DictEntry& gdcm::Dict::GetDictEntry (const Tag & tag) const [inline]`

Examples:

[GenFakeIdentifyFile.cxx](#), and [PublicDict.cxx](#).

25.79.4.5 `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](#).

25.79.4.6 `const DictEntry& gdcm::Dict::GetDictEntryByName (const char * name, Tag & tag) const [inline]`

Inefficient way of looking up tag by name. Technically DICOM does not guarantee uniqueness (and [Curve](#) / [Overlay](#) are there to prove it). But most of the time name is in fact uniq and can be uniquely link to a tag

Examples:

[ReadAndPrintAttributes.cxx](#).

25.79.4.7 `const char* gdcmm::Dict::GetKeywordFromTag (Tag const & tag) const` `[inline]`

Function to return the Keyword from a [Tag](#).

25.79.4.8 `bool gdcmm::Dict::IsEmpty () const` `[inline]`

Referenced by `gdcmm::Dicts::IsEmpty()`.

25.79.4.9 `void gdcmm::Dict::LoadDefault ()` `[protected]`

25.79.5 Friends And Related Function Documentation

25.79.5.1 `friend class Dicts` `[friend]`

25.79.5.2 `std::ostream& operator<< (std::ostream & _os, const Dict & _val)` `[friend]`

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

- [gdcmmDict.h](#)

25.80 gdcmm::DictConverter Class Reference

Class to convert a .dic file into something else:

```
#include <gdcmmDictConverter.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](#) ()

25.80.1 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

25.80.2 Member Enumeration Documentation

25.80.2.1 enum `gdcmm::DictConverter::OutputTypes`

Enumerator

DICT_DEFAULT
DICT_DEBUG
DICT_XML

25.80.3 Constructor & Destructor Documentation

25.80.3.1 `gdcmm::DictConverter::DictConverter ()`

25.80.3.2 `gdcmm::DictConverter::~~DictConverter ()`

25.80.4 Member Function Documentation

25.80.4.1 `void gdcmm::DictConverter::AddGroupLength ()` [protected]

25.80.4.2 `void gdcmm::DictConverter::Convert ()`

25.80.4.3 `bool gdcmm::DictConverter::ConvertToCXX (const char * raw, std::string & cxx)` [protected]

- 25.80.4.4 `bool gdcmm::DictConverter::ConvertToXML (const char * raw, std::string & cxx)` [protected]
- 25.80.4.5 `const std::string& gdcmm::DictConverter::GetDictName ()` const
- 25.80.4.6 `const std::string& gdcmm::DictConverter::GetInputFilename ()` const
- 25.80.4.7 `const std::string& gdcmm::DictConverter::GetOutputFilename ()` const
- 25.80.4.8 `int gdcmm::DictConverter::GetOutputType ()` const [inline]
- 25.80.4.9 `static bool gdcmm::DictConverter::Readuint16 (const char * raw, uint16_t & ov)` [static]
- 25.80.4.10 `static bool gdcmm::DictConverter::ReadVM (const char * raw, VM::VMType & type)` [static]
- 25.80.4.11 `static bool gdcmm::DictConverter::ReadVR (const char * raw, VR::VRType & type)` [static]
- 25.80.4.12 `void gdcmm::DictConverter::SetDictName (const char * name)`
- 25.80.4.13 `void gdcmm::DictConverter::SetInputFileName (const char * filename)`
- 25.80.4.14 `void gdcmm::DictConverter::SetOutputFileName (const char * filename)`
- 25.80.4.15 `void gdcmm::DictConverter::SetOutputType (int type)` [inline]
- 25.80.4.16 `void gdcmm::DictConverter::WriteFooter ()` [protected]
- 25.80.4.17 `void gdcmm::DictConverter::WriteHeader ()` [protected]

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

- [gdcmmDictConverter.h](#)

25.81 gdcmm::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 [gdcmm::Tag](#) to the needed information.

```
#include <gdcmmDictEntry.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

- std::ostream & [operator<<](#) (std::ostream &_os, const [DictEntry](#) &_val)

25.81.1 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 Private-DictEntry...

See Also

[gdcm::Dict](#)

Examples:

[GenAllVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [PublicDict.cxx](#), and [TraverseModules.cxx](#).

25.81.2 Constructor & Destructor Documentation

- 25.81.2.1 `gdcm::DictEntry::DictEntry (const char * name = " ", const char * keyword = " ", VR const & vr = VR::INVALID, VM const & vm = VM::VMO, bool ret = false) [inline]`

25.81.3 Member Function Documentation

- 25.81.3.1 `const char* gdcm::DictEntry::GetKeyword () const [inline]`

same as GetName but without spaces...

- 25.81.3.2 `const char* gdcm::DictEntry::GetName () const [inline]`

Set/Get Name.

Referenced by `gdcm::PrivateDict::PrintXML()`.

25.81.3.3 `bool gdcmm::DictEntry::GetRetired () const [inline]`

Set/Get Retired flag.

Examples:

[GenAllVR.cxx](#).

25.81.3.4 `const VM& gdcmm::DictEntry::GetVM () const [inline]`

Set/Get [VM](#).

Referenced by `gdcmm::PrivateDict::AddDictEntry()`, and `gdcmm::PrivateDict::PrintXML()`.

25.81.3.5 `const VR& gdcmm::DictEntry::GetVR () const [inline]`

Set/Get [VR](#).

Examples:

[GenAllVR.cxx](#), and [GenFakeIdentifyFile.cxx](#).

Referenced by `gdcmm::PrivateDict::AddDictEntry()`, and `gdcmm::PrivateDict::PrintXML()`.

25.81.3.6 `bool gdcmm::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 explicitly 'XX' ones)

25.81.3.7 `void gdcmm::DictEntry::SetElementXX (bool v) [inline]`

Set whether element is shared in multiple elements (Source [Image](#) IDs typically)

25.81.3.8 `void gdcmm::DictEntry::SetGroupXX (bool v) [inline]`

Set whether element is shared in multiple groups (Curve/Overlay typically)

25.81.3.9 `void gdcmm::DictEntry::SetKeyword (const char * keyword) [inline]`

25.81.3.10 `void gdcmm::DictEntry::SetName (const char * name) [inline]`

25.81.3.11 `void gdcmm::DictEntry::SetRetired (bool retired) [inline]`

25.81.3.12 `void gdcmm::DictEntry::SetVM (VM const & vm) [inline]`

25.81.3.13 `void gdcmm::DictEntry::SetVR (const VR & vr) [inline]`

Referenced by `gdcmm::PrivateDict::AddDictEntry()`.

25.81.4 Friends And Related Function Documentation

25.81.4.1 `std::ostream& operator<< (std::ostream & _os, const DictEntry & _val)` `[friend]`

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

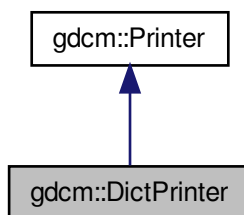
- [gdcMDictEntry.h](#)

25.82 gdcM::DictPrinter Class Reference

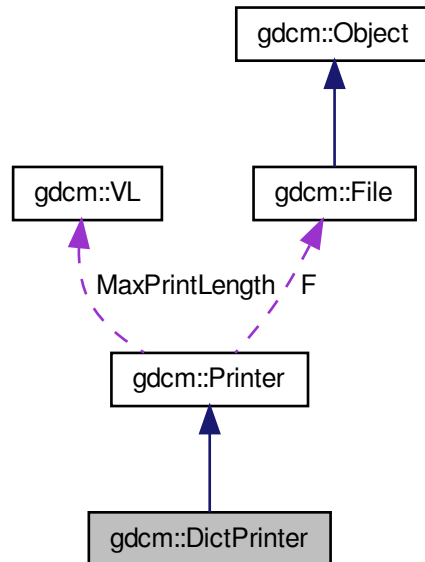
[DictPrinter](#) class.

```
#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

25.82.1 Detailed Description

[DictPrinter](#) class.

25.82.2 Constructor & Destructor Documentation

25.82.2.1 gdcM::DictPrinter::DictPrinter ()

25.82.2.2 `gdcm::DictPrinter::~~DictPrinter ()`

25.82.3 Member Function Documentation

25.82.3.1 `void gdcm::DictPrinter::Print (std::ostream & os)`

25.82.3.2 `void gdcm::DictPrinter::PrintDataElement2 (std::ostream & os, const DataSet & ds, const DataElement & ide)`
[protected]

25.82.3.3 `void gdcm::DictPrinter::PrintDataSet2 (std::ostream & os, const DataSet & ds)` [protected]

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

- [gdcmDictPrinter.h](#)

25.83 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
- 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)`

25.83.1 Detailed Description

Class to manipulate the sum of knowledge (all the dict user load)

Note

bla

Examples:

[GenAllVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

25.83.2 Member Enumeration Documentation

25.83.2.1 `enum gdcm::Dicts::ConstructorType` `[protected]`

Enumerator

PHILIPS

GEMS

SIEMENS

25.83.3 Constructor & Destructor Documentation

25.83.3.1 `gdcm::Dicts::Dicts ()`

25.83.3.2 `gdcm::Dicts::~~Dicts ()`

25.83.4 Member Function Documentation

25.83.4.1 `static const char* gdcm::Dicts::GetConstructorString (ConstructorType type)` `[static], [protected]`

25.83.4.2 `const CSAHeaderDict& gdcm::Dicts::GetCSAHeaderDict () const`

Examples:

[MrProtocol.cxx](#).

25.83.4.3 `const DictEntry& gdcm::Dicts::GetDictEntry (const Tag & tag, const char * owner = NULL) const`

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](#).

25.83.4.4 `const DictEntry& gdcM::Dicts::GetDictEntry (const PrivateTag & tag) const`

25.83.4.5 `const PrivateDict& gdcM::Dicts::GetPrivateDict () const`

25.83.4.6 `PrivateDict& gdcM::Dicts::GetPrivateDict ()`

25.83.4.7 `const Dict& gdcM::Dicts::GetPublicDict () const`

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

25.83.4.8 `bool gdcM::Dicts::IsEmpty () const [inline]`

References `gdcM::Dict::IsEmpty()`.

25.83.4.9 `void gdcM::Dicts::LoadDefaults () [protected]`

25.83.5 Friends And Related Function Documentation

25.83.5.1 `friend class Global [friend]`

25.83.5.2 `std::ostream& operator<< (std::ostream & _os, const Dicts & d) [friend]`

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

- [gdcMDicts.h](#)

25.84 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 }

25.84.1 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)

25.84.2 Member Enumeration Documentation

25.84.2.1 enum gdcmm::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](#)

25.85 gdcmd::DirectionCosines Class Reference

class to handle [DirectionCosines](#)

```
#include <gdcmdDirectionCosines.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)

25.85.1 Detailed Description

class to handle [DirectionCosines](#)

Examples:

[DiscriminateVolume.cxx](#).

25.85.2 Constructor & Destructor Documentation

25.85.2.1 `gdcm::DirectionCosines::DirectionCosines ()`

25.85.2.2 `gdcm::DirectionCosines::DirectionCosines (const double dircos[6])`

25.85.2.3 `gdcm::DirectionCosines::~~DirectionCosines ()`

25.85.3 Member Function Documentation

25.85.3.1 `double gdcm::DirectionCosines::ComputeDistAlongNormal (const double ipp[3]) const`

Compute the distance along the normal.

25.85.3.2 `void gdcm::DirectionCosines::Cross (double z[3]) const`

Compute Cross product.

25.85.3.3 `double gdcm::DirectionCosines::CrossDot (DirectionCosines const & dc) const`

Compute the Dot product of the two cross vector of both [DirectionCosines](#) object.

Examples:

[DiscriminateVolume.cxx](#).

25.85.3.4 `double gdcm::DirectionCosines::Dot () const`

Compute Dot.

25.85.3.5 `bool gdcm::DirectionCosines::IsValid () const`

Return whether or not this is a valid direction cosines.

25.85.3.6 `void gdcm::DirectionCosines::Normalize ()`

Normalize in-place.

25.85.3.7 `gdcm::DirectionCosines::operator const double * () const` `[inline]`

Make the class behave like a const double *.

25.85.3.8 void `gdcm::DirectionCosines::Print (std::ostream &)` const

Print.

25.85.3.9 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](#)

25.86 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)`

25.86.1 Detailed Description

Class for manipulation directories.

Note

This implementation provide a cross platform implementation for manipulating diretores: basically traversing directories and harvesting files
will not take into account unix type hidden file recursive option will not look into UNIX type hidden directory (those starting with a '.')
Since python or C# provide there own equivalent implementation, in which case [gdcm::Directory](#) does not make much sense.

Examples:

[DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcm.cxx](#), and [VolumeSorter.cxx](#).

25.86.2 Member Typedef Documentation

25.86.2.1 `typedef std::vector<FilenameType> gdcm::Directory::FilenameType`

Examples:

[DiscriminateVolume.cxx](#).

25.86.2.2 `typedef std::string gdcm::Directory::FilenameType`

25.86.3 Constructor & Destructor Documentation

25.86.3.1 `gdcm::Directory::Directory () \[inline\]`

25.86.3.2 `gdcm::Directory::~~Directory () \[inline\]`

25.86.4 Member Function Documentation

25.86.4.1 `unsigned int gdcm::Directory::Explore (FilenameType const & name, bool recursive) \[protected\]`

Return number of file found when 'recursive'ly exploring directory *name*

25.86.4.2 `FilenameType const& gdcm::Directory::GetDirectories () const \[inline\]`

Return the Directories traversed.

25.86.4.3 `FilenameType` `const& gdcmm::Directory::GetFilenames () const` `[inline]`

Set/Get the file names within the directory.

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8Qt-Dir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcmm.cxx](#), and [VolumeSorter.cxx](#).

25.86.4.4 `FilenameType` `const& gdcmm::Directory::GetToplevel () const` `[inline]`

Get the name of the toplevel directory.

25.86.4.5 `unsigned int gdcmm::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:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8Qt-Dir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcmm.cxx](#), and [VolumeSorter.cxx](#).

25.86.4.6 `void gdcmm::Directory::Print (std::ostream & os = std::cout) const`

Print.

Examples:

[SortImage.cxx](#).

Referenced by `gdcmm::operator<<()`.

25.86.5 Friends And Related Function Documentation

25.86.5.1 `std::ostream& operator<< (std::ostream & _os, const Directory & d)` `[friend]`

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

- [gdcmmDirectory.h](#)

25.87 `gdcmm::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::FilenameType GetCTImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType GetFilenamesFromSeriesUIDs](#) (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string [GetFrameOfReference](#) (const std::vector< [DataSet](#) > &inDS)
- static [Directory::FilenameType GetMRImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType GetRTStructSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenameType GetSeriesUIDsBySOPClassUID](#) (const std::string &inDirectory, const std::string &inSOPClassUID)
- static std::string [GetSOPClassUID](#) (const std::vector< [DataSet](#) > &inDS)
- static std::string [GetStringValueFromTag](#) (const [gdcm::Tag](#) &t, const [gdcm::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)

25.87.1 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.

25.87.2 Member Function Documentation

25.87.2.1 static [Directory::FilenameType](#) [gdcm::DirectoryHelper::GetCTImageSeriesUIDs](#) (const std::string & *inDirectory*)
[static]

25.87.2.2 static [Directory::FilenameType](#) [gdcm::DirectoryHelper::GetFilenamesFromSeriesUIDs](#) (const std::string & *inDirectory*, const std::string & *inSeriesUID*) [static]

Examples:

[GenerateRTSTRUCT.cxx](#).

25.87.2.3 static std::string [gdcm::DirectoryHelper::GetFrameOfReference](#) (const std::vector< [DataSet](#) > & *inDS*) [static]

25.87.2.4 static [Directory::FilenameType](#) [gdcm::DirectoryHelper::GetMRImageSeriesUIDs](#) (const std::string & *inDirectory*)
[static]

25.87.2.5 **static** **Directory::FilenameType** **gdcm::DirectoryHelper::GetRTStructSeriesUIDs** (**const** **std::string** & *inDirectory*)
[static]

Examples:

[GenerateRTSTRUCT.cxx](#).

25.87.2.6 **static** **Directory::FilenameType** **gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID** (**const** **std::string** & *inDirectory*, **const** **std::string** & *inSOPClassUID*) [static]

25.87.2.7 **static** **std::string** **gdcm::DirectoryHelper::GetSOPClassUID** (**const** **std::vector**< **DataSet** > & *inDS*) [static]

25.87.2.8 **static** **std::string** **gdcm::DirectoryHelper::GetStringValueFromTag** (**const** **gdcm::Tag** & *t*, **const** **gdcm::DataSet** & *ds*)
[static]

25.87.2.9 **static** **std::vector**<**DataSet**> **gdcm::DirectoryHelper::LoadImageFromFiles** (**const** **std::string** & *inDirectory*, **const** **std::string** & *inSeriesUID*) [static]

25.87.2.10 **static** **std::string** **gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex** (**int** *inIndex*, **const** **std::vector**< **DataSet** > & *inDS*) [static]

25.87.2.11 **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](#)

25.88 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcmDummyValueGenerator.h>
```

Static Public Member Functions

- **static** **const char** * [Generate](#) (**const** **char** **input*)

25.88.1 Detailed Description

Class for generating dummy value.

See Also

[Anonymizer](#)

25.88.2 Member Function Documentation

25.88.2.1 `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:

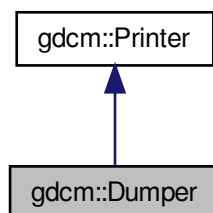
- [gdcmDummyValueGenerator.h](#)

25.89 gdcm::Dumper Class Reference

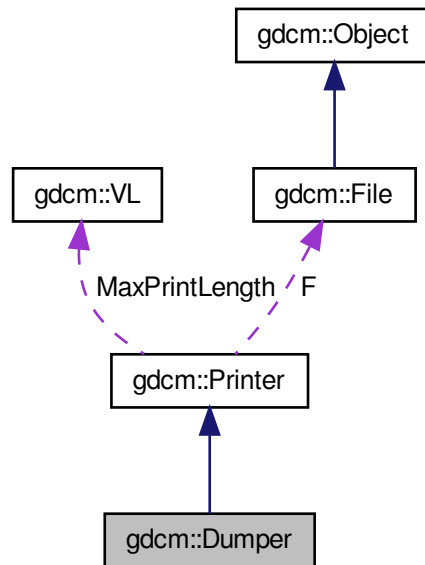
[Codec](#) class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for `gdcm::Dumper`:



Collaboration diagram for `gdc::Dumper`:



Public Member Functions

- [Dumper](#) ()
- [~Dumper](#) ()

Additional Inherited Members

25.89.1 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

25.89.2 Constructor & Destructor Documentation

25.89.2.1 `gdc::Dumper::Dumper ()` `[inline]`

25.89.2.2 `gdc::Dumper::~~Dumper ()` `[inline]`

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

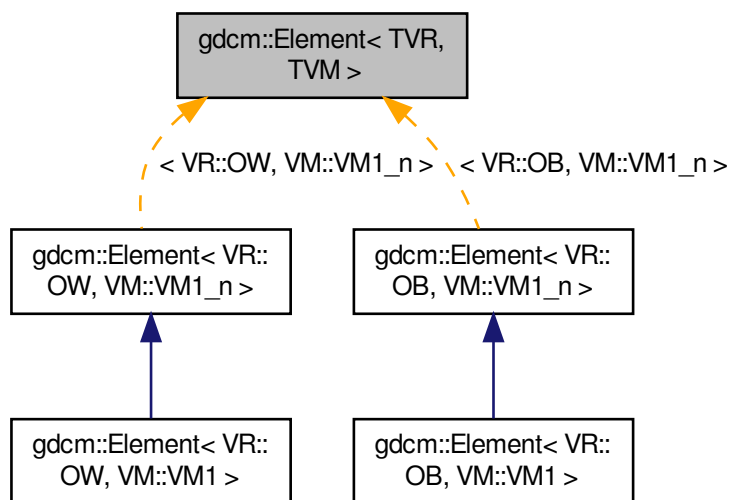
- [gdcDumper.h](#)

25.90 gdcmm::Element< TVR, TVM > Class Template Reference

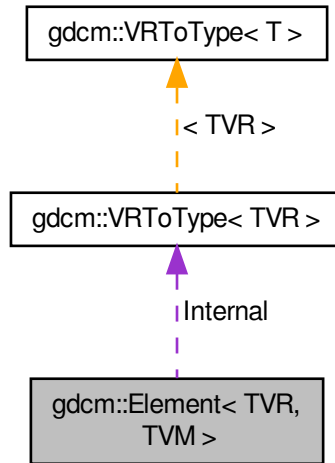
[Element](#) class.

```
#include <gdcmmElement.h>
```

Inheritance diagram for gdcmm::Element< TVR, TVM >:



Collaboration diagram for `gdcM::Element< TVR, TVM >`:



Public Types

- typedef `VRTToType< TVR >::Type` `Type`

Public Member Functions

- `DataElement GetAsDataElement ()` const
- unsigned long `GetLength ()` const
- const `VRTToType< TVR >::Type` & `GetValue` (unsigned int idx=0) const
- `VRTToType< TVR >::Type` & `GetValue` (unsigned int idx=0)
- const `VRTToType< TVR >::Type` * `GetValues ()` const
- `VRTToType< 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 `VRTToType< 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)

25.90.1 Detailed Description

template<int TVR, int TVM>class gdcm::Element< TVR, TVM >

[Element](#) class.

Note

TODO

Examples:

[csa2img.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

25.90.2 Member Typedef Documentation

25.90.2.1 template<int TVR, int TVM> typedef [VRToType](#)<TVR>::Type [gdcm::Element](#)< TVR, TVM >::Type

25.90.3 Member Function Documentation

25.90.3.1 template<int TVR, int TVM> [DataElement](#) [gdcm::Element](#)< TVR, TVM >::GetAsDataElement () const
[inline]

25.90.3.2 template<int TVR, int TVM> unsigned long [gdcm::Element](#)< TVR, TVM >::GetLength () const [inline]

25.90.3.3 template<int TVR, int TVM> const [VRToType](#)<TVR>::Type& [gdcm::Element](#)< TVR, TVM >::GetValue (unsigned int *idx* = 0) const [inline]

25.90.3.4 template<int TVR, int TVM> [VRToType](#)<TVR>::Type& [gdcm::Element](#)< TVR, TVM >::GetValue (unsigned int *idx* = 0) [inline]

25.90.3.5 template<int TVR, int TVM> const [VRToType](#)<TVR>::Type* [gdcm::Element](#)< TVR, TVM >::GetValues () const
[inline]

25.90.3.6 template<int TVR, int TVM> static VM [gdcm::Element](#)< TVR, TVM >::GetVM () [inline],[static]

25.90.3.7 template<int TVR, int TVM> static VR [gdcm::Element](#)< TVR, TVM >::GetVR () [inline],[static]

25.90.3.8 template<int TVR, int TVM> [VRToType](#)<TVR>::Type [gdcm::Element](#)< TVR, TVM >::operator[] (unsigned int *idx*) const [inline]

25.90.3.9 template<int TVR, int TVM> void [gdcm::Element](#)< TVR, TVM >::Print (std::ostream &_os) const [inline]

25.90.3.10 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Read (std::istream & _is) [inline]`

25.90.3.11 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Set (Value const & v) [inline]`

25.90.3.12 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetFromDataElement (DataElement< TVR, TVM > const & de) [inline]`

25.90.3.13 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetNoSwap (Value const & v) [inline], [protected]`

25.90.3.14 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::SetValue (typename VRToType< TVR >::Type v, unsigned int idx = 0) [inline]`

25.90.3.15 `template<int TVR, int TVM> void gdcM::Element< TVR, TVM >::Write (std::ostream & _os) const [inline]`

25.90.4 Member Data Documentation

25.90.4.1 `template<int TVR, int TVM> VRToType<TVR>::Type gdcM::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]`

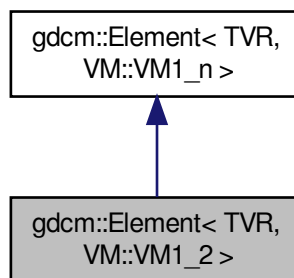
The documentation for this class was generated from the following file:

- [gdcMElement.h](#)

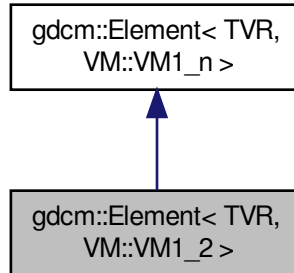
25.91 gdcM::Element< TVR, VM::VM1_2 > Class Template Reference

```
#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

25.91.1 Member Typedef Documentation

25.91.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM1_2 >::Parent`

25.91.2 Member Function Documentation

25.91.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM1_2 >::SetLength (int len) [inline]`

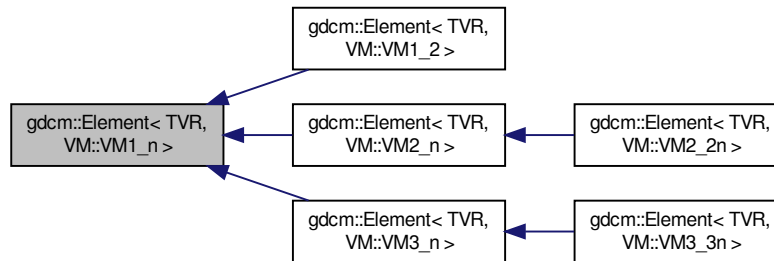
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

25.92 gdcm::Element< TVR, VM::VM1_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcmm::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)`

25.92.1 Member Typedef Documentation

25.92.1.1 `template<int TVR> typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type`

25.92.2 Constructor & Destructor Documentation

25.92.2.1 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element () [inline],[explicit]`

25.92.2.2 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::~~Element () [inline]`

25.92.2.3 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element (const Element< TVR, VM::VM1_n > & _val) [inline]`

25.92.3 Member Function Documentation

25.92.3.1 `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`.

25.92.3.2 `template<int TVR> unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength () const [inline]`

25.92.3.3 `template<int TVR> const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int idx = 0) const [inline]`

25.92.3.4 `template<int TVR> VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue (unsigned int idx = 0) [inline]`

25.92.3.5 `template<int TVR> static VM gdcm::Element< TVR, VM::VM1_n >::GetVM () [inline],[static]`

References `gdcm::VM::VM1_n`.

25.92.3.6 `template<int TVR> static VR gdcm::Element< TVR, VM::VM1_n >::GetVR () [inline],[static]`

25.92.3.7 `template<int TVR> Element& gdcm::Element< TVR, VM::VM1_n >::operator= (const Element< TVR, VM::VM1_n > &_val) [inline]`

25.92.3.8 `template<int TVR> VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::operator[] (unsigned int idx) const [inline]`

25.92.3.9 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Print (std::ostream &_os) const [inline]`

25.92.3.10 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Read (std::istream &_is) [inline]`

25.92.3.11 `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`.

25.92.3.12 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::SetArray (const Type * array, unsigned long len, bool save = false) [inline]`

25.92.3.13 `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`.

25.92.3.14 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::SetLength (unsigned long len) [inline]`

25.92.3.15 `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`.

25.92.3.16 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::SetValue (typename VRToType< TVR >::Type v, unsigned int idx = 0) [inline]`

25.92.3.17 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::Write (std::ostream & _os) const [inline]`

25.92.3.18 `template<int TVR> void gdcM::Element< TVR, VM::VM1_n >::WriteASCII (std::ostream & os) const [inline]`

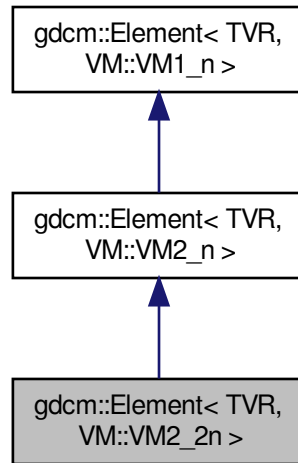
The documentation for this class was generated from the following file:

- [gdcMElement.h](#)

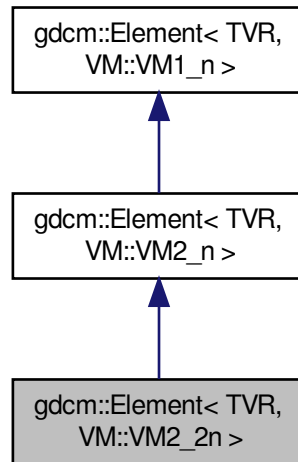
25.93 `gdcM::Element< TVR, VM::VM2_2n >` Class Template Reference

```
#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

25.93.1 Member Typedef Documentation

25.93.1.1 `template<int TVR> typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent`

25.93.2 Member Function Documentation

25.93.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM2_2n >::SetLength (int len)` `[inline]`

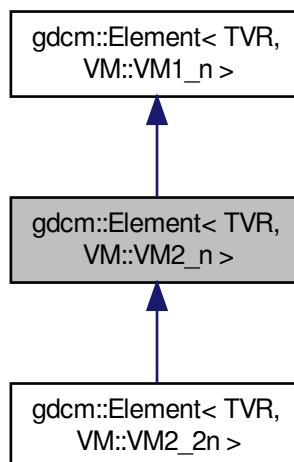
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

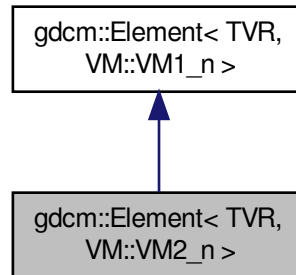
25.94 gdcm::Element< TVR, VM::VM2_n > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for gdcm::Element< TVR, VM::VM2_n >:



Collaboration diagram for gdcmm::Element< TVR, VM::VM2_n >:



Public Types

- typedef [Element](#)< TVR, [VM::VM1_n](#) > [Parent](#)

Public Member Functions

- void [SetLength](#) (int len)

Additional Inherited Members

25.94.1 Member Typedef Documentation

25.94.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcmm::Element< TVR, VM::VM2_n >::Parent`

25.94.2 Member Function Documentation

25.94.2.1 `template<int TVR> void gdcmm::Element< TVR, VM::VM2_n >::SetLength (int len) [inline]`

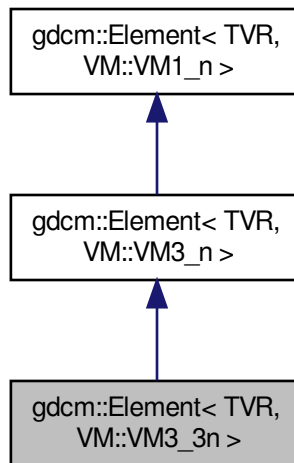
The documentation for this class was generated from the following file:

- [gdcmmElement.h](#)

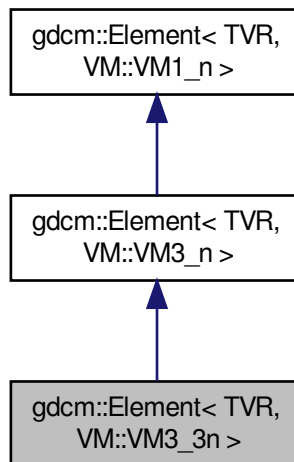
25.95 gdcmm::Element< TVR, VM::VM3_3n > Class Template Reference

```
#include <gdcmmElement.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

25.95.1 Member Typedef Documentation

25.95.1.1 `template<int TVR> typedef Element<TVR, VM::VM3_n> gdcm::Element< TVR, VM::VM3_3n >::Parent`

25.95.2 Member Function Documentation

25.95.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM3_3n >::SetLength (int len) [inline]`

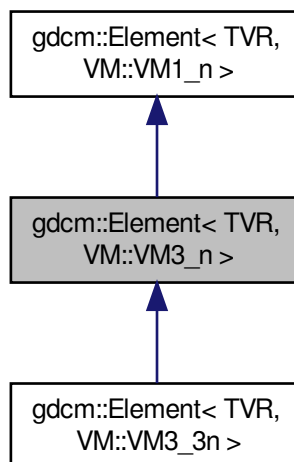
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

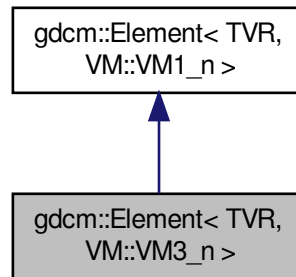
25.96 `gdcm::Element< TVR, VM::VM3_n >` Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< TVR, VM::VM3_n >`:



Collaboration diagram for `gdcmm::Element< TVR, VM::VM3_n >`:



Public Types

- typedef [Element< TVR, VM::VM1_n >](#) [Parent](#)

Public Member Functions

- void [SetLength](#) (int len)

Additional Inherited Members

25.96.1 Member Typedef Documentation

25.96.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcmm::Element< TVR, VM::VM3_n >::Parent`

25.96.2 Member Function Documentation

25.96.2.1 `template<int TVR> void gdcmm::Element< TVR, VM::VM3_n >::SetLength (int len) [inline]`

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

- [gdcmmElement.h](#)

25.97 gdcmm::Element< VR::AS, VM::VM5 > Class Template Reference

```
#include <gdcmmElement.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)]

25.97.1 Member Function Documentation

25.97.1.1 unsigned long [gdcm::Element< VR::AS, VM::VM5 >::GetLength](#) () const [\[inline\]](#)

25.97.1.2 void [gdcm::Element< VR::AS, VM::VM5 >::Print](#) (std::ostream &_os) const [\[inline\]](#)

25.97.2 Member Data Documentation

25.97.2.1 char [gdcm::Element< VR::AS, VM::VM5 >::Internal](#)[[VMToLength](#)< [VM::VM5](#)>::Length *sizeof([VRToType](#)< [VR::AS](#)>::Type)]

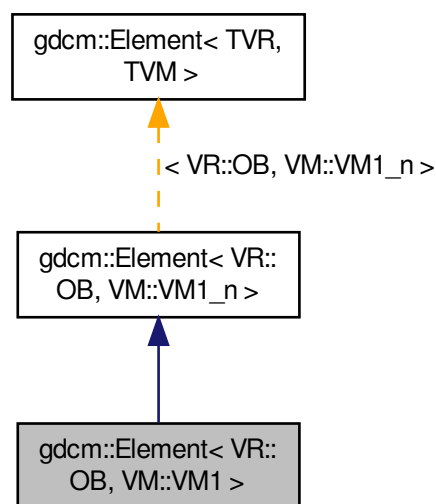
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

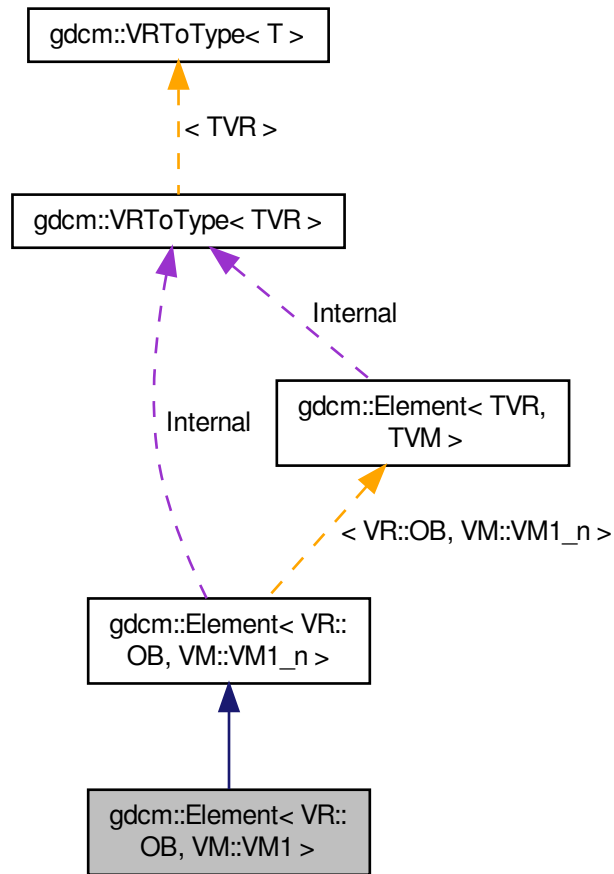
25.98 gdcm::Element< VR::OB, VM::VM1 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for [gdcm::Element< VR::OB, VM::VM1 >](#):



Collaboration diagram for `gdcM::Element< VR::OB, VM::VM1 >`:



Additional Inherited Members

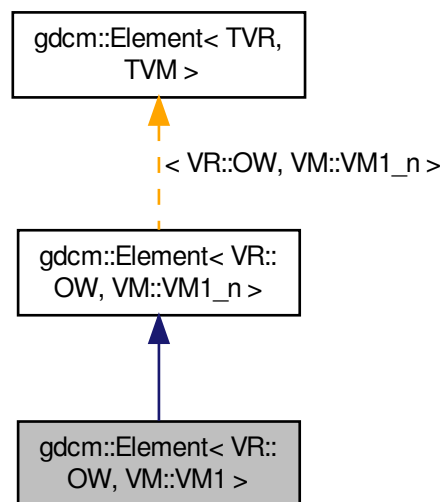
The documentation for this class was generated from the following file:

- [gdcMElement.h](#)

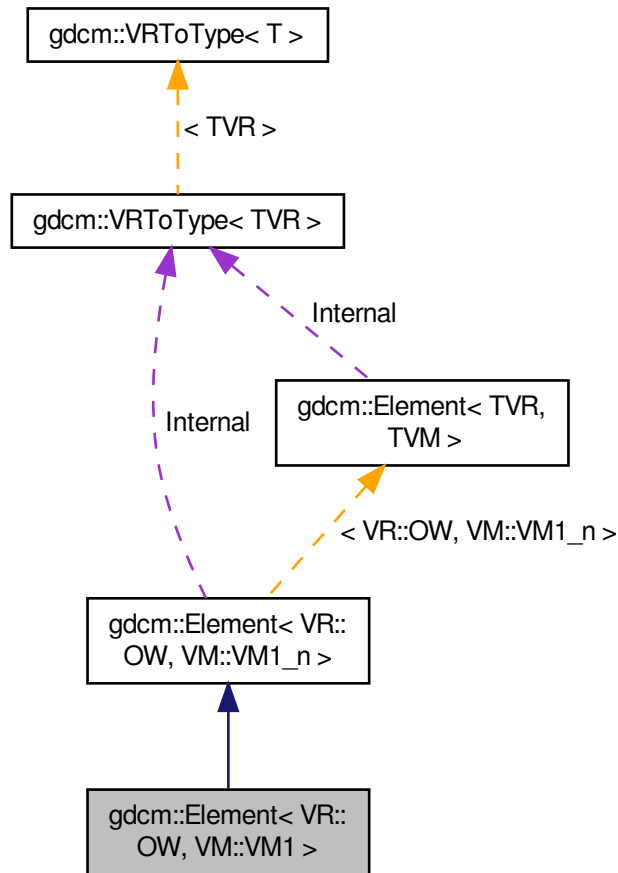
25.99 `gdcM::Element< VR::OW, VM::VM1 >` Class Template Reference

```
#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](#)

25.100 `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>
```

25.100.1 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](#)

25.101 `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](#)

25.102 `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](#)

25.103 `gdcm::EncapsulatedDocument` Class Reference

[EncapsulatedDocument.](#)

```
#include <gdcmEncapsulatedDocument.h>
```

Public Member Functions

- [EncapsulatedDocument\(\)](#)

25.103.1 Detailed Description

[EncapsulatedDocument.](#)

25.103.2 Constructor & Destructor Documentation

25.103.2.1 `gdcm::EncapsulatedDocument::EncapsulatedDocument()` `[inline]`

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

- [gdcmEncapsulatedDocument.h](#)

25.104 `gdcm::EncodingImplementation< T >` Class Template Reference

[EncodingImplementation.](#)

```
#include <gdcmElement.h>
```

25.104.1 Detailed Description

```
template<int T>class gdcm::EncodingImplementation< T >
```

[EncodingImplementation.](#)

Note

TODO

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

- [gdcmElement.h](#)

25.105 `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)

25.105.1 Member Function Documentation

25.105.1.1 `template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Read (T * data, unsigned long length, std::istream &_is)` `[inline]`, `[static]`

25.105.1.2 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength (T * data, unsigned int & length, std::istream & _is) [inline], [static]`

References `gdcm::backslash()`.

25.105.1.3 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap (T * data, unsigned long length, std::istream & _is) [inline], [static]`

25.105.1.4 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::Write (const T * data, unsigned long length, std::ostream & _os) [inline], [static]`

25.105.1.5 `template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write (const float * data, unsigned long length, std::ostream & _os) [inline]`

References `gdcm::to_string()`.

25.105.1.6 `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](#)

25.106 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)`

25.106.1 Member Function Documentation

25.106.1.1 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::Read (T * data, unsigned long length, std::istream & _is) [inline], [static]`

References `gdcm::SwapperNoOp::SwapArray()`.

25.106.1.2 `template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength (T * data, unsigned int & length, std::istream & _is) [inline], [static]`

25.106.1.3 `template<typename T > static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap (T * data, unsigned long length, std::istream & _is) [inline], [static]`

25.106.1.4 `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()`.

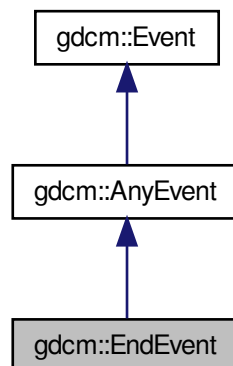
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

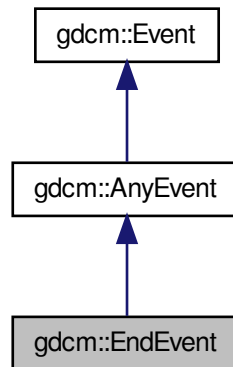
25.107 gdcm::EndEvent Class Reference

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

Inheritance diagram for `gdcm::EndEvent`:



Collaboration diagram for gdcmm::EndEvent:



Additional Inherited Members

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

- [gdcmmEvent.h](#)

25.108 gdcmm::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 <gdcmmEnumeratedValues.h>
```

Public Member Functions

- [EnumeratedValues](#) ()

25.108.1 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](#).

25.108.2 Constructor & Destructor Documentation

25.108.2.1 `gdcm::EnumeratedValues::EnumeratedValues ()` `[inline]`

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

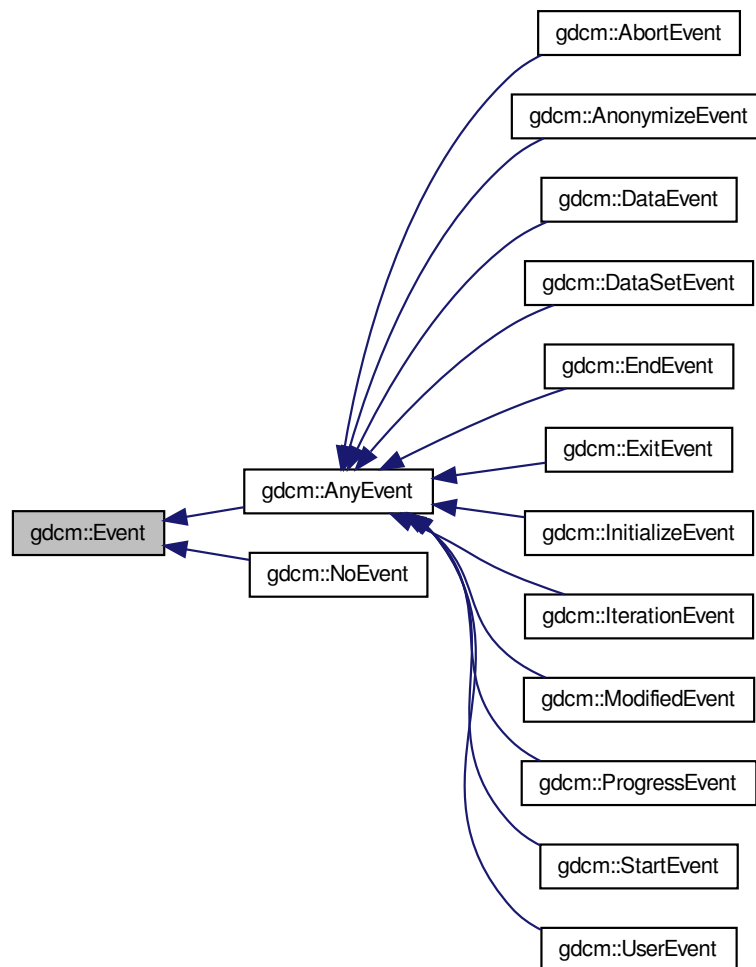
- [gdcmEnumeratedValues.h](#)

25.109 `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

25.109.1 Detailed Description

superclass for callback/observer methods

See Also

[Command Subject](#)

25.109.2 Constructor & Destructor Documentation

25.109.2.1 `gdcm::Event::Event ()`

25.109.2.2 `gdcm::Event::Event (const Event &)`

25.109.2.3 `virtual gdcm::Event::~~Event () [virtual]`

25.109.3 Member Function Documentation

25.109.3.1 `virtual bool gdcm::Event::CheckEvent (const Event *) const [pure virtual]`

Check if given event matches or derives from this event.

25.109.3.2 `virtual const char* gdcm::Event::GetEventName (void) const [pure virtual]`

Return the StringName associated with the event.

Implemented in [gdcm::ProgressEvent](#), [gdcm::DataSetEvent](#), [gdcm::AnonymizeEvent](#), and [gdcm::DataEvent](#).

25.109.3.3 `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::ProgressEvent](#), [gdcm::DataSetEvent](#), [gdcm::AnonymizeEvent](#), and [gdcm::DataEvent](#).

25.109.3.4 `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:

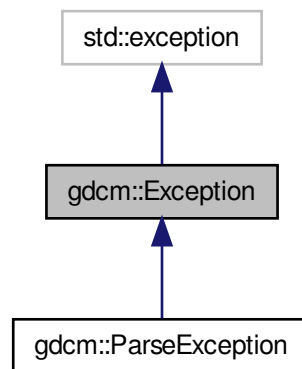
- [gdcmEvent.h](#)

25.110 gdcm::Exception Class Reference

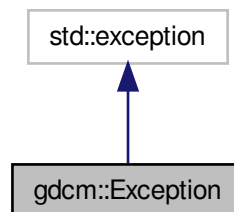
[Exception.](#)

```
#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

25.110.1 Detailed Description

[Exception](#).

Standard exception handling object.

Note

Its copy-constructor and assignment operator are generated by the compiler.

25.110.2 Constructor & Destructor Documentation

25.110.2.1 `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.

25.110.2.2 `virtual gdcm::Exception::~~Exception () throw) [inline], [virtual]`

25.110.3 Member Function Documentation

25.110.3.1 `const char* gdcm::Exception::GetDescription () const [inline]`

Return the Description.

Referenced by `gdcm::SequenceOfItems::Read()`.

25.110.3.2 `const char* gdcm::Exception::what () const throw) [inline]`

what implementation

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

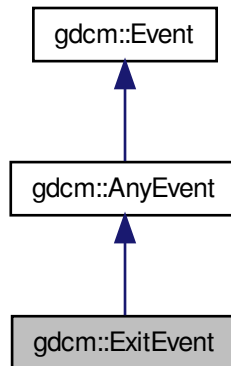
The documentation for this class was generated from the following file:

- [gdcmException.h](#)

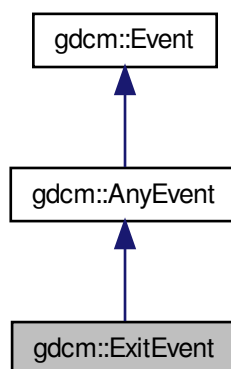
25.111 gdcm::ExitEvent Class Reference

```
#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:

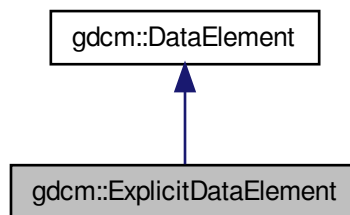
- [gdcmEvent.h](#)

25.112 gdcm::ExplicitDataElement Class Reference

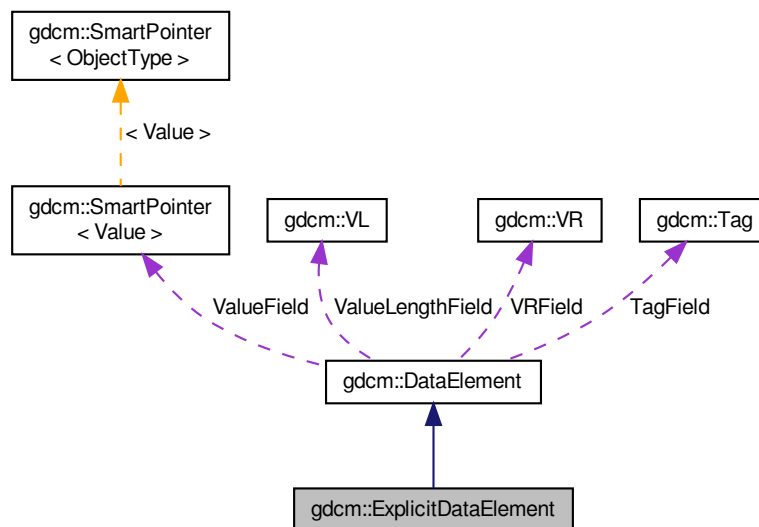
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#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)`
- `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

25.112.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

Note

bla

25.112.2 Member Function Documentation

25.112.2.1 `VL gdcM::ExplicitDataElement::GetLength () const`

25.112.2.2 `template<typename TSwap > std::istream& gdcM::ExplicitDataElement::Read (std::istream & is)`

25.112.2.3 `template<typename TSwap > std::istream& gdcM::ExplicitDataElement::ReadPreValue (std::istream & is)`

25.112.2.4 `template<typename TSwap > std::istream& gdcM::ExplicitDataElement::ReadValue (std::istream & is)`

25.112.2.5 `template<typename TSwap > std::istream& gdcM::ExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

25.112.2.6 `template<typename TSwap > const std::ostream& gdcM::ExplicitDataElement::Write (std::ostream & os) const`

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

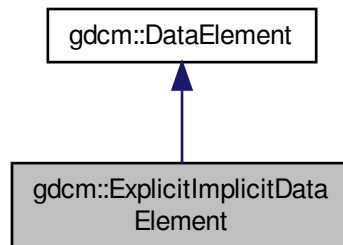
- [gdcMExplicitDataElement.h](#)

25.113 gdcM::ExplicitImplicitDataElement Class Reference

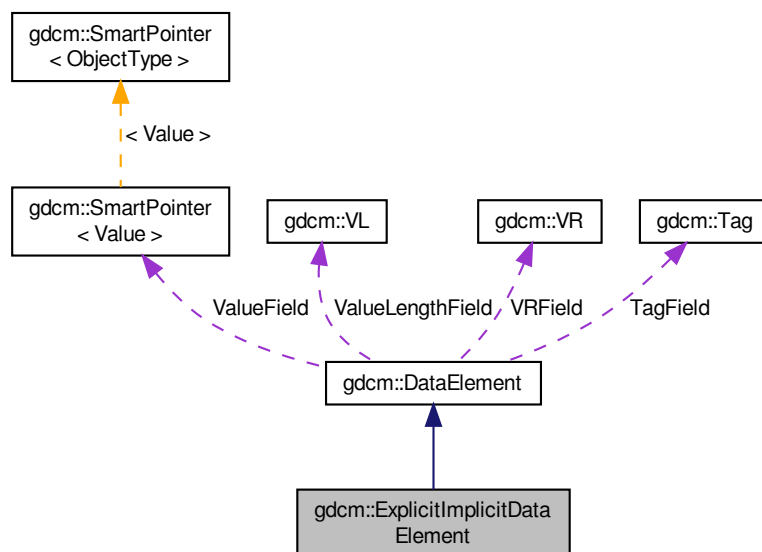
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#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)`
- `template<typename TSwap >`
`std::istream & ReadWithLength (std::istream &is, VL &length)`

Additional Inherited Members

25.113.1 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.

25.113.2 Member Function Documentation

25.113.2.1 `VL gdcm::ExplicitImplicitDataElement::GetLength () const`

25.113.2.2 `template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::Read (std::istream & is)`

25.113.2.3 `template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadPreValue (std::istream & is)`

25.113.2.4 `template<typename TSwap > std::istream& gdcm::ExplicitImplicitDataElement::ReadValue (std::istream & is)`

25.113.2.5 `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](#)

25.114 [gdcm::Fiducials](#) Class Reference

[Fiducials](#).

```
#include <gdcmFiducials.h>
```

Public Member Functions

- [Fiducials](#) ()

25.114.1 Detailed Description

[Fiducials](#).

25.114.2 Constructor & Destructor Documentation

25.114.2.1 gdcm::Fiducials::Fiducials () [inline]

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

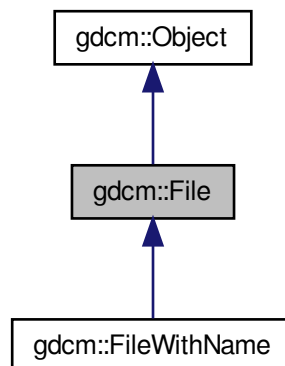
- [gdcmFiducials.h](#)

25.115 gdcm::File Class Reference

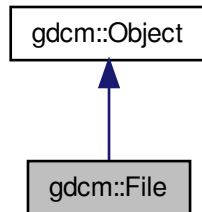
a DICOM [File](#) See PS 3.10 [File](#): A [File](#) is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the [File](#). Files are identified by a unique [File](#) ID and may be written, read and/or deleted.

```
#include <gdcmFile.h>
```

Inheritance diagram for gdcm::File:



Collaboration diagram for `gdcm::File`:



Public Member Functions

- [File](#) ()
- [~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

25.115.1 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 be written, read and/or deleted.

See Also

[Reader Writer](#)

Examples:

[ChangeSequenceUltrasound.cxx](#), [CreateJPIPDataSet.cxx](#), [DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [Extracting_All_Resolution.cxx](#), [ExtractOneFrame.cs](#), [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](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrintPatientName.cs](#), and [StreamImageReaderTest.cxx](#).

25.115.2 Constructor & Destructor Documentation

25.115.2.1 `gdcm::File::File ()` `[inline]`

25.115.2.2 `gdcm::File::~~File ()` `[inline]`

25.115.3 Member Function Documentation

25.115.3.1 `const DataSet& gdcm::File::GetDataSet () const` `[inline]`

Get Data Set.

Examples:

[ChangeSequenceUltrasound.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpImageHeaderInfo.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](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

25.115.3.2 `DataSet& gdcm::File::GetDataSet ()` `[inline]`

Get Data Set.

25.115.3.3 `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](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

Referenced by `gdcm::operator<<()`.

25.115.3.4 **FileMetaInformation& gdcmm::File::GetHeader ()** `[inline]`

Get [File](#) Meta Information.

25.115.3.5 **std::istream& gdcmm::File::Read (std::istream & *is*)**

Read.

25.115.3.6 **void gdcmm::File::SetDataSet (const DataSet & *ds*)** `[inline]`

Set Data Set.

25.115.3.7 **void gdcmm::File::SetHeader (const FileMetaInformation & *fmi*)** `[inline]`

Set [File](#) Meta Information.

25.115.3.8 **std::ostream const& gdcmm::File::Write (std::ostream & *os*) const**

Write.

25.115.4 Friends And Related Function Documentation

25.115.4.1 **std::ostream& operator<< (std::ostream & *os*, const File & *val*)** `[friend]`

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

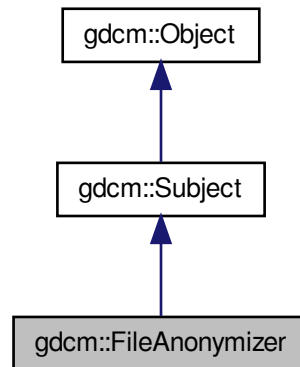
- [gdcmmFile.h](#)

25.116 gdcmm::FileAnonymizer Class Reference

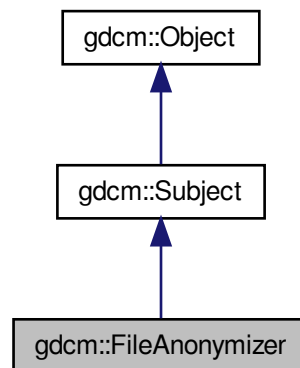
[FileAnonymizer](#).

```
#include <gdcmmFileAnonymizer.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)

- void [Replace](#) ([Tag](#) const &t, const char *value, [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

25.116.1 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 data into memory and should consume much less memory than [gdcm::Anonymizer](#)

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

Examples:

[FileAnonymize.cs](#).

25.116.2 Constructor & Destructor Documentation

25.116.2.1 [gdcm::FileAnonymizer::FileAnonymizer](#) ()

25.116.2.2 [gdcm::FileAnonymizer::~~FileAnonymizer](#) ()

25.116.3 Member Function Documentation

25.116.3.1 void [gdcm::FileAnonymizer::Empty](#) ([Tag](#) const & t)

Make [Tag](#) t empty Warning: does not handle SQ element

25.116.3.2 void [gdcm::FileAnonymizer::Remove](#) ([Tag](#) const & t)

remove a tag (even a SQ can be removed)

25.116.3.3 void [gdcm::FileAnonymizer::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 WARNING: Do not ever try to write a value in a SQ Data [Element](#) !

25.116.3.4 void gdcm::FileAnonymizer::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

25.116.3.5 void gdcm::FileAnonymizer::SetInputFileName (const char * *filename_native*)

Set input filename.

25.116.3.6 void gdcm::FileAnonymizer::SetOutputFileName (const char * *filename_native*)

Set output filename.

25.116.3.7 bool gdcm::FileAnonymizer::Write ()

Write the output file.

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

- [gdcmFileAnonymizer.h](#)

25.117 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](#) ()

25.117.1 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 derivation applied to the new image data from the original. Both these context groups are extensible.

[File](#) Derivation is compulsory when creating a lossy derived image.

Examples:

[GenFakelImage.cxx](#).

25.117.2 Constructor & Destructor Documentation

25.117.2.1 `gdcm::FileDerivation::FileDerivation ()`

25.117.2.2 `gdcm::FileDerivation::~~FileDerivation ()`

25.117.3 Member Function Documentation

25.117.3.1 `bool gdcm::FileDerivation::AddDerivationDescription ()` [protected]

25.117.3.2 `bool gdcm::FileDerivation::AddPurposeOfReferenceCodeSequence (DataSet & ds)` [protected]

25.117.3.3 `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:

[GenFakelImage.cxx](#).

25.117.3.4 `bool gdcm::FileDerivation::AddSourceImageSequence ()` [protected]

25.117.3.5 `bool gdcm::FileDerivation::Derive ()`

Change.

Examples:

[GenFakelImage.cxx](#).

25.117.3.6 `File& gdcm::FileDerivation::GetFile () [inline]`

Examples:

[GenFakelImage.cxx](#).

25.117.3.7 `const File& gdcm::FileDerivation::GetFile () const [inline]`

25.117.3.8 `void gdcm::FileDerivation::SetDerivationCodeSequenceCodeValue (unsigned int codevalue)`

Specify the Derivation Code Sequence Code [Value](#). Eg 113040.

Examples:

[GenFakelImage.cxx](#).

25.117.3.9 `void gdcm::FileDerivation::SetDerivationDescription (const char * dd)`

Specify the Derivation Description. Eg "lossy conversion".

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

Set/Get [File](#).

Examples:

[GenFakelImage.cxx](#).

25.117.3.11 `void gdcm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue (unsigned int codevalue)`

Specify the Purpose Of Reference Code [Value](#). Eg. 121320.

Examples:

[GenFakelImage.cxx](#).

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

- [gdcmFileDerivation.h](#)

25.118 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 representation, 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)

25.118.1 Detailed Description

[FileExplicitFilter](#) class After changing a file from Implicit to Explicit representation (see [ImageChangeTransferSyntax](#)) one operation is to make sure the [VR](#) of each DICOM attribute are accurate and do match the one from PS 3.6. Indeed when a file is written in Implicit representation, the [VR](#) is not stored directly in the file.

Warning

changing an implicit dataset to an explicit dataset is NOT a trivial task of simply changing the [VR](#) to the dict one:

- One has to make sure SQ is properly set
- One has to recompute the explicit length SQ
- One has to make sure that [VR](#) is valid for the encoding
- One has to make sure that [VR](#) 16bits can store the original value length

Examples:

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

25.118.2 Constructor & Destructor Documentation

25.118.2.1 `gdcm::FileExplicitFilter::FileExplicitFilter () [inline]`

25.118.2.2 `gdcm::FileExplicitFilter::~~FileExplicitFilter () [inline]`

25.118.3 Member Function Documentation

25.118.3.1 `bool gdcm::FileExplicitFilter::Change ()`

Set FMI Transfer Syntax.

Change

Examples:

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

25.118.3.2 `bool gdcm::FileExplicitFilter::ChangeFMI ()` `[protected]`

25.118.3.3 `File& gdcm::FileExplicitFilter::GetFile ()` `[inline]`

25.118.3.4 `bool gdcm::FileExplicitFilter::ProcessDataSet (DataSet & ds, Dicts const & dicts)` `[protected]`

25.118.3.5 `void gdcm::FileExplicitFilter::SetChangePrivateTags (bool b)` `[inline]`

Decide whether or not to [VR](#)ify private tags.

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

Set/Get [File](#).

Examples:

[GenAllVR.cxx](#), and [LargeVRDSExplicit.cxx](#).

25.118.3.7 `void gdcm::FileExplicitFilter::SetRecomputeItemLength (bool b)`

By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:

25.118.3.8 `void gdcm::FileExplicitFilter::SetRecomputeSequenceLength (bool b)`

25.118.3.9 `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:

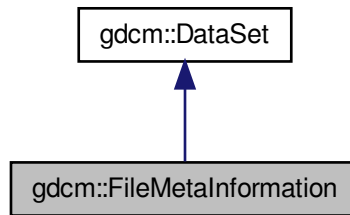
- [gdcmFileExplicitFilter.h](#)

25.119 gdcm::FileMetaInformation Class Reference

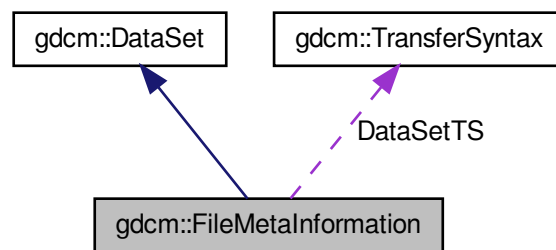
Class to represent a [File](#) Meta Information.

```
#include <gdcmFileMetaInformation.h>
```

Inheritance diagram for `gdcm::FileMetaInformation`:



Collaboration diagram for `gdcm::FileMetaInformation`:



Public Member Functions

- [FileMetaInformation](#) ()
- [FileMetaInformation](#) ([FileMetaInformation](#) const &fmi)
- [~FileMetaInformation](#) ()
- void [FillFromDataSet](#) ([DataSet](#) const &ds)
 - Construct a [FileMetaInformation](#) from an already existing [DataSet](#):*
- const [TransferSyntax](#) & [GetDataSetTransferSyntax](#) () const
- [VL](#) [GetFullLength](#) () const
- [MediaStorage](#) [GetMediaStorage](#) () 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

25.119.1 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:

[ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAnd-DumpDICOMDIR.cxx](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

25.119.2 Constructor & Destructor Documentation

25.119.2.1 `gdcm::FileMetaInformation::FileMetaInformation ()` `[inline]`

25.119.2.2 `gdcm::FileMetaInformation::~~FileMetaInformation ()` `[inline]`

25.119.2.3 `gdcm::FileMetaInformation::FileMetaInformation (FileMetaInformation const & fmi)` `[inline]`

References [DataSetMS](#), [DataSetTS](#), and [MetaInformationTS](#).

25.119.3 Member Function Documentation

25.119.3.1 `static void gdcm::FileMetaInformation::AppendImplementationClassUID (const char * imp)` `[static]`

25.119.3.2 `void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ()` `[protected]`

25.119.3.3 `void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ()` `[protected]`

25.119.3.4 `void gdcm::FileMetaInformation::Default ()` `[protected]`

25.119.3.5 `void gdcm::FileMetaInformation::FillFromDataSet (DataSet const & ds)`

Construct a [FileMetaInformation](#) from an already existing [DataSet](#):

25.119.3.6 `const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax () const` `[inline]`

Examples:

[GetJPEGSamplePrecision.cxx](#), and [MergeTwoFiles.cxx](#).

25.119.3.7 `static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion () [static],[protected]`

25.119.3.8 `VL gdcm::FileMetaInformation::GetFullLength () const [inline]`

References `gdcm::VL::GetLength()`.

25.119.3.9 `static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID () [static],[protected]`

25.119.3.10 `static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName () [static],[protected]`

25.119.3.11 `static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle () [static],[protected]`

25.119.3.12 `static const char* gdcm::FileMetaInformation::GetImplementationClassUID () [static]`

25.119.3.13 `static const char* gdcm::FileMetaInformation::GetImplementationVersionName () [static]`

25.119.3.14 `MediaStorage gdcm::FileMetaInformation::GetMediaStorage () const`

25.119.3.15 `TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS () const [inline]`

25.119.3.16 `const Preamble& gdcm::FileMetaInformation::GetPreamble () const [inline]`

Get [Preamble](#).

Referenced by `gdcm::operator<<()`.

25.119.3.17 `Preamble& gdcm::FileMetaInformation::GetPreamble () [inline]`

25.119.3.18 `static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle () [static]`

25.119.3.19 `void gdcm::FileMetaInformation::Insert (const DataElement & de) [inline]`

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

25.119.3.20 `bool gdcm::FileMetaInformation::IsValid () const [inline]`

25.119.3.21 `std::istream& gdcm::FileMetaInformation::Read (std::istream & is)`

Read.

25.119.3.22 `std::istream& gdcm::FileMetaInformation::ReadCompat (std::istream & is)`

25.119.3.23 `template<typename TSwap > std::istream& gdcm::FileMetaInformation::ReadCompatInternal (std::istream & is) [protected]`

25.119.3.24 `void gdcM::FileMetaInformation::Replace (const DataElement & de) [inline]`

Examples:

[LargeVRDSExplicit.cxx](#).

References `gdcM::DataElement::GetTag()`.

25.119.3.25 `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](#), [GenAIIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [pmsct_rgb1.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

25.119.3.26 `static void gdcM::FileMetaInformation::SetImplementationClassUID (const char * imp) [static]`

Override the GDCM default values:

25.119.3.27 `static void gdcM::FileMetaInformation::SetImplementationVersionName (const char * version) [static]`

25.119.3.28 `void gdcM::FileMetaInformation::SetPreamble (const Preamble & p) [inline]`

25.119.3.29 `static void gdcM::FileMetaInformation::SetSourceApplicationEntityTitle (const char * title) [static]`

Examples:

[FixJAIBugJPEGLS.cxx](#).

25.119.3.30 `std::ostream& gdcM::FileMetaInformation::Write (std::ostream & os) const`

Write.

25.119.4 Friends And Related Function Documentation

25.119.4.1 `std::ostream& operator<< (std::ostream & _os, const FileMetaInformation & _val) [friend]`

25.119.5 Member Data Documentation

25.119.5.1 `MediaStorage::MSType gdcM::FileMetaInformation::DataSetMS [protected]`

Referenced by `FileMetaInformation()`.

25.119.5.2 `TransferSyntax gdcM::FileMetaInformation::DataSetTS [protected]`

Referenced by `FileMetaInformation()`.

25.119.5.3 **TransferSyntax::NegociatedType** gdcm::FileMetaInformation::MetaInformationTS [protected]

Referenced by FileMetaInformation().

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

- [gdcmFileMetaInformation.h](#)

25.120 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)

25.120.1 Detailed Description

Class to manipulate file name's.

Note

OS independant representation of a filename (to query path, name and extension from a filename)

Examples:

[ClinicalTrialIdentificationWorkflow.cs](#).

25.120.2 Constructor & Destructor Documentation

25.120.2.1 `gdcm::Filename::Filename (const char * filename = " ") [inline]`

25.120.3 Member Function Documentation

25.120.3.1 `bool gdcm::Filename::EndWith (const char ending[]) const`

Does the filename ends with a particular string ?

25.120.3.2 `const char* gdcm::Filename::GetExtension ()`

return only the extension part of a filename

25.120.3.3 `const char* gdcm::Filename::GetFileName () const [inline]`

Return the full filename.

25.120.3.4 `const char* gdcm::Filename::GetName ()`

return only the name part of a filename

25.120.3.5 `const char* gdcm::Filename::GetPath ()`

Return only the path component of a filename.

25.120.3.6 `bool gdcm::Filename::IsEmpty () const [inline]`

return whether the filename is empty

25.120.3.7 `bool gdcm::Filename::IsIdentical (Filename const & fn) const`

25.120.3.8 `static const char* gdcm::Filename::Join (const char * path, const char * filename) [static]`

Join two paths NOT THREAD SAFE

25.120.3.9 `gdcm::Filename::operator const char * () const [inline]`

Simple operator to allow `Filename myfilename("...")`; `const char * s = myfilename`;

25.120.3.10 `const char* gdcm::Filename::ToUnixSlashes ()`

Convert backslash (windows style) to UNIX style slash.

25.120.3.11 `const char* gdcm::Filename::ToWindowsSlashes ()`

Convert forward slash (UNIX style) to windows style slash.

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

- [gdcmFilename.h](#)

25.121 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.

25.121.1 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 `sprintf` of `Pattern` using value `i`

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

25.121.2 Member Typedef Documentation

25.121.2.1 `typedef std::vector<FilenameType> gdcm::FilenameGenerator::FileNamesType`

25.121.2.2 `typedef std::string gdcm::FilenameGenerator::FilenameType`

25.121.2.3 `typedef FileNamesType::size_type gdcm::FilenameGenerator::SizeType`

25.121.3 Constructor & Destructor Documentation

25.121.3.1 `gdcm::FilenameGenerator::FilenameGenerator ()` `[inline]`

25.121.3.2 `gdcm::FilenameGenerator::~~FilenameGenerator ()` `[inline]`

25.121.4 Member Function Documentation

25.121.4.1 `bool gdcm::FilenameGenerator::Generate ()`

Generate (return success)

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

25.121.4.2 `const char* gdcm::FilenameGenerator::GetFilename (SizeType n) const`

Get a particular filename (call after Generate)

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

25.121.4.3 `FileNamesType const& gdcm::FilenameGenerator::GetFileNames () const` `[inline]`

Return all filenames.

25.121.4.4 `SizeType gdcm::FilenameGenerator::GetNumberOfFileNames () const`

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

25.121.4.5 `const char* gdcm::FilenameGenerator::GetPattern () const` `[inline]`

25.121.4.6 `const char* gdcm::FilenameGenerator::GetPrefix () const` `[inline]`

25.121.4.7 `void gdcm::FilenameGenerator::SetNumberOfFileNames (SizeType nfiles)`

Set/Get the number of filenames to generate.

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

25.121.4.8 `void gdcm::FilenameGenerator::SetPattern (const char * pattern)` `[inline]`

Set/Get pattern.

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

25.121.4.9 `void gdcm::FilenameGenerator::SetPrefix (const char * prefix)` `[inline]`

Set/Get prefix.

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

- [gdcmFilenameGenerator.h](#)

25.122 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)`

25.122.1 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.

25.122.2 Member Typedef Documentation

25.122.2.1 `typedef std::vector<FileType> gdcm::FileSet::FileType`

25.122.2.2 `typedef std::string gdcm::FileSet::FileType`

25.122.3 Constructor & Destructor Documentation

25.122.3.1 `gdcm::FileSet::FileSet () [inline]`

25.122.4 Member Function Documentation

25.122.4.1 `void gdcm::FileSet::AddFile (File const &) [inline]`

Deprecated . Does nothing

25.122.4.2 `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.

25.122.4.3 `FileType const& gdcm::FileSet::GetFiles () const [inline]`

25.122.4.4 `void gdcm::FileSet::SetFiles (FileType const & files)`

25.122.5 Friends And Related Function Documentation

25.122.5.1 `std::ostream& operator<< (std::ostream &_os, const FileSet &d) [friend]`

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

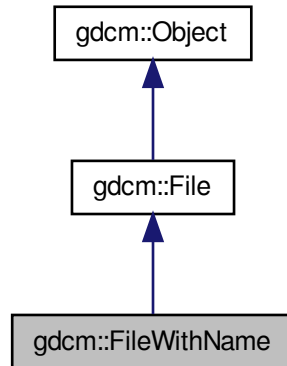
- [gdcmFileSet.h](#)

25.123 gdcm::FileWithName Class Reference

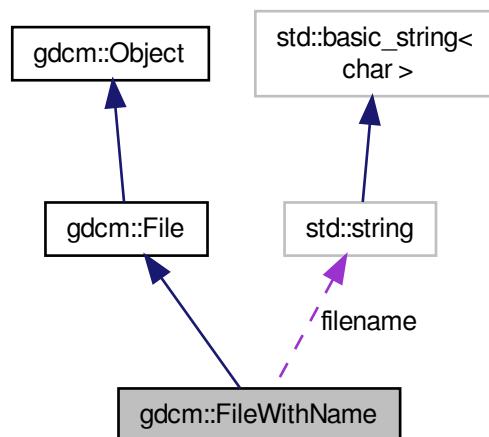
[FileWithName.](#)

```
#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

25.123.1 Detailed Description

[FileWithName.](#)

Backward only class do not use in newer code

25.123.2 Constructor & Destructor Documentation

25.123.2.1 `gdcm::FileWithName::FileWithName (File & f)` `[inline]`

25.123.3 Member Data Documentation

25.123.3.1 `std::string gdcm::FileWithName::filename`

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

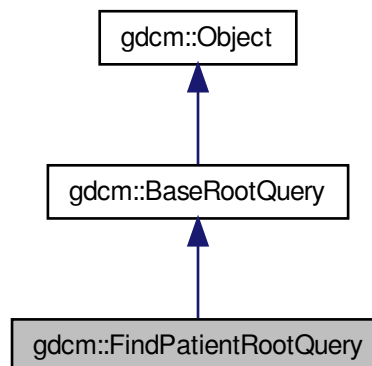
- [gdcmSerieHelper.h](#)

25.124 gdcm::FindPatientRootQuery Class Reference

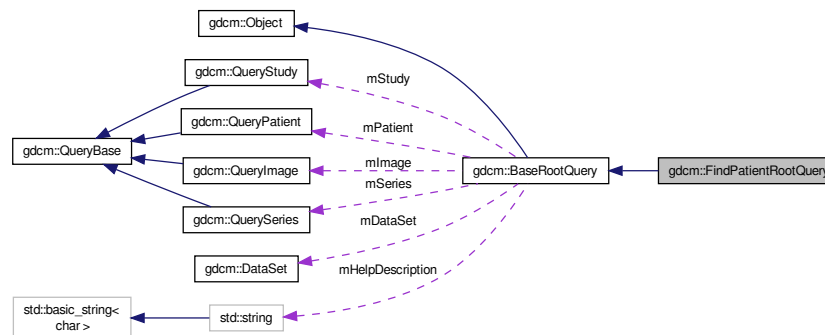
PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

```
#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

25.124.1 Detailed Description

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

25.124.2 Constructor & Destructor Documentation

25.124.2.1 gdcm::FindPatientRootQuery::FindPatientRootQuery ()

25.124.3 Member Function Documentation

25.124.3.1 [UIDs::TSName](#) gdcm::FindPatientRootQuery::GetAbstractSyntaxUID () const [virtual]

Implements [gdcm::BaseRootQuery](#).

25.124.3.2 `std::vector<Tag> gdcmm::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 [gdcmm::BaseRootQuery](#).

25.124.3.3 `void gdcmm::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 dcm4k

Implements [gdcmm::BaseRootQuery](#).

25.124.3.4 `bool gdcmm::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 [gdcmm::BaseRootQuery](#).

25.124.4 Friends And Related Function Documentation

25.124.4.1 `friend class QueryFactory` `[friend]`

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

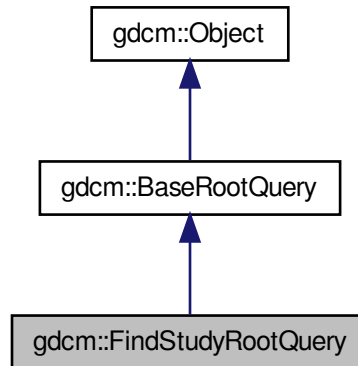
- [gdcmmFindPatientRootQuery.h](#)

25.125 gdcmm::FindStudyRootQuery Class Reference

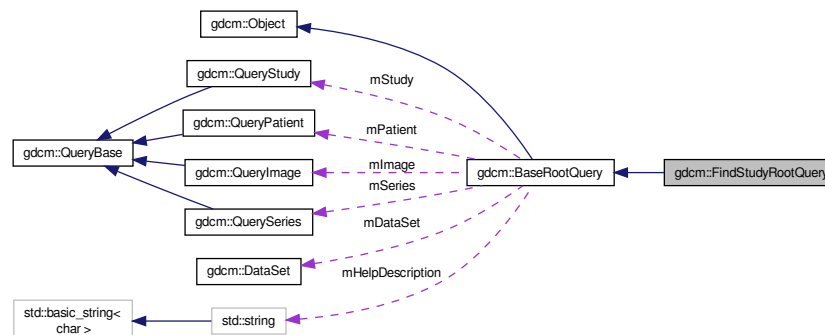
[FindStudyRootQuery](#) contains: the class which will produce a dataset for C-FIND with study root.

```
#include <gdcmmFindStudyRootQuery.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

25.125.1 Detailed Description

[FindStudyRootQuery](#) contains: the class which will produce a dataset for C-FIND with study root.

25.125.2 Constructor & Destructor Documentation

25.125.2.1 `gdcM::FindStudyRootQuery::FindStudyRootQuery ()`

25.125.3 Member Function Documentation

25.125.3.1 `UIDs::TSName gdcM::FindStudyRootQuery::GetAbstractSyntaxUID () const` `[virtual]`

Implements [gdcM::BaseRootQuery](#).

25.125.3.2 `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](#).

25.125.3.3 `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](#).

25.125.3.4 `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](#).

25.125.4 Friends And Related Function Documentation

25.125.4.1 `friend class QueryFactory` `[friend]`

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

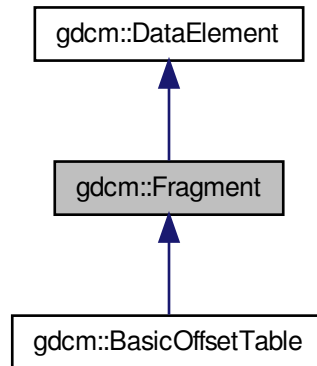
- [gdcMFindStudyRootQuery.h](#)

25.126 gdcM::Fragment Class Reference

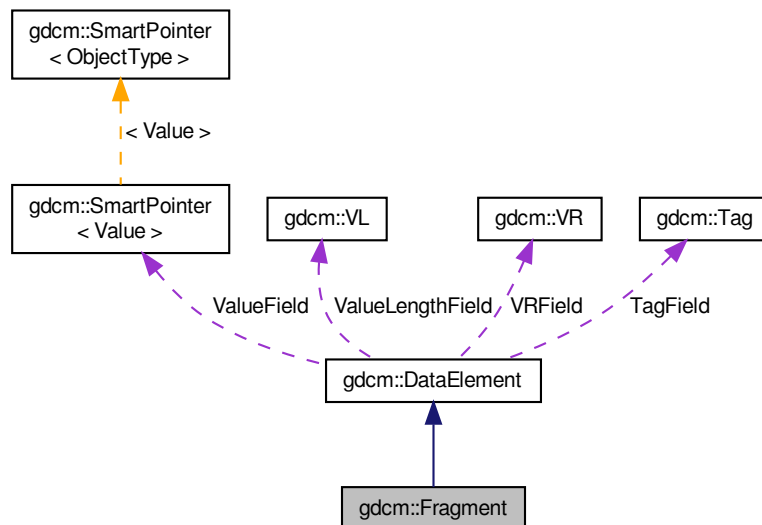
Class to represent a [Fragment](#).

```
#include <gdcmFragment.h>
```

Inheritance diagram for gdcm::Fragment:



Collaboration diagram for gdcm::Fragment:



Public Member Functions

- [Fragment](#) ()
- [VL 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

25.126.1 Detailed Description

Class to represent a [Fragment](#).

Examples:

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

25.126.2 Constructor & Destructor Documentation

25.126.2.1 `gdcm::Fragment::Fragment ()` [`inline`]

25.126.3 Member Function Documentation

25.126.3.1 `VL gdcm::Fragment::GetLength () const` [`inline`]

References `gdcm::VL::GetLength()`.

25.126.3.2 `template<typename TSwap > std::istream& gdcm::Fragment::Read (std::istream & is)` [`inline`]

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

25.126.3.3 `template<typename TSwap > std::istream& gdcm::Fragment::ReadBacktrack (std::istream & is)` [`inline`]

References `gdcmErrorMacro`, `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

25.126.3.4 `template<typename TSwap > std::istream& gdcm::Fragment::ReadPreValue (std::istream & is)` [`inline`]

25.126.3.5 `template<typename TSwap > std::istream& gdcm::Fragment::ReadValue (std::istream & is)` [`inline`]

References `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

25.126.3.6 `template<typename TSwap > std::ostream& gdcm::Fragment::Write (std::ostream & os) const` `[inline]`

References `gdcm::ByteValue::GetLength()`, `gdcm::VL::Write()`, and `gdcm::ByteValue::Write()`.

25.126.4 Friends And Related Function Documentation

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

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

- [gdcmFragment.h](#)

25.127 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 ressource file.

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Global](#) &g)

25.127.1 Detailed Description

[Global](#).

Note

[Global](#) should be included in any translation unit that will use [Dict](#) or that implements the singleton pattern. It makes sure that the [Dict](#) singleton is created before and destroyed after all other singletons in GDCM.

Examples:

[BasicAnonymizer.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

25.127.2 Constructor & Destructor Documentation

25.127.2.1 `gdcmm::Global::Global ()`

25.127.2.2 `gdcmm::Global::~~Global ()`

25.127.3 Member Function Documentation

25.127.3.1 `bool gdcmm::Global::Append (const char * path)`

Append path at the end of the path list

Warning

not thread safe !

25.127.3.2 `Defs const& gdcmm::Global::GetDefs () const`

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

Examples:

[GenerateStandardSOPClasses.cxx](#).

25.127.3.3 `Dicts const& gdcmm::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](#), and [ReadAndPrintAttributes.cxx](#).

25.127.3.4 `Dicts& gdcmm::Global::GetDicts ()`

25.127.3.5 `static Global& gdcmm::Global::GetInstance () [static]`

return the singleton instance

Examples:

[GenAllVR.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeIdentifyFile.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

25.127.3.6 `bool gdcm::Global::LoadResourcesFiles ()`

Load all internal XML files, ressource path need to have been set before calling this member function (see Append/-Prepend members func)

Warning

not thread safe !

Examples:

[GenerateStandardSOPClasses.cxx](#).

25.127.3.7 `const char* gdcm::Global::Locate (const char * resfile) const` `[protected]`

Locate a ressource file.

25.127.3.8 `bool gdcm::Global::Prepend (const char * path)`

Prepend path at the begining of the path list

Warning

not thread safe !

25.127.4 Friends And Related Function Documentation

25.127.4.1 `std::ostream& operator<< (std::ostream & _os, const Global & g)` `[friend]`

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

- [gdcmGlobal.h](#)

25.128 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)

25.128.1 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 ?

25.128.2 Member Typedef Documentation

25.128.2.1 typedef std::vector<std::string> [gdcmm::GroupDict::GroupStringVector](#)

25.128.3 Constructor & Destructor Documentation

25.128.3.1 [gdcmm::GroupDict::GroupDict](#) () `[inline]`

25.128.3.2 [gdcmm::GroupDict::~~GroupDict](#) () `[inline]`

25.128.4 Member Function Documentation

25.128.4.1 void [gdcmm::GroupDict::Add](#) (std::string const & *abbreviation*, std::string const & *name*) `[protected]`

25.128.4.2 std::string const& [gdcmm::GroupDict::GetAbbreviation](#) (uint16_t *num*) const

Referenced by [gdcmm::operator<<\(\)](#).

25.128.4.3 std::string const& [gdcmm::GroupDict::GetName](#) (uint16_t *num*) const

Referenced by [gdcmm::operator<<\(\)](#).

25.128.4.4 void gdcm::GroupDict::Insert (uint16_t num, std::string const & abbreviation, std::string const & name)
[protected]

25.128.4.5 size_t gdcm::GroupDict::Size () const [inline]

Referenced by gdcm::operator<<().

25.128.5 Friends And Related Function Documentation

25.128.5.1 std::ostream& operator<< (std::ostream & _os, const GroupDict & _val) [friend]

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

- [gdcmGroupDict.h](#)

25.129 gdcm::IconImageFilter Class Reference

IconImageFilter This filter will extract icons from a [gdcm::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 [ExtractVeprolIconImages](#) ()

25.129.1 Detailed Description

IconImageFilter This filter will extract icons from a [gdcm::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](#).

25.129.2 Constructor & Destructor Documentation

25.129.2.1 `gdcm::IconImageFilter::IconImageFilter ()`

25.129.2.2 `gdcm::IconImageFilter::~~IconImageFilter ()`

25.129.3 Member Function Documentation

25.129.3.1 `bool gdcm::IconImageFilter::Extract ()`

Extract all Icon found in [File](#).

Examples:

[ExtractIconFromFile.cxx](#).

25.129.3.2 `void gdcm::IconImageFilter::ExtractIconImages ()` `[protected]`

25.129.3.3 `void gdcm::IconImageFilter::ExtractVeprolconImages ()` `[protected]`

25.129.3.4 `File& gdcm::IconImageFilter::GetFile ()` `[inline]`

25.129.3.5 `const File& gdcm::IconImageFilter::GetFile () const` `[inline]`

25.129.3.6 `IconImage& gdcm::IconImageFilter::GetIconImage (unsigned int i) const`

Examples:

[ExtractIconFromFile.cxx](#).

25.129.3.7 `unsigned int gdcm::IconImageFilter::GetNumberOfIconImages () const`

Retrieve extract IconImage (need to call Extract first)

Examples:

[ExtractIconFromFile.cxx](#).

25.129.3.8 `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](#)

25.130 gdcm::IconImageGenerator Class Reference

[IconImageGenerator](#) This filter will generate a valid Icon from the Pixel Data element (an instance of [gdcm::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.

25.130.1 Detailed Description

[IconImageGenerator](#) This filter will generate a valid Icon from the Pixel Data element (an instance of [gdcm::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](#).

25.130.2 Constructor & Destructor Documentation

25.130.2.1 `gdcm::IconImageGenerator::IconImageGenerator ()`

25.130.2.2 `gdcm::IconImageGenerator::~~IconImageGenerator ()`

25.130.3 Member Function Documentation

25.130.3.1 `void gdcm::IconImageGenerator::AutoPixelMinMax (bool b)`

Instead of explicitly specifying the min/max value for the rescale operation, let the internal mechanism compute the min/max of icon and rescale to best appropriate.

Examples:

[ExtractIconFromFile.cxx](#).

25.130.3.2 `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. true, false generates invalid Icon [Image](#) Sequence

25.130.3.3 `bool gdcm::IconImageGenerator::Generate ()`

Generate Icon.

Examples:

[ExtractIconFromFile.cxx](#).

25.130.3.4 `const IconImage& gdcm::IconImageGenerator::GetIconImage () const` `[inline]`

Retrieve generated Icon.

Examples:

[ExtractIconFromFile.cxx](#).

25.130.3.5 `Pixmap& gdcm::IconImageGenerator::GetPixmap ()` `[inline]`

25.130.3.6 `const Pixmap& gdcm::IconImageGenerator::GetPixmap () const` `[inline]`

25.130.3.7 `void gdcm::IconImageGenerator::SetOutputDimensions (const unsigned int dims[2])`

Set Target dimension of output Icon.

Examples:

[ExtractIconFromFile.cxx](#).

25.130.3.8 `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)`

25.130.3.9 `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.

25.130.3.10 `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](#)

25.131 gdcm::ignore_char Struct Reference

```
#include <gdcmElement.h>
```

Public Member Functions

- [ignore_char](#) (char c)

Public Attributes

- char [m_char](#)

25.131.1 Constructor & Destructor Documentation

25.131.1.1 `gdcm::ignore_char::ignore_char (char c)` `[inline]`

25.131.2 Member Data Documentation

25.131.2.1 `char gdcm::ignore_char::m_char`

Referenced by `gdcm::operator>>()`.

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

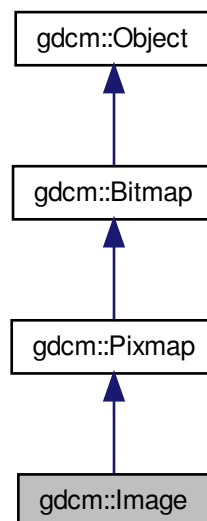
- [gdcmElement.h](#)

25.132 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`:



- `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)`

25.132.1 Detailed Description

Generated on Sat Dec 21 2013 01:40:25 for GDCM by Doxygen

- 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 [gdcm::Image](#) with [gdcm::JPEGImage](#) which would from the stream extract the header info and fill it to please [gdcm::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](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

25.132.2 Constructor & Destructor Documentation

25.132.2.1 `gdcm::Image::Image () [inline]`

25.132.2.2 `gdcm::Image::~~Image () [inline]`

25.132.3 Member Function Documentation

25.132.3.1 `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.

25.132.3.2 `double gdcm::Image::GetDirectionCosines (unsigned int idx) const`

25.132.3.3 `double gdcm::Image::GetIntercept () const [inline]`

25.132.3.4 `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](#).

25.132.3.5 double gdcm::Image::GetOrigin (unsigned int *idx*) const

25.132.3.6 double gdcm::Image::GetSlope () const [inline]

25.132.3.7 const double* gdcm::Image::GetSpacing () const

Return a 3-tuples specifying the spacing NOTE: 3rd value can be an arbitrary 1 value when the spacing was not specified (ex. 2D image). WARNING: when the spacing is not specifier, a default value of 1 will be returned

25.132.3.8 double gdcm::Image::GetSpacing (unsigned int *idx*) const

25.132.3.9 void gdcm::Image::Print (std::ostream & *os*) const [virtual]

print

Reimplemented from [gdcm::Bitmap](#).

Examples:

[CompressImage.cxx](#), and [PatchFile.cxx](#).

25.132.3.10 void gdcm::Image::SetDirectionCosines (const float * *dircos*)

25.132.3.11 void gdcm::Image::SetDirectionCosines (const double * *dircos*)

25.132.3.12 void gdcm::Image::SetDirectionCosines (unsigned int *idx*, double *dircos*)

25.132.3.13 void gdcm::Image::SetIntercept (double *intercept*) [inline]

intercept

25.132.3.14 void gdcm::Image::SetOrigin (const float * *ori*)

25.132.3.15 void gdcm::Image::SetOrigin (const double * *ori*)

25.132.3.16 void gdcm::Image::SetOrigin (unsigned int *idx*, double *ori*)

25.132.3.17 void gdcm::Image::SetSlope (double *slope*) [inline]

slope

25.132.3.18 void gdcm::Image::SetSpacing (const double * *spacing*)

Examples:

[csa2img.cxx](#), and [iU22tomultisc.cxx](#).

25.132.3.19 void `gdcm::Image::SetSpacing` (unsigned int *idx*, double *spacing*)

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

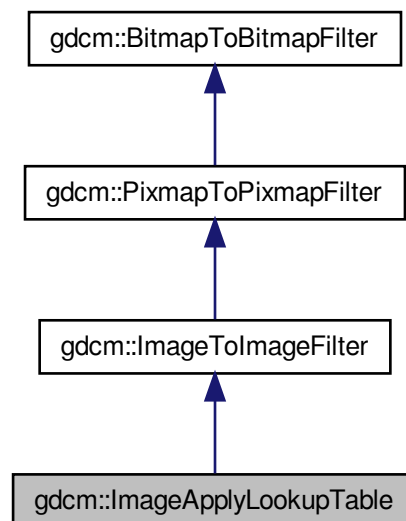
- [gdcmImage.h](#)

25.133 `gdcm::ImageApplyLookupTable` Class Reference

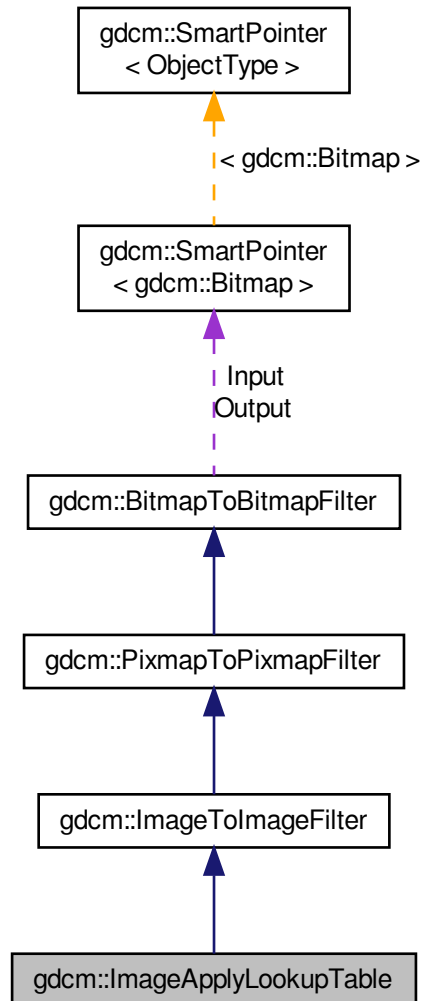
[ImageApplyLookupTable](#) class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for `gdcm::ImageApplyLookupTable`:



Collaboration diagram for gdcm::ImageApplyLookupTable:



Public Member Functions

- [ImageApplyLookupTable \(\)](#)
- [~ImageApplyLookupTable \(\)](#)
- `bool` [Apply \(\)](#)

Apply.

Additional Inherited Members

25.133.1 Detailed Description

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

25.133.2 Constructor & Destructor Documentation

25.133.2.1 `gdcm::ImageApplyLookupTable::ImageApplyLookupTable ()` `[inline]`

25.133.2.2 `gdcm::ImageApplyLookupTable::~~ImageApplyLookupTable ()` `[inline]`

25.133.3 Member Function Documentation

25.133.3.1 `bool gdcm::ImageApplyLookupTable::Apply ()`

Apply.

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

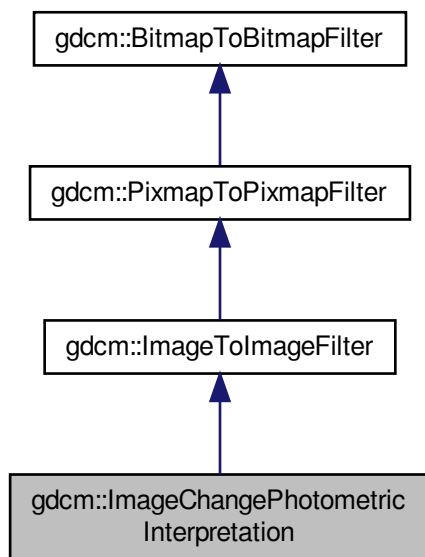
- [gdcmImageApplyLookupTable.h](#)

25.134 gdcm::ImageChangePhotometricInterpretation Class Reference

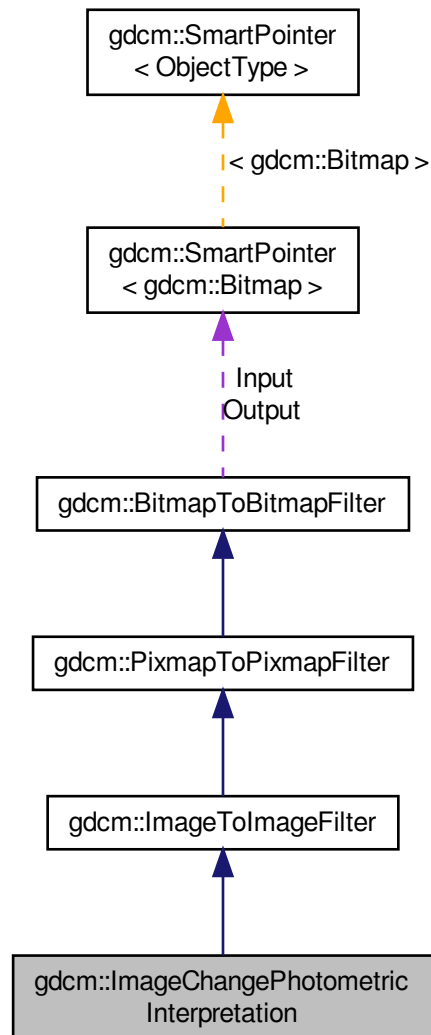
[ImageChangePhotometricInterpretation](#) class Class to change the Photometric Interpretation of an input DICOM.

```
#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 conversion (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

25.134.1 Detailed Description

[ImageChangePhotometricInterpretation](#) class Class to change the Photometric Interpretation of an input DICOM.

25.134.2 Constructor & Destructor Documentation

25.134.2.1 `gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ()` `[inline]`

25.134.2.2 `gdcm::ImageChangePhotometricInterpretation::~~ImageChangePhotometricInterpretation ()` `[inline]`

25.134.3 Member Function Documentation

25.134.3.1 `bool gdcm::ImageChangePhotometricInterpretation::Change ()`

Change.

25.134.3.2 `bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ()` `[protected]`

25.134.3.3 `const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation ()`
`const` `[inline]`

25.134.3.4 `template<typename T > void gdcm::ImageChangePhotometricInterpretation::RGB2YBR (T ybr[3], const T rgb[3])`
`[static]`

colorspace conversion (based on CCIR Recommendation 601-2)

25.134.3.5 `void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation (PhotometricInterpretation const & pi)` `[inline]`

Set/Get requested [PhotometricInterpretation](#).

25.134.3.6 `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:

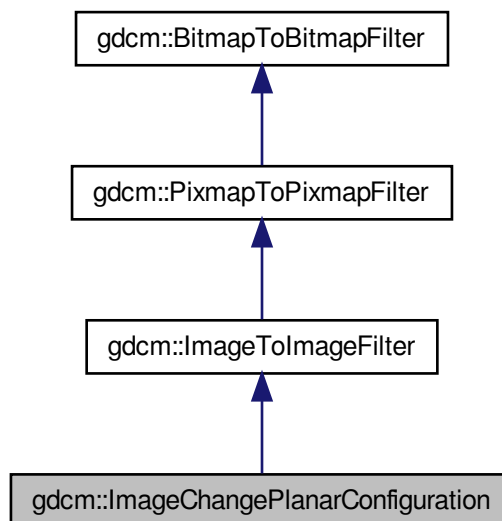
- [gdcmImageChangePhotometricInterpretation.h](#)

25.135 gdcm::ImageChangePlanarConfiguration Class Reference

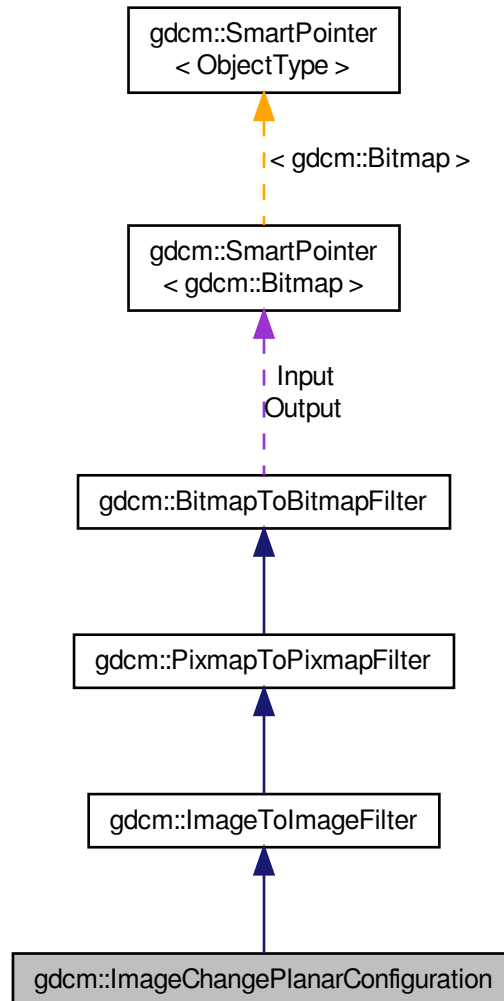
[ImageChangePlanarConfiguration](#) class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0.

```
#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 PlanarConfiguration.

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

25.135.1 Detailed Description

[ImageChangePlanarConfiguration](#) class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0.

25.135.2 Constructor & Destructor Documentation

25.135.2.1 `gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ()` `[inline]`

25.135.2.2 `gdcm::ImageChangePlanarConfiguration::~~ImageChangePlanarConfiguration ()` `[inline]`

25.135.3 Member Function Documentation

25.135.3.1 `bool gdcm::ImageChangePlanarConfiguration::Change ()`

Change.

25.135.3.2 `unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration () const` `[inline]`

25.135.3.3 `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

25.135.3.4 `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

25.135.3.5 `void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration (unsigned int pc)` `[inline]`

Set/Get requested PlanarConfiguration.

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

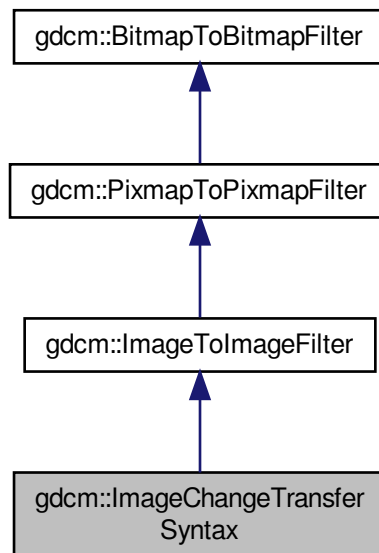
- [gdcmImageChangePlanarConfiguration.h](#)

25.136 gdcm::ImageChangeTransferSyntax Class Reference

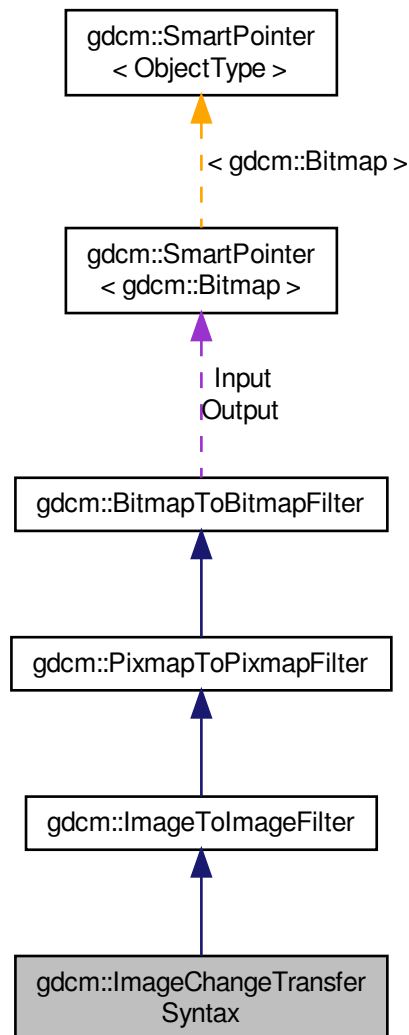
[ImageChangeTransferSyntax](#) class Class to change the transfer syntax of an input DICOM.

```
#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

25.136.1 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](#).

25.136.2 Constructor & Destructor Documentation

25.136.2.1 `gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ()` `[inline]`

25.136.2.2 `gdcm::ImageChangeTransferSyntax::~~ImageChangeTransferSyntax ()` `[inline]`

25.136.3 Member Function Documentation

25.136.3.1 `bool gdcm::ImageChangeTransferSyntax::Change ()`

Change.

Examples:

[CompressImage.cxx](#).

25.136.3.2 `const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax () const` `[inline]`

Get Transfer Syntax.

25.136.3.3 `void gdcmm::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

25.136.3.4 `void gdcmm::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 recompress using GDCM internal implementation a JPEG (for example) image

25.136.3.5 `void gdcmm::ImageChangeTransferSyntax::SetTransferSyntax (const TransferSyntax & ts)` `[inline]`

Set target Transfer Syntax.

Examples:

[CompressImage.cxx](#).

25.136.3.6 `void gdcmm::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

is 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)`

25.136.3.7 `bool gdcmm::ImageChangeTransferSyntax::TryJPEG2000Codec (const DataElement & pixelde, Bitmap const & input, Bitmap & output)` `[protected]`

25.136.3.8 `bool gdcmm::ImageChangeTransferSyntax::TryJPEGCodec (const DataElement & pixelde, Bitmap const & input, Bitmap & output)` `[protected]`

25.136.3.9 `bool gdcmm::ImageChangeTransferSyntax::TryJPEGLSCodec (const DataElement & pixelde, Bitmap const & input, Bitmap & output)` `[protected]`

25.136.3.10 `bool gdcmm::ImageChangeTransferSyntax::TryRAWCodec (const DataElement & pixelde, Bitmap const & input, Bitmap & output)` `[protected]`

25.136.3.11 `bool gdcmm::ImageChangeTransferSyntax::TryRLECodec (const DataElement & pixelde, Bitmap const & input, Bitmap & output)` `[protected]`

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

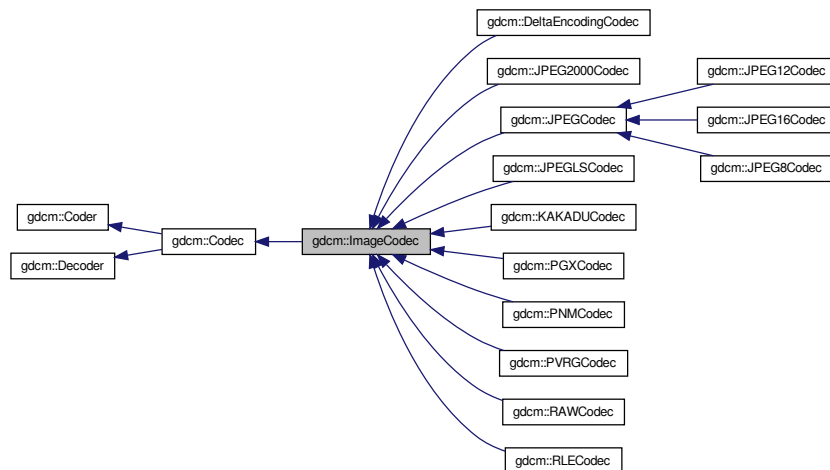
- [gdcmmImageChangeTransferSyntax.h](#)

25.137 gdcm::ImageCodec Class Reference

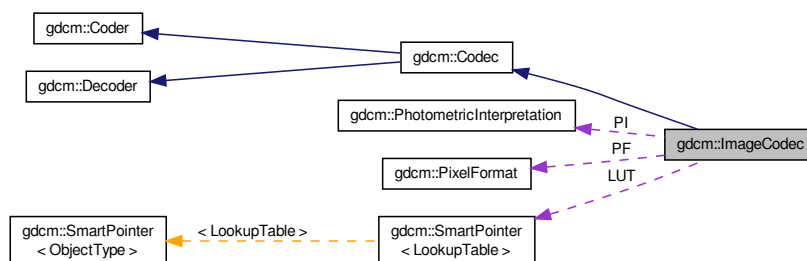
[ImageCodec](#).

```
#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)
- 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

- 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 [IsValid](#) ([PhotometricInterpretation](#) const &pi)

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 [ImageChangePhotometricInterpretation](#)

25.137.1 Detailed Description

[ImageCodec](#).

Note

Main codec, this is a central place for all implementation

25.137.2 Member Typedef Documentation

25.137.2.1 `typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr` `[protected]`

25.137.3 Constructor & Destructor Documentation

25.137.3.1 `gdcm::ImageCodec::ImageCodec ()`

25.137.3.2 `gdcm::ImageCodec::~~ImageCodec ()`

25.137.4 Member Function Documentation

25.137.4.1 `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::JPEG-LSCoec](#), [gdcm::PNMCodec](#), [gdcm::PGXCodec](#), [gdcm::KAKADUCoec](#), and [gdcm::RAWCodec](#).

25.137.4.2 `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::JPEG-LSCoec](#), [gdcm::PNMCodec](#), [gdcm::RAWCodec](#), [gdcm::PGXCodec](#), and [gdcm::KAKADUCoec](#).

25.137.4.3 `bool gdcm::ImageCodec::Decode (DataElement const & , DataElement &)` `[virtual]`

Decode.

Reimplemented from [gdcm::Decoder](#).

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::JPEGLSCoec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::KAKADUCoec](#), and [gdcm::RAWCodec](#).

25.137.4.4 `bool gdcmm::ImageCodec::DecodeByStreams (std::istream & is, std::ostream & os)` [protected],
[virtual]

Reimplemented from [gdcmm::Decoder](#).

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::RLECodec](#), [gdcmm::RAWCodec](#), [gdcmm::JPEG12Codec](#), [gdcmm::JPEG16Codec](#), and [gdcmm::JPEG8Codec](#).

25.137.4.5 `bool gdcmm::ImageCodec::DoByteSwap (std::istream & is, std::ostream & os)` [protected]

25.137.4.6 `bool gdcmm::ImageCodec::DoInvertMonochrome (std::istream & is, std::ostream & os)` [protected]

25.137.4.7 `bool gdcmm::ImageCodec::DoOverlayCleanup (std::istream & is, std::ostream & os)` [protected]

25.137.4.8 `bool gdcmm::ImageCodec::DoPaddedCompositePixelCode (std::istream & is, std::ostream & os)` [protected]

25.137.4.9 `bool gdcmm::ImageCodec::DoPlanarConfiguration (std::istream & is, std::ostream & os)` [protected]

25.137.4.10 `bool gdcmm::ImageCodec::DoSimpleCopy (std::istream & is, std::ostream & os)` [protected]

25.137.4.11 `bool gdcmm::ImageCodec::DoYBR (std::istream & is, std::ostream & os)` [protected]

25.137.4.12 `const unsigned int* gdcmm::ImageCodec::GetDimensions () const` [inline]

25.137.4.13 `virtual bool gdcmm::ImageCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::RLECodec](#), [gdcmm::JPEGLSCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::PNMCodec](#), [gdcmm::JPEG12Codec](#), [gdcmm::JPEG16Codec](#), [gdcmm::JPEG8Codec](#), [gdcmm::RAWCodec](#), and [gdcmm::PGXCodec](#).

25.137.4.14 `bool gdcmm::ImageCodec::GetLossyFlag () const`

25.137.4.15 `const LookupTable& gdcmm::ImageCodec::GetLUT () const` [inline]

25.137.4.16 `bool gdcmm::ImageCodec::GetNeedByteSwap () const` [inline]

25.137.4.17 `unsigned int gdcmm::ImageCodec::GetNumberOfDimensions () const`

25.137.4.18 `const PhotometricInterpretation& gdcmm::ImageCodec::GetPhotometricInterpretation () const`

25.137.4.19 `PixelFormat& gdcmm::ImageCodec::GetPixelFormat ()` [inline]

Examples:

[GetJPEGSamplePrecision.cxx](#).

25.137.4.20 `const PixelFormat& gdcmm::ImageCodec::GetPixelFormat () const` [inline]

25.137.4.21 `unsigned int gdcmm::ImageCodec::GetPlanarConfiguration () const` [inline]

25.137.4.22 `bool gdcmm::ImageCodec::IsLossy () const`

25.137.4.23 `virtual bool gdcm::ImageCodec::IsValid (PhotometricInterpretation const & pi)` [protected],
[virtual]

Reimplemented in [gdcm::JPEGCodec](#).

25.137.4.24 `void gdcm::ImageCodec::SetDimensions (const unsigned int d[3])`

Examples:

[ExtractIconFromFile.cxx](#).

25.137.4.25 `void gdcm::ImageCodec::SetDimensions (const std::vector< unsigned int > & d)`

25.137.4.26 `void gdcm::ImageCodec::SetLossyFlag (bool l)`

25.137.4.27 `void gdcm::ImageCodec::SetLUT (LookupTable const & lut)` [inline]

Examples:

[ExtractIconFromFile.cxx](#).

25.137.4.28 `void gdcm::ImageCodec::SetNeedByteSwap (bool b)` [inline]

25.137.4.29 `void gdcm::ImageCodec::SetNeedOverlayCleanup (bool b)` [inline]

25.137.4.30 `void gdcm::ImageCodec::SetNumberOfDimensions (unsigned int dim)`

25.137.4.31 `void gdcm::ImageCodec::SetPhotometricInterpretation (PhotometricInterpretation const & pi)`

Examples:

[ExtractIconFromFile.cxx](#).

25.137.4.32 `virtual void gdcm::ImageCodec::SetPixelFormat (PixelFormat const & pf)` [inline],[virtual]

Reimplemented in [gdcm::JPEGCodec](#).

Examples:

[ExtractIconFromFile.cxx](#).

25.137.4.33 `void gdcm::ImageCodec::SetPlanarConfiguration (unsigned int pc)` [inline]

25.137.5 Friends And Related Function Documentation

25.137.5.1 `friend class ImageChangePhotometricInterpretation` [friend]

25.137.6 Member Data Documentation

- 25.137.6.1 `unsigned int gdcm::ImageCodec::Dimensions[3]` [protected]
- 25.137.6.2 `bool gdcm::ImageCodec::LossyFlag` [protected]
- 25.137.6.3 `LUTPtr gdcm::ImageCodec::LUT` [protected]
- 25.137.6.4 `bool gdcm::ImageCodec::NeedByteSwap` [protected]
- 25.137.6.5 `bool gdcm::ImageCodec::NeedOverlayCleanup` [protected]
- 25.137.6.6 `unsigned int gdcm::ImageCodec::NumberOfDimensions` [protected]
- 25.137.6.7 `PixelFormat gdcm::ImageCodec::PF` [protected]
- 25.137.6.8 `PhotometricInterpretation gdcm::ImageCodec::PI` [protected]
- 25.137.6.9 `unsigned int gdcm::ImageCodec::PlanarConfiguration` [protected]
- 25.137.6.10 `bool gdcm::ImageCodec::RequestPaddedCompositePixelCode` [protected]
- 25.137.6.11 `bool gdcm::ImageCodec::RequestPlanarConfiguration` [protected]

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

- [gdcmImageCodec.h](#)

25.138 gdcm::ImageConverter Class Reference

[Image](#) Converter.

```
#include <gdcmImageConverter.h>
```

Public Member Functions

- [ImageConverter](#) ()
- [~ImageConverter](#) ()
- void [Convert](#) ()
- const [Image](#) & [GetOutput](#) () const
- void [SetInput](#) ([Image](#) const &input)

25.138.1 Detailed Description

[Image](#) Converter.

Note

This is the class used to convert from on [gdcm::Image](#) to another This is typically used to convert let say YBR JPEG compressed [gdcm::Image](#) to a RAW RGB [gdcm::Image](#). So that the buffer can be directly pass to third party application. This filter is application level and not integrated directly in GDCM

25.138.2 Constructor & Destructor Documentation

25.138.2.1 `gdcm::ImageConverter::ImageConverter ()`

25.138.2.2 `gdcm::ImageConverter::~~ImageConverter ()`

25.138.3 Member Function Documentation

25.138.3.1 `void gdcm::ImageConverter::Convert ()`

25.138.3.2 `const Image& gdcm::ImageConverter::GetOutput () const`

25.138.3.3 `void gdcm::ImageConverter::SetInput (Image const & input)`

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

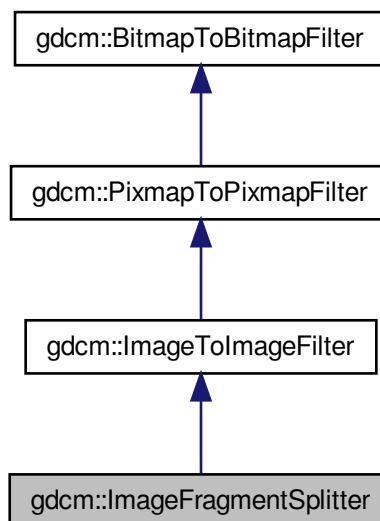
- [gdcmImageConverter.h](#)

25.139 gdcm::ImageFragmentSplitter Class Reference

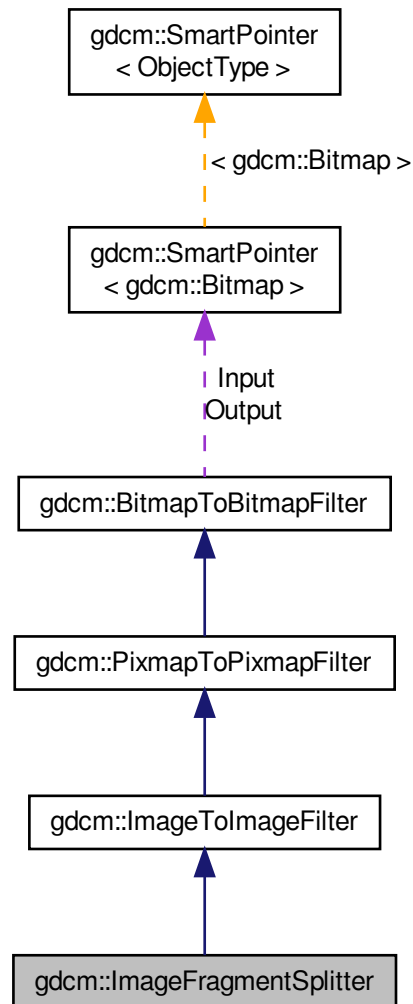
[ImageFragmentSplitter](#) class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

```
#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

25.139.1 Detailed Description

[ImageFragmentSplitter](#) class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

25.139.2 Constructor & Destructor Documentation

25.139.2.1 `gdcm::ImageFragmentSplitter::ImageFragmentSplitter ()` [\[inline\]](#)

25.139.2.2 `gdcm::ImageFragmentSplitter::~~ImageFragmentSplitter ()` [\[inline\]](#)

25.139.3 Member Function Documentation

25.139.3.1 `unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax () const` [\[inline\]](#)

25.139.3.2 `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 !

25.139.3.3 `void gdcm::ImageFragmentSplitter::SetFragmentSizeMax (unsigned int fragsize)`

FragmentSizeMax needs to be an even number.

25.139.3.4 `bool gdcm::ImageFragmentSplitter::Split ()`

Split.

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

- [gdcmImageFragmentSplitter.h](#)

25.140 gdcm::ImageHelper Class Reference

[ImageHelper](#) (internal class, not intended for user level)

```
#include <gdcmImageHelper.h>
```

Static Public Member Functions

- 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)

25.140.1 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.

25.140.2 Member Function Documentation

- 25.140.2.1 static bool [gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient](#) (const std::vector< double > & *imageposition*, std::vector< double > & *spacing*) [static]

DO NOT USE.

25.140.2.2 `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](#).

25.140.2.3 `static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet (DataSet const & ds, std::vector< double > & dircos) [static]`

25.140.2.4 `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

25.140.2.5 `static bool gdcm::ImageHelper::GetForcePixelSpacing () [static]`

25.140.2.6 `static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope () [static]`

25.140.2.7 `static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT (File const & f) [static]`

25.140.2.8 `static std::vector<double> gdcm::ImageHelper::GetOriginValue (File const & f) [static]`

Set/Get Origin (IPP) from/to a file.

25.140.2.9 `static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue (File const & f) [static]`

25.140.2.10 `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)

25.140.2.11 `static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue (const File & f) [static]`

25.140.2.12 `static const ByteValue* gdcm::ImageHelper::GetPointerFromElement (Tag const & tag, File const & f) [static]`

Moved from PixampReader to here. Generally used for photometric interpretation.

25.140.2.13 `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

25.140.2.14 `static Tag gdcmm::ImageHelper::GetSpacingTagFromMediaStorage (MediaStorage const & ms) [static], [protected]`

25.140.2.15 `static std::vector<double> gdcmm::ImageHelper::GetSpacingValue (File const & f) [static]`

Set/Get [Spacing](#) from/to a [File](#).

25.140.2.16 `static Tag gdcmm::ImageHelper::GetZSpacingTagFromMediaStorage (MediaStorage const & ms) [static], [protected]`

25.140.2.17 `static void gdcmm::ImageHelper::SetDimensionsValue (File & f, const Image & img) [static]`

25.140.2.18 `static void gdcmm::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.

25.140.2.19 `static void gdcmm::ImageHelper::SetForcePixelSpacing (bool) [static]`

GDCM 1.x compatibility issue: When using ReWrite an MR [Image](#) Storage would be rewritten as Secondary Capture [Object](#) while still having a Pixel [Spacing](#) tag (0028,0030). If you have deal with those files, use this very special flag to handle them Unless explicitly set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel [Spacing](#) of the [Image](#)

25.140.2.20 `static void gdcmm::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 explicitly set elsewhere by the standard, it will use value from 0028,1052 / 0028,1053 for the Rescale Slope & Rescale Intercept values

25.140.2.21 `static void gdcmm::ImageHelper::SetOriginValue (DataSet & ds, const Image & img) [static]`

25.140.2.22 `static void gdcmm::ImageHelper::SetRescaleInterceptSlopeValue (File & f, const Image & img) [static]`

25.140.2.23 `static void gdcmm::ImageHelper::SetSpacingValue (DataSet & ds, const std::vector< double > & spacing) [static]`

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

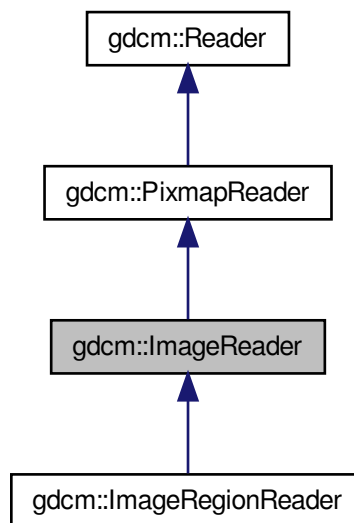
- [gdcmmImageHelper.h](#)

25.141 gdcmm::ImageReader Class Reference

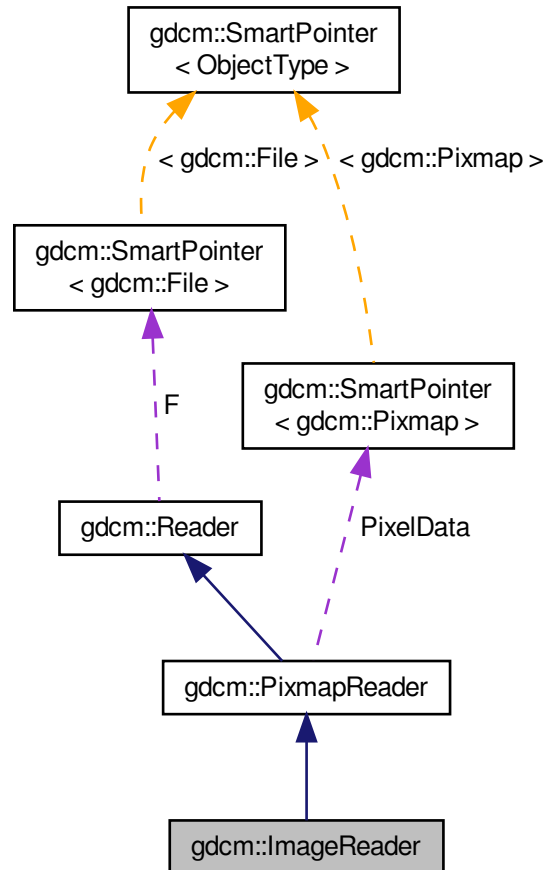
[ImageReader](#).

```
#include <gdcmmImageReader.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 ReadACRNEMAIImage ()`
- `bool ReadImage (MediaStorage const &ms)`

Additional Inherited Members

25.141.1 Detailed Description

[ImageReader](#).

Note

its role is to convert the DICOM [DataSet](#) into a [gdcm::Image](#) representation [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space.

See Also

[Image](#)

Examples:

[BasicImageAnonymizer.cs](#), [CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

25.141.2 Constructor & Destructor Documentation

25.141.2.1 `gdcm::ImageReader::ImageReader ()`

25.141.2.2 `virtual gdcm::ImageReader::~~ImageReader ()` [virtual]

25.141.3 Member Function Documentation

25.141.3.1 `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](#).

25.141.3.2 `Image& gdcm::ImageReader::GetImage ()`

25.141.3.3 `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](#).

25.141.3.4 `bool gdcm::ImageReader::ReadACRNEMAImage ()` [protected],[virtual]

Reimplemented from [gdcm::PixmapReader](#).

25.141.3.5 `bool gdcm::ImageReader::ReadImage (MediaStorage const & ms)` [protected],[virtual]

Reimplemented from [gdcm::PixmapReader](#).

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

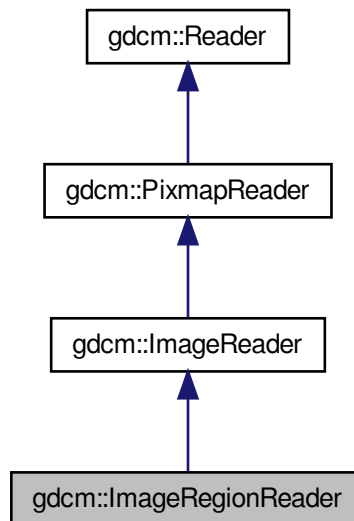
- [gdcmImageReader.h](#)

25.142 gdcm::ImageRegionReader Class Reference

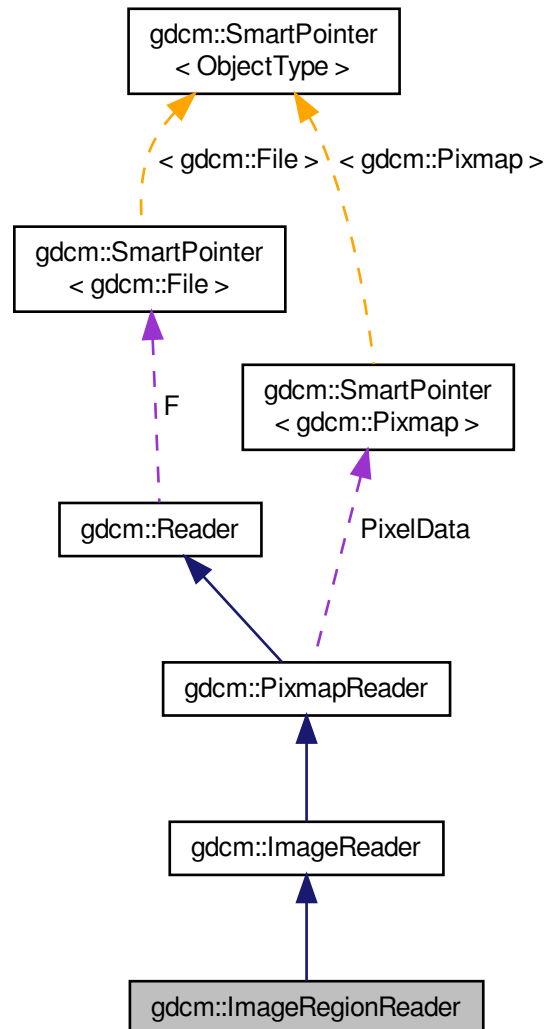
[ImageRegionReader](#).

```
#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 ®ion)

Set/Get [Region](#) to be read.

Protected Member Functions

- bool [Read](#) ()

To prevent user from calling super class [Read\(\)](#) function.

Additional Inherited Members

25.142.1 Detailed Description

[ImageRegionReader](#).

See Also

[ImageReader](#)

Examples:

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

25.142.2 Constructor & Destructor Documentation

25.142.2.1 `gdcm::ImageRegionReader::ImageRegionReader ()`

25.142.2.2 `gdcm::ImageRegionReader::~~ImageRegionReader ()`

25.142.3 Member Function Documentation

25.142.3.1 `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

25.142.3.2 `Region const& gdcm::ImageRegionReader::GetRegion () const`

25.142.3.3 `bool gdcm::ImageRegionReader::Read () [protected],[virtual]`

To prevent user from calling super class [Read\(\)](#) function.

Reimplemented from [gdcm::ImageReader](#).

25.142.3.4 `bool gdcm::ImageRegionReader::ReadInformation ()`

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

25.142.3.5 `bool gdcm::ImageRegionReader::ReadIntoBuffer (char * inreadbuffer, size_t buflen)`

Read into buffer:

Returns

false upon error

25.142.3.6 `void gdcm::ImageRegionReader::SetRegion (Region const & region)`

Set/Get [Region](#) to be read.

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

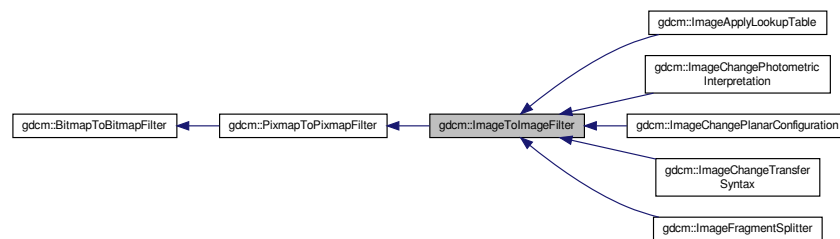
- [gdcmImageRegionReader.h](#)

25.143 gdcm::ImageToImageFilter Class Reference

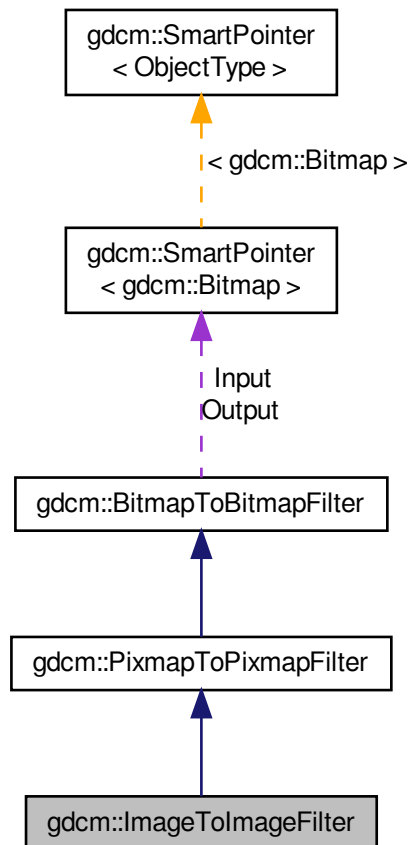
[ImageToImageFilter](#) class Super class for all filter taking an image and producing an output image.

```
#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

25.143.1 Detailed Description

[ImageToImageFilter](#) class Super class for all filter taking an image and producing an output image.

25.143.2 Constructor & Destructor Documentation

25.143.2.1 `gdcm::ImageToImageFilter::ImageToImageFilter ()`

25.143.2.2 `gdcm::ImageToImageFilter::~~ImageToImageFilter ()` `[inline]`

25.143.3 Member Function Documentation

25.143.3.1 `Image& gdcm::ImageToImageFilter::GetInput ()`

25.143.3.2 `const Image& gdcm::ImageToImageFilter::GetOutput () const`

Get Output image.

Examples:

[CompressImage.cxx](#).

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

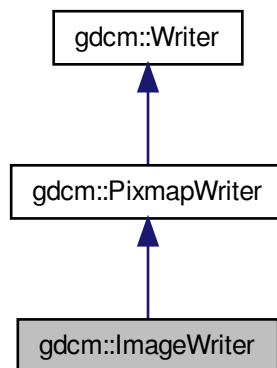
- [gdcmImageToImageFilter.h](#)

25.144 gdcm::ImageWriter Class Reference

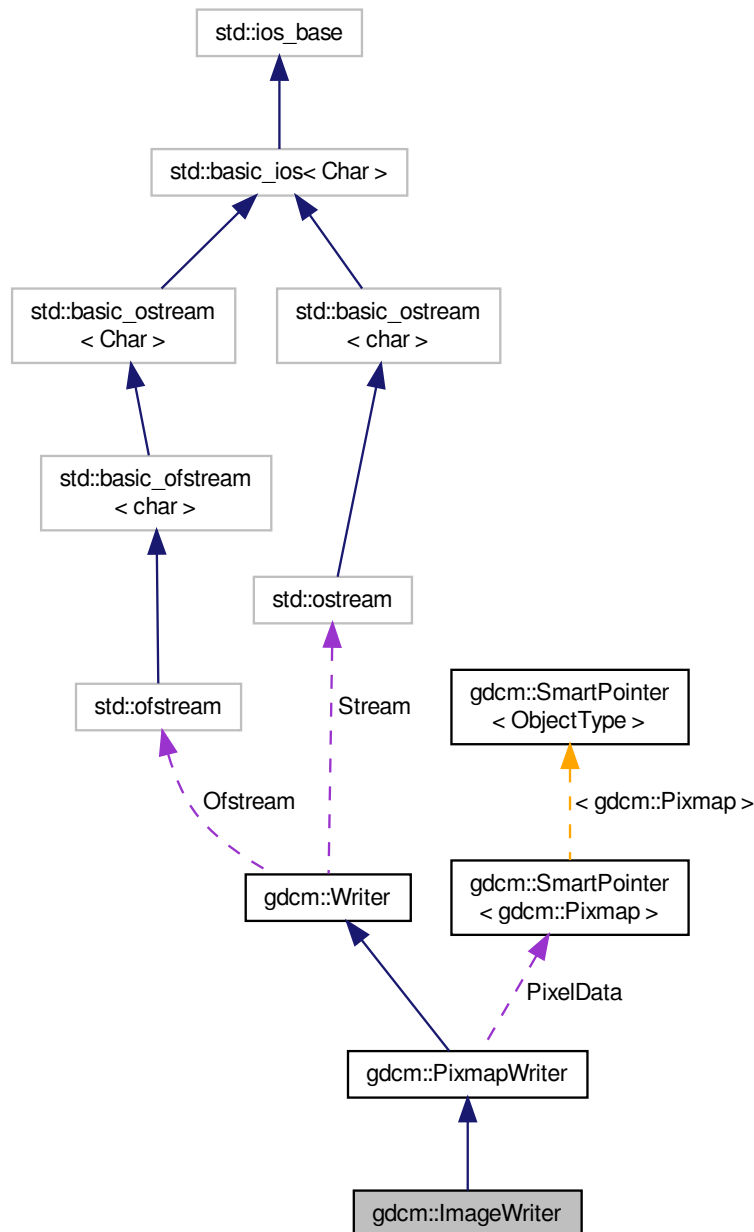
[ImageWriter](#).

```
#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

25.144.1 Detailed Description

[ImageWriter](#).

Examples:

[CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), and [MergeTwoFiles.cxx](#).

25.144.2 Constructor & Destructor Documentation

25.144.2.1 `gdcm::ImageWriter::ImageWriter ()`

25.144.2.2 `gdcm::ImageWriter::~~ImageWriter ()`

25.144.3 Member Function Documentation

25.144.3.1 `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](#).

25.144.3.2 `Image& gdcm::ImageWriter::GetImage ()` `[inline],[virtual]`

Reimplemented from [gdcm::PixmapWriter](#).

25.144.3.3 `bool gdcm::ImageWriter::Write ()` `[virtual]`

Write.

Reimplemented from [gdcm::Writer](#).

Examples:

[CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakeImage.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), and [MergeTwoFiles.cxx](#).

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

- [gdcmImageWriter.h](#)

25.145 gdcm::network::ImplementationClassUIDSub Class Reference

[ImplementationClassUIDSub](#) PS 3.7 [Table D.3-1](#) IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-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

25.145.1 Detailed Description

[ImplementationClassUIDSub](#) PS 3.7 [Table D.3-1](#) IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

25.145.2 Constructor & Destructor Documentation

25.145.2.1 `gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ()`

25.145.3 Member Function Documentation

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

25.145.3.2 `std::istream& gdcm::network::ImplementationClassUIDSub::Read (std::istream & is)`

25.145.3.3 `size_t gdcm::network::ImplementationClassUIDSub::Size () const`

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

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

- [gdcmImplementationClassUIDSub.h](#)

25.146 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

25.146.1 Detailed Description

[ImplementationUIDSub Table](#) D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

25.146.2 Constructor & Destructor Documentation

25.146.2.1 `gdcm::network::ImplementationUIDSub::ImplementationUIDSub ()`

25.146.3 Member Function Documentation

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

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

- [gdcmImplementationUIDSub.h](#)

25.147 `gdcm::network::ImplementationVersionNameSub` Class Reference

[ImplementationVersionNameSub Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-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

25.147.1 Detailed Description

[ImplementationVersionNameSub Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

25.147.2 Constructor & Destructor Documentation

25.147.2.1 `gdcm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ()`

25.147.3 Member Function Documentation

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

25.147.3.2 `std::istream& gdcm::network::ImplementationVersionNameSub::Read (std::istream & is)`

25.147.3.3 `size_t gdcm::network::ImplementationVersionNameSub::Size () const`

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

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

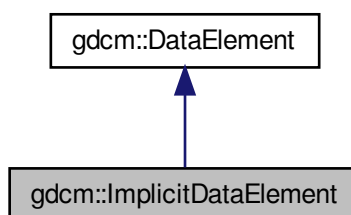
- [gdcmImplementationVersionNameSub.h](#)

25.148 gdcm::ImplicitDataElement Class Reference

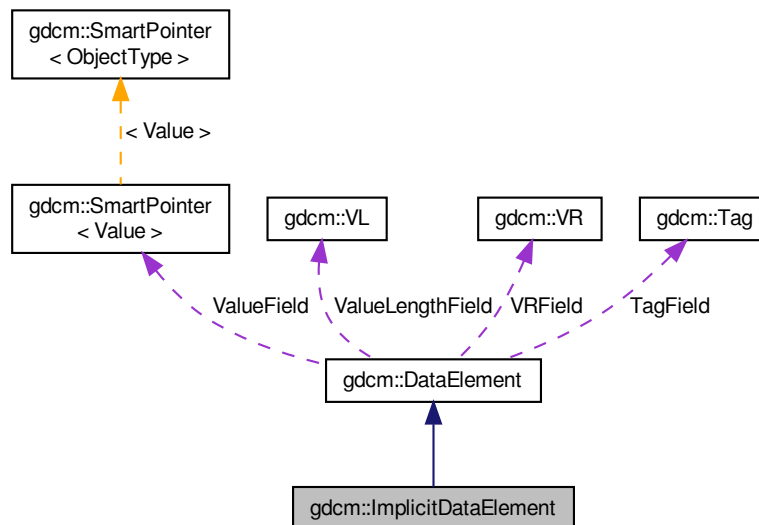
Class to represent an *Implicit VR Data Element*.

```
#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)
- 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

25.148.1 Detailed Description

Class to represent an *Implicit VR Data Element*.

Note

bla

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

25.148.2 Member Function Documentation

25.148.2.1 `VL gdcm::ImplicitDataElement::GetLength () const`

25.148.2.2 `template<typename TSwap > std::istream& gdcm::ImplicitDataElement::Read (std::istream & is)`

25.148.2.3 `template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadPreValue (std::istream & is)`

25.148.2.4 `template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValue (std::istream & is)`

25.148.2.5 `template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

25.148.2.6 `template<typename TSwap > const std::ostream& gdcm::ImplicitDataElement::Write (std::ostream & os) const`

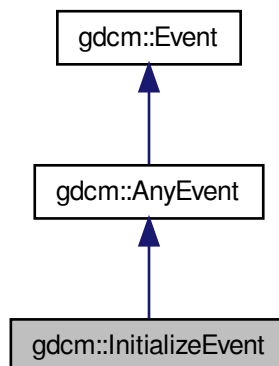
The documentation for this class was generated from the following file:

- [gdcmImplicitDataElement.h](#)

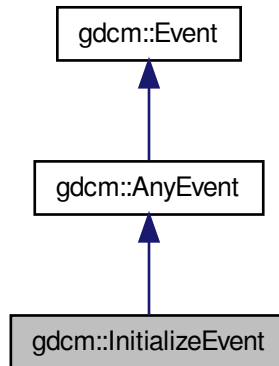
25.149 gdcm::InitializeEvent Class Reference

```
#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](#)

25.150 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)`

25.150.1 Detailed Description

Class for representing a [IOD](#).

Note

bla

See Also

[Dict](#)

Examples:

[TraverseModules.cxx](#).

25.150.2 Member Typedef Documentation

25.150.2.1 `typedef std::vector<IODEntry> gdcm::IOD::MapIODEntry`

25.150.2.2 `typedef MapIODEntry::size_type gdcm::IOD::SizeType`

25.150.3 Constructor & Destructor Documentation

25.150.3.1 `gdcm::IOD::IOD () [inline]`

25.150.4 Member Function Documentation

25.150.4.1 `void gdcm::IOD::AddIODEntry (const IODEntry & iode) [inline]`

25.150.4.2 `void gdcm::IOD::Clear () [inline]`

25.150.4.3 `const IODEntry& gdcm::IOD::GetIODEntry (SizeType idx) const [inline]`

Examples:

[TraverseModules.cxx](#).

25.150.4.4 `SizeType gdcm::IOD::GetNumberOfIODs () const [inline]`

Examples:

[TraverseModules.cxx](#).

25.150.4.5 Type `gdcm::IOD::GetTypeFromTag (const Defs & defs, const Tag & tag) const`

25.150.5 Friends And Related Function Documentation

25.150.5.1 `std::ostream& operator<< (std::ostream & _os, const IOD & _val)` [*friend*]

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

- [gdcmIOD.h](#)

25.151 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)

25.151.1 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](#).

25.151.2 Constructor & Destructor Documentation

25.151.2.1 `gdcmlIOEntry::IOEntry (const char * name = " ", const char * ref = " ", const char * usag = " ")` `[inline]`

25.151.3 Member Function Documentation

25.151.3.1 `const char* gdcmlIOEntry::GetIE ()` `const` `[inline]`

25.151.3.2 `const char* gdcmlIOEntry::GetName ()` `const` `[inline]`

25.151.3.3 `const char* gdcmlIOEntry::GetRef ()` `const` `[inline]`

Examples:

[TraverseModules.cxx](#).

25.151.3.4 `const char* gdcmlIOEntry::GetUsage ()` `const` `[inline]`

25.151.3.5 `Usage::UsageType gdcmlIOEntry::GetUsageType ()` `const`

25.151.3.6 `void gdcmlIOEntry::SetIE (const char * ie)` `[inline]`

25.151.3.7 `void gdcmlIOEntry::SetName (const char * name)` `[inline]`

25.151.3.8 `void gdcmlIOEntry::SetRef (const char * ref)` `[inline]`

25.151.3.9 `void gdcmlIOEntry::SetUsage (const char * usag)` `[inline]`

25.151.4 Friends And Related Function Documentation

25.151.4.1 `std::ostream& operator<< (std::ostream &_os, const IOEntry &_val)` `[friend]`

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

- [gdcmlIOEntry.h](#)

25.152 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)

25.152.1 Detailed Description

Class for representing a [IODs](#).

Note

bla

See Also

[IOD](#)

Examples:

[TraverseModules.cxx](#).

25.152.2 Member Typedef Documentation

25.152.2.1 typedef std::map<[IODName](#), [IOD](#)> gdcm::IODs::IODMapType

25.152.2.2 typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator

25.152.2.3 typedef std::string gdcm::IODs::IODName

25.152.3 Constructor & Destructor Documentation

25.152.3.1 `gdcm::IODs::IODs ()` `[inline]`

25.152.4 Member Function Documentation

25.152.4.1 `void gdcm::IODs::AddIOD (const char * name, const IOD & module)` `[inline]`

25.152.4.2 `IODMapTypeConstIterator gdcm::IODs::Begin () const` `[inline]`

25.152.4.3 `void gdcm::IODs::Clear ()` `[inline]`

25.152.4.4 `IODMapTypeConstIterator gdcm::IODs::End () const` `[inline]`

25.152.4.5 `const IOD& gdcm::IODs::GetIOD (const char * name) const` `[inline]`

25.152.5 Friends And Related Function Documentation

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

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

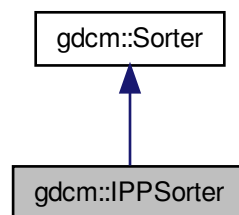
- [gdcmIODs.h](#)

25.153 gdcm::IPPSorter Class Reference

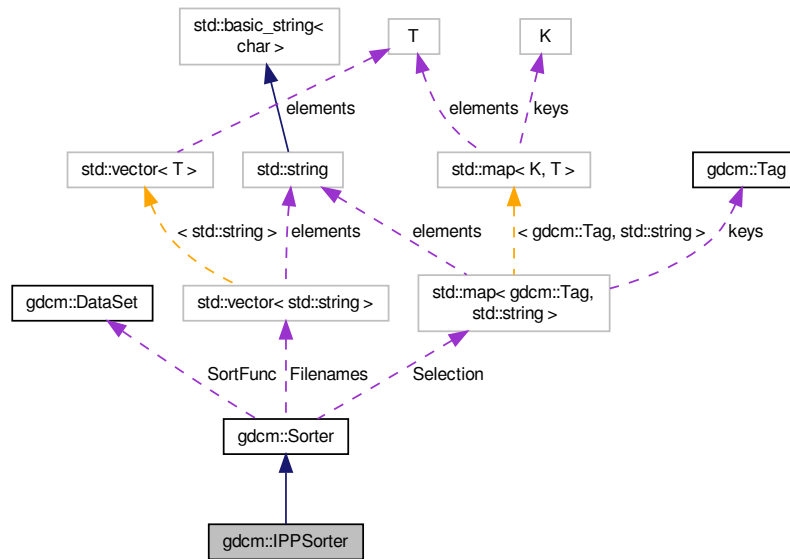
IPPSorter Implement a simple **Image** Position (**Patient**) sorter, along the **Image Orientation** (**Patient**) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

```
#include <gdcmIPPSorter.h>
```

Inheritance diagram for `gdcm::IPPSorter`:



Collaboration diagram for gdcm::IPPSorter:



Public Member Functions

- `IPPSorter ()`
- `~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

25.153.1 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 constant spacing is: 1e-6mm)

For more information on **Spacing**, and how it is defined in DICOM, advanced users may refer 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:

[gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

25.153.2 Constructor & Destructor Documentation

25.153.2.1 `gdcm::IPPSorter::IPPSorter ()`

25.153.2.2 `gdcm::IPPSorter::~~IPPSorter ()`

25.153.3 Member Function Documentation

25.153.3.1 `double gdcm::IPPSorter::GetDirectionCosinesTolerance () const` `[inline]`

25.153.3.2 `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:

[gdcmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

25.153.3.3 `double gdcm::IPPSorter::GetZSpacingTolerance () const` `[inline]`

25.153.3.4 `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 is missing then ZSpacing will be set to 0 since the spacing will not be found to be regular along the [Series](#)

Examples:

[gdcmmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

25.153.3.5 `void gdcmm::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).

25.153.3.6 `void gdcmm::IPPSorter::SetDropDuplicatePositions (bool b) [inline]`

Makes the [IPPSorter](#) ignore multiple images located at the same position. Only the first occurrence will be kept. Drop-DuplicatePositions defaults to false.

25.153.3.7 `void gdcmm::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:

[gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

25.153.3.8 `virtual bool gdcmm::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 (SetZSpacingTolerance, ...) Return value indicate if sorting could be achieved. Warning this does *NOT* imply that spacing is constant, 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 [gdcmm::Sorter](#).

Examples:

[gdcmmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

25.153.4 Member Data Documentation

25.153.4.1 `bool gdcmm::IPPSorter::ComputeZSpacing [protected]`

25.153.4.2 `double gdcmm::IPPSorter::DirCosTolerance [protected]`

25.153.4.3 `bool gdcM::IPPSorter::DropDuplicatePositions` [protected]

25.153.4.4 `double gdcM::IPPSorter::ZSpacing` [protected]

25.153.4.5 `double gdcM::IPPSorter::ZTolerance` [protected]

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

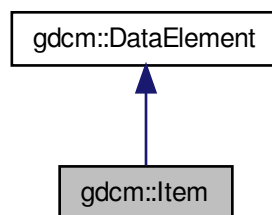
- [gdcMIPPSorter.h](#)

25.154 gdcM::Item Class Reference

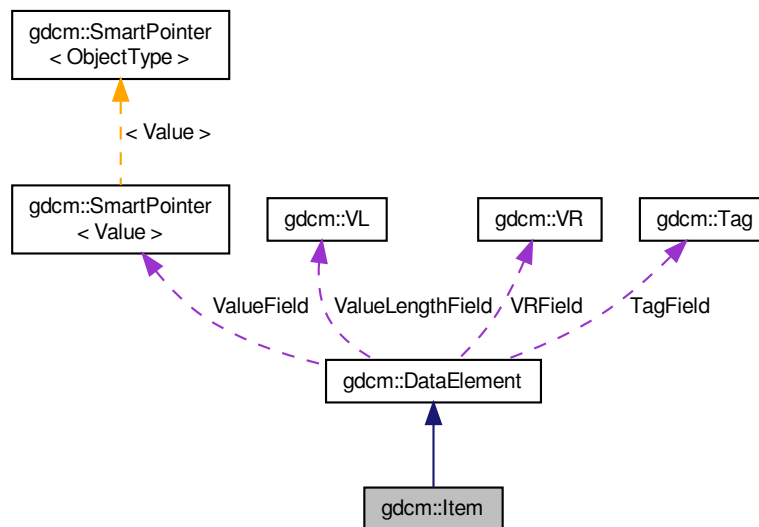
Class to represent an [Item](#) A component of the value of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set . See PS 3.5 7.5.1 [Item](#) Encoding Rules Each [Item](#) of a Data [Element](#) of [VR](#) SQ shall be encoded as a DICOM Standart 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

25.154.1 Detailed Description

Class to represent an [Item](#) A component of the value of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set . See PS 3.5 7.5.1 [Item](#) Encoding Rules Each [Item](#) of a Data [Element](#) of [VR](#) SQ shall be encoded as a DICOM Standard Data [Element](#) with a specific Data [Element](#) Tag of [Value](#) (FFFE,E000). The [Item](#) Tag is followed by a 4 byte [Item](#) Length field encoded in one of the following two ways Explicit/ Implicit.

Note

ITEM: A component of the [Value](#) of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set.

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpGEMSMovieGroup.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](#), [LargeVRDSExplicit.cxx](#), and [NewSequence.cs](#).

25.154.2 Constructor & Destructor Documentation

25.154.2.1 `gdcm::Item::Item ()` `[inline]`

25.154.2.2 `gdcm::Item::Item (Item const & val)` `[inline]`

25.154.3 Member Function Documentation

25.154.3.1 `void gdcm::Item::Clear ()` `[inline]`

References `gdcm::DataElement::Clear()`.

Referenced by `gdcm::SequenceOfItems::Read()`.

25.154.3.2 `bool gdcm::Item::FindDataElement (const Tag & t) const` `[inline]`

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

25.154.3.3 `const DataElement& gdcm::Item::GetDataElement (const Tag & t) const` `[inline]`

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

25.154.3.4 `template<typename TDE > VL gdcm::Item::GetLength () const`

25.154.3.5 `const DataSet& gdcm::Item::GetNestedDataSet () const` `[inline]`

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllVR-](#)

[R.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), and [LargeVRDSExplicit.cxx](#).

Referenced by `gdcm::SequenceOfItems::Read()`.

25.154.3.6 `DataSet& gdcm::Item::GetNestedDataSet ()` `[inline]`

25.154.3.7 `void gdcm::Item::InsertDataElement (const DataElement & de)` `[inline]`

25.154.3.8 `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()`.

25.154.3.9 `void gdcm::Item::SetNestedDataSet (const DataSet & nested)` `[inline]`

25.154.3.10 `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()`.

25.154.4 Friends And Related Function Documentation

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

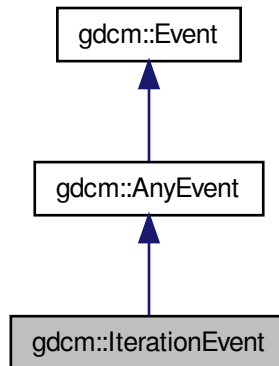
The documentation for this class was generated from the following file:

- [gdcmItem.h](#)

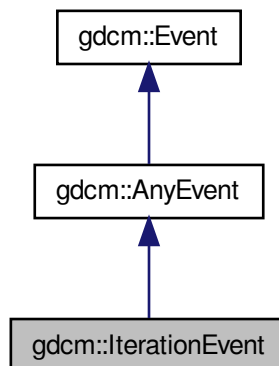
25.155 gdcm::IterationEvent Class Reference

```
#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:

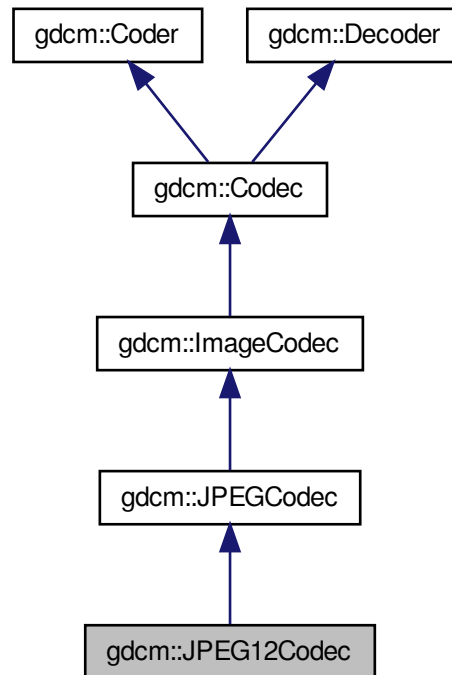
- [gdcmEvent.h](#)

25.156 gdcm::JPEG12Codec Class Reference

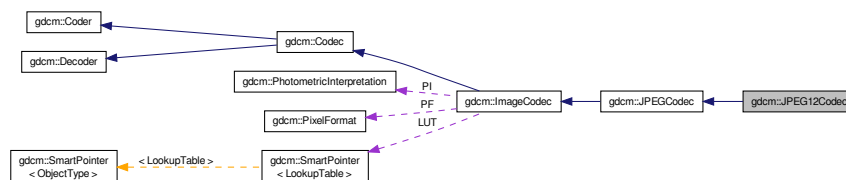
Class to do JPEG 12bits (lossy & lossless)

```
#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

- bool [IsStateSuspension](#) () const

Additional Inherited Members

25.156.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

Note

internal class

25.156.2 Constructor & Destructor Documentation

25.156.2.1 `gdcm::JPEG12Codec::JPEG12Codec ()`

25.156.2.2 `gdcm::JPEG12Codec::~~JPEG12Codec ()`

25.156.3 Member Function Documentation

25.156.3.1 `bool gdcm::JPEG12Codec::DecodeByStreams (std::istream & is, std::ostream & os)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

25.156.3.2 `bool gdcm::JPEG12Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

25.156.3.3 `bool gdcm::JPEG12Codec::InternalCode (const char * input, unsigned long len, std::ostream & os)` [virtual]

Reimplemented from [gdcm::Coder](#).

25.156.3.4 `bool gdcm::JPEG12Codec::IsStateSuspension () const` [protected],[virtual]

Reimplemented from [gdcm::JPEGCodec](#).

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

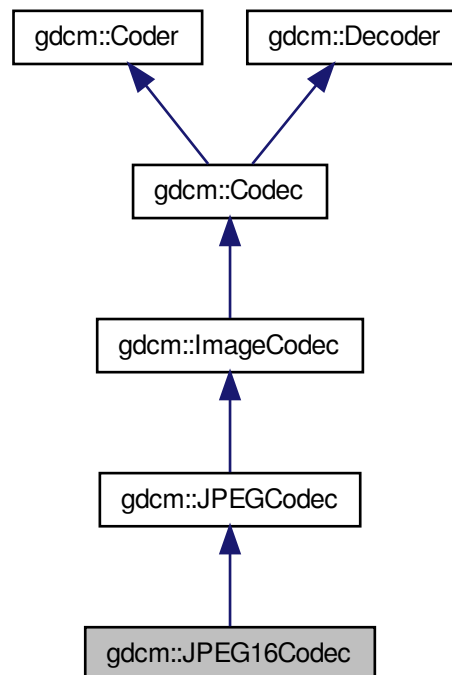
- [gdcmJPEG12Codec.h](#)

25.157 gdcm::JPEG16Codec Class Reference

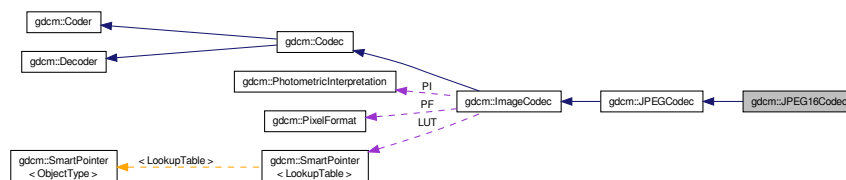
Class to do JPEG 16bits (lossless)

```
#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

- bool [IsStateSuspension](#) () const

Additional Inherited Members

25.157.1 Detailed Description

Class to do JPEG 16bits (lossless)

Note

internal class

25.157.2 Constructor & Destructor Documentation

25.157.2.1 `gdcm::JPEG16Codec::JPEG16Codec ()`

25.157.2.2 `gdcm::JPEG16Codec::~~JPEG16Codec ()`

25.157.3 Member Function Documentation

25.157.3.1 `bool gdcm::JPEG16Codec::DecodeByStreams (std::istream & is, std::ostream & os)` `[virtual]`

Reimplemented from [gdcm::ImageCodec](#).

25.157.3.2 `bool gdcm::JPEG16Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` `[virtual]`

Reimplemented from [gdcm::JPEGCodec](#).

25.157.3.3 `bool gdcm::JPEG16Codec::InternalCode (const char * input, unsigned long len, std::ostream & os)` `[virtual]`

Reimplemented from [gdcm::Coder](#).

25.157.3.4 `bool gdcm::JPEG16Codec::IsStateSuspension () const` `[protected]`, `[virtual]`

Reimplemented from [gdcm::JPEGCodec](#).

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

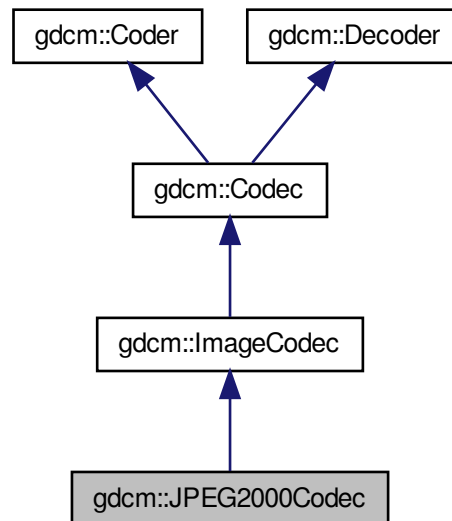
- [gdcmJPEG16Codec.h](#)

25.158 gdcm::JPEG2000Codec Class Reference

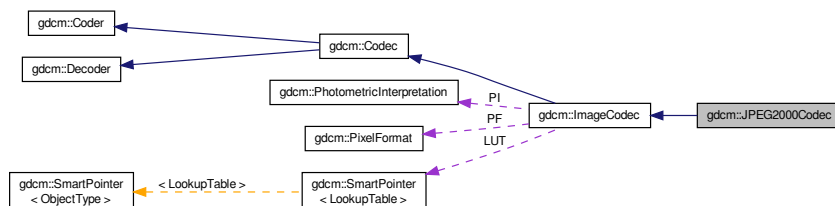
Class to do JPEG 2000.

```
#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)

- 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 [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)

Friends

- class [Bitmap](#)
- class [ImageRegionReader](#)

Additional Inherited Members

25.158.1 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](#)

25.158.2 Constructor & Destructor Documentation

25.158.2.1 `gdcm::JPEG2000Codec::JPEG2000Codec ()`

25.158.2.2 `gdcm::JPEG2000Codec::~~JPEG2000Codec ()`

25.158.3 Member Function Documentation

25.158.3.1 `bool gdcm::JPEG2000Codec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.158.3.2 `bool gdcm::JPEG2000Codec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.158.3.3 `bool gdcm::JPEG2000Codec::Code (DataElement const & in_, DataElement & out_)` [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

25.158.3.4 `bool gdcm::JPEG2000Codec::Decode (DataElement const &, DataElement &)` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

25.158.3.5 `bool gdcm::JPEG2000Codec::DecodeByStreams (std::istream & is, std::ostream & os)` [protected],
[virtual]

Reimplemented from [gdcm::ImageCodec](#).

25.158.3.6 `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]

25.158.3.7 `virtual bool gdcm::JPEG2000Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

25.158.3.8 `double gdcm::JPEG2000Codec::GetQuality (unsigned int idx = 0) const`

25.158.3.9 `double gdcm::JPEG2000Codec::GetRate (unsigned int idx = 0) const`

25.158.3.10 `void gdcm::JPEG2000Codec::SetNumberOfResolutions (unsigned int nres)`

25.158.3.11 `void gdcm::JPEG2000Codec::SetQuality (unsigned int idx, double q)`

25.158.3.12 `void gdcm::JPEG2000Codec::SetRate (unsigned int idx, double rate)`

25.158.3.13 `void gdcm::JPEG2000Codec::SetReversible (bool res)`

25.158.3.14 `void gdcm::JPEG2000Codec::SetTileSize (unsigned int tx, unsigned int ty)`

25.158.4 Friends And Related Function Documentation

25.158.4.1 `friend class Bitmap` [friend]

25.158.4.2 `friend class ImageRegionReader` [friend]

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

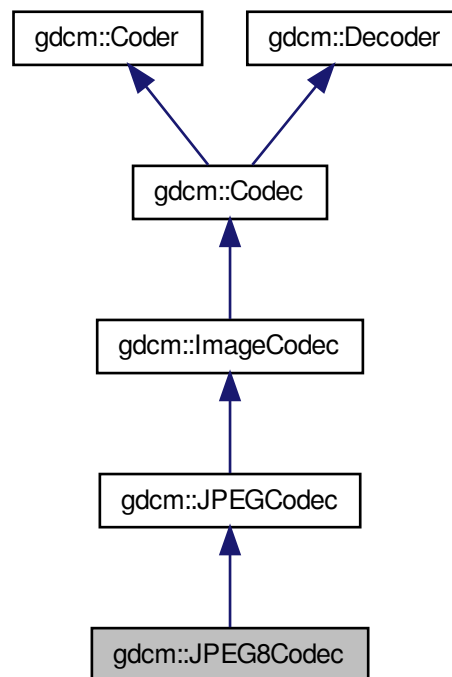
- [gdcmJPEG2000Codec.h](#)

25.159 gdcm::JPEG8Codec Class Reference

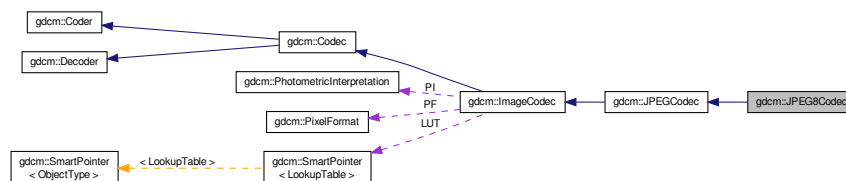
Class to do JPEG 8bits (lossy & lossless)

```
#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

- bool [IsStateSuspension](#) () const

Additional Inherited Members

25.159.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

Note

internal class

25.159.2 Constructor & Destructor Documentation

25.159.2.1 `gdcm::JPEG8Codec::JPEG8Codec ()`

25.159.2.2 `gdcm::JPEG8Codec::~~JPEG8Codec ()`

25.159.3 Member Function Documentation

25.159.3.1 `bool gdcm::JPEG8Codec::DecodeByStreams (std::istream & is, std::ostream & os)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

25.159.3.2 `bool gdcm::JPEG8Codec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

25.159.3.3 `bool gdcm::JPEG8Codec::InternalCode (const char * input, unsigned long len, std::ostream & os)` [virtual]

Reimplemented from [gdcm::Coder](#).

25.159.3.4 `bool gdcm::JPEG8Codec::IsStateSuspension () const` [protected],[virtual]

Reimplemented from [gdcm::JPEGCodec](#).

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

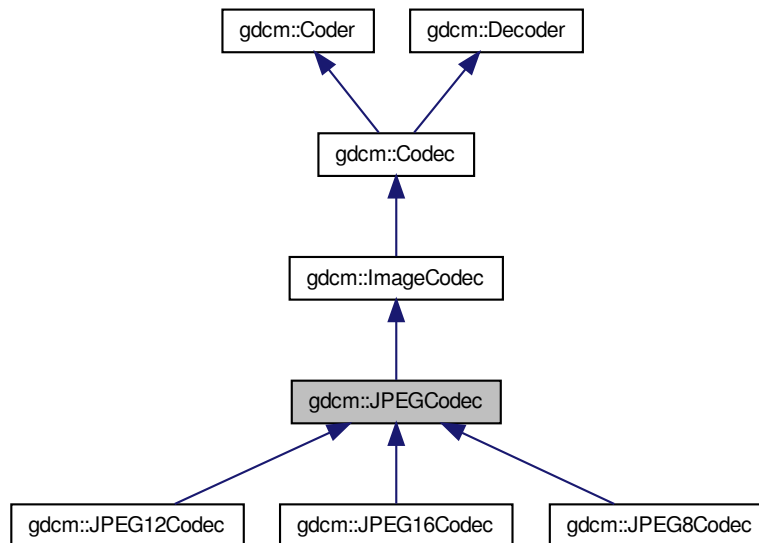
- [gdcmJPEG8Codec.h](#)

25.160 gdcm::JPEGCodec Class Reference

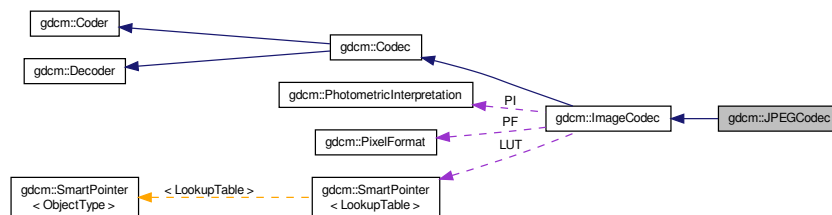
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: [gdcm::JPEG8Codec](#), [gdcm::JPEG12Codec](#) & [gdcm::JPEG16Codec](#). It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation 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)

- 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 [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 [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)
- virtual bool [IsStateSuspension](#) () const
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- void [SetBitSample](#) (int bit)

Protected Attributes

- int [BitSample](#)
- bool [Lossless](#)
- int [Quality](#)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

25.160.1 Detailed Description

JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: [gdcm::JPEG8Codec](#), [gdcm::JPEG12Codec](#) & [gdcm::JPEG16Codec](#). It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case.

Note

Things you should know if you ever want to dive into DICOM/JPEG world (among other):

- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/625e46919f208
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/75fdfccc65a62
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/2d525ef6a2f09
- http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c9

Examples:

[GetJPEGSamplePrecision.cxx](#).

25.160.2 Constructor & Destructor Documentation

25.160.2.1 `gdcm::JPEGCodec::JPEGCodec ()`

25.160.2.2 `gdcm::JPEGCodec::~~JPEGCodec ()`

25.160.3 Member Function Documentation

25.160.3.1 `bool gdcm::JPEGCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.160.3.2 `bool gdcm::JPEGCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.160.3.3 `bool gdcm::JPEGCodec::Code (DataElement const & in, DataElement & out)` [virtual]

Compress into JPEG.

Reimplemented from [gdcm::Coder](#).

25.160.3.4 `void gdcm::JPEGCodec::ComputeOffsetTable (bool b)`

Compute the offset table:

25.160.3.5 `bool gdcm::JPEGCodec::Decode (DataElement const & , DataElement &)` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

25.160.3.6 `bool gdcm::JPEGCodec::DecodeByStreams (std::istream & is, std::ostream & os)` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

25.160.3.7 `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]

25.160.3.8 `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](#).

25.160.3.9 `bool gdcm::JPEGCodec::GetLossless ()` const

25.160.3.10 `double gdcm::JPEGCodec::GetQuality ()` const

25.160.3.11 `virtual bool gdcm::JPEGCodec::IsStateSuspension ()` const [protected],[virtual]

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

25.160.3.12 `bool gdcm::JPEGCodec::IsValid (PhotometricInterpretation const & pi)` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

25.160.3.13 `void gdcm::JPEGCodec::SetBitSample (int bit)` [protected]

25.160.3.14 `void gdcm::JPEGCodec::SetLossless (bool l)`

25.160.3.15 `void gdcm::JPEGCodec::SetPixelFormat (PixelFormat const & pf)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

Examples:

[GetJPEGSamplePrecision.cxx](#).

25.160.3.16 `void gdcm::JPEGCodec::SetQuality (double q)`

25.160.4 Friends And Related Function Documentation

25.160.4.1 `friend class ImageRegionReader` [friend]

25.160.5 Member Data Documentation

25.160.5.1 `int gdcM::JPEGCodec::BitSample` [protected]

25.160.5.2 `bool gdcM::JPEGCodec::Lossless` [protected]

25.160.5.3 `int gdcM::JPEGCodec::Quality` [protected]

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

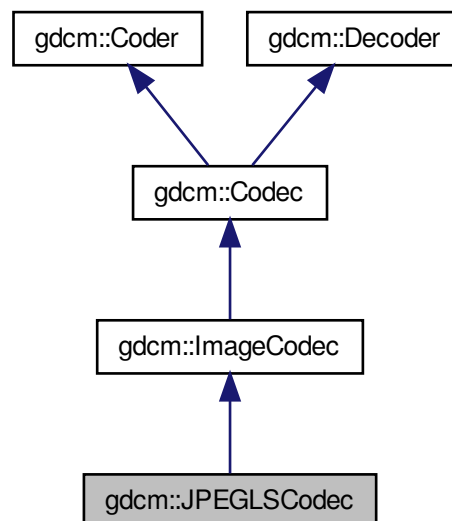
- [gdcMJPEGCodec.h](#)

25.161 gdcM::JPEGLSCodec Class Reference

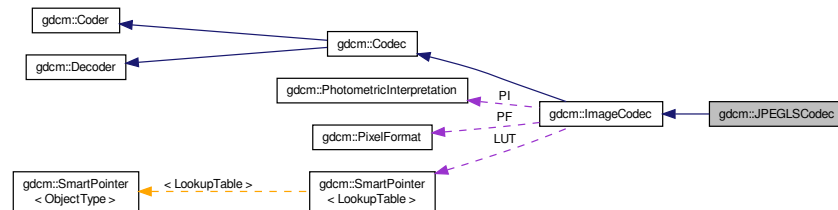
JPEG-LS.

```
#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)
- 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 [DecodeExtent](#) (char *buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

25.161.1 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>

25.161.2 Constructor & Destructor Documentation

25.161.2.1 `gdcm::JPEGLSCodec::JPEGLSCodec ()`

25.161.2.2 `gdcm::JPEGLSCodec::~~JPEGLSCodec ()`

25.161.3 Member Function Documentation

25.161.3.1 `bool gdcm::JPEGLSCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.161.3.2 `bool gdcm::JPEGLSCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.161.3.3 `bool gdcm::JPEGLSCodec::Code (DataElement const & in_, DataElement & out_)` [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

25.161.3.4 `bool gdcm::JPEGLSCodec::Decode (DataElement const &, DataElement &)` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

25.161.3.5 `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)`

25.161.3.6 `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]

25.161.3.7 `unsigned long gdcm::JPEGLSCodec::GetBufferLength () const` [inline]

25.161.3.8 `bool gdcm::JPEGLSCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

25.161.3.9 `bool gdcm::JPEGLSCodec::GetLossless ()` const

25.161.3.10 `void gdcm::JPEGLSCodec::SetBufferLength (unsigned long l)` [inline]

25.161.3.11 `void gdcm::JPEGLSCodec::SetLossless (bool l)`

25.161.3.12 `void gdcm::JPEGLSCodec::SetLossyError (int error)`

[0-3] generally

25.161.4 Friends And Related Function Documentation

25.161.4.1 `friend class ImageRegionReader` [friend]

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

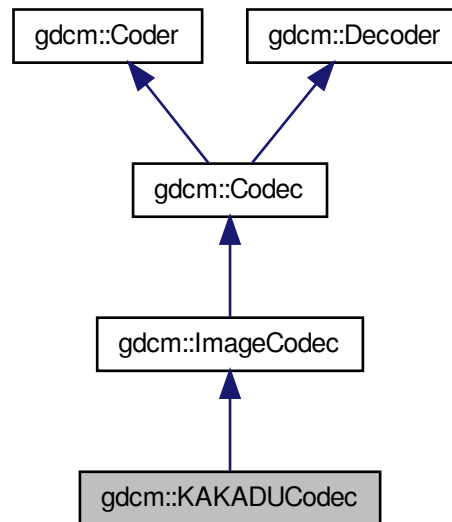
- [gdcmJPEGLSCodec.h](#)

25.162 gdcm::KAKADUCodec Class Reference

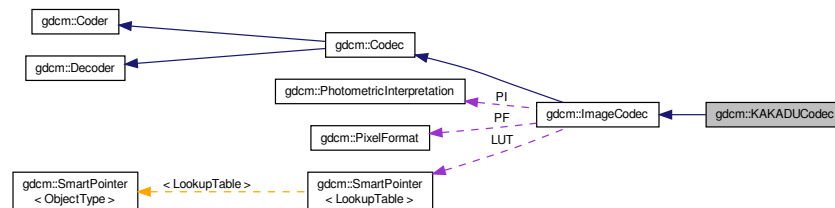
[KAKADUCodec](#).

```
#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)
- `bool Code (DataElement const &in, DataElement &out)`
Code.
- `bool Decode (DataElement const &is, DataElement &os)`
Decode.

Additional Inherited Members

25.162.1 Detailed Description

[KAKADUCodec](#).

25.162.2 Constructor & Destructor Documentation

25.162.2.1 `gdcm::KAKADUCodec::KAKADUCodec ()`

25.162.2.2 `gdcm::KAKADUCodec::~~KAKADUCodec ()`

25.162.3 Member Function Documentation

25.162.3.1 `bool gdcm::KAKADUCodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.162.3.2 `bool gdcm::KAKADUCodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.162.3.3 `bool gdcm::KAKADUCodec::Code (DataElement const & in_, DataElement & out_)` `[virtual]`

Code.

Reimplemented from [gdcm::Coder](#).

25.162.3.4 `bool gdcm::KAKADUCodec::Decode (DataElement const & , DataElement &)` `[virtual]`

Decode.

Reimplemented from [gdcm::ImageCodec](#).

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

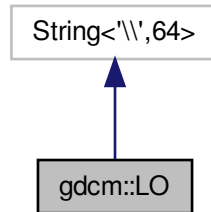
- [gdcmKAKADUCodec.h](#)

25.163 gdcm::LO Class Reference

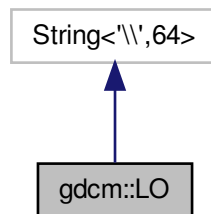
[LO](#).

```
#include <gdcmLO.h>
```

Inheritance diagram for `gdc::LO`:



Collaboration diagram for `gdc::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=npow)
- bool [IsValid](#) () const

25.163.1 Detailed Description

[LO](#).

Note

TODO

25.163.2 Member Typedef Documentation

25.163.2.1 `typedef Superclass::const_iterator gdcm::LO::const_iterator`

25.163.2.2 `typedef Superclass::const_reference gdcm::LO::const_reference`

25.163.2.3 `typedef Superclass::const_reverse_iterator gdcm::LO::const_reverse_iterator`

25.163.2.4 `typedef Superclass::difference_type gdcm::LO::difference_type`

25.163.2.5 `typedef Superclass::iterator gdcm::LO::iterator`

25.163.2.6 `typedef Superclass::pointer gdcm::LO::pointer`

25.163.2.7 `typedef Superclass::reference gdcm::LO::reference`

25.163.2.8 `typedef Superclass::reverse_iterator gdcm::LO::reverse_iterator`

25.163.2.9 `typedef Superclass::size_type gdcm::LO::size_type`

25.163.2.10 `typedef String<'\\',64> gdcm::LO::Superclass`

25.163.2.11 `typedef Superclass::value_type gdcm::LO::value_type`

25.163.3 Constructor & Destructor Documentation

25.163.3.1 `gdcm::LO::LO () [inline]`

25.163.3.2 `gdcm::LO::LO (const value_type * s) [inline]`

25.163.3.3 `gdcm::LO::LO (const value_type * s, size_type n) [inline]`

25.163.3.4 `gdcm::LO::LO (const Superclass & s, size_type pos = 0, size_type n = npow) [inline]`

25.163.4 Member Function Documentation

25.163.4.1 `bool gdcM::LO::IsValid () const [inline]`

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

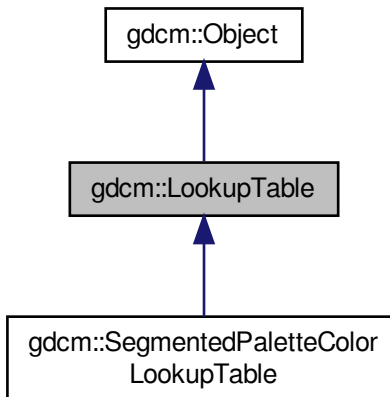
- [gdcMLO.h](#)

25.164 gdcM::LookupTable Class Reference

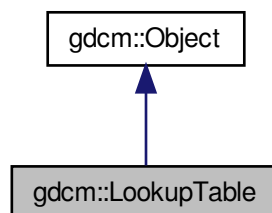
[LookupTable](#) class.

```
#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

25.164.1 Detailed Description

[LookupTable](#) class.

Examples:

[ExtractImageRegionWithLUT.cs](#), and [ScanDirectory.java](#).

25.164.2 Member Enumeration Documentation

25.164.2.1 enum gdcm::LookupTable::LookupTableType

Enumerator

RED

GREEN

BLUE

GRAY

UNKNOWN

25.164.3 Constructor & Destructor Documentation

25.164.3.1 gdcm::LookupTable::LookupTable ()

25.164.3.2 gdcm::LookupTable::~~LookupTable ()

25.164.3.3 gdcm::LookupTable::LookupTable ([LookupTable](#) const & *lut*) `[inline]`

25.164.4 Member Function Documentation

25.164.4.1 void gdcm::LookupTable::Allocate (unsigned short *bitsample* = 8)

Allocate the LUT.

25.164.4.2 void gdcm::LookupTable::Clear ()

Clear the LUT.

25.164.4.3 void gdcm::LookupTable::Decode (std::istream & *is*, std::ostream & *os*) const

Decode the LUT.

25.164.4.4 `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

25.164.4.5 `unsigned short gdcm::LookupTable::GetBitSample () const [inline]`

return the bit sample

25.164.4.6 `bool gdcm::LookupTable::GetBufferAsRGBA (unsigned char * rgba) const`

return the LUT as RGBA buffer

25.164.4.7 `void gdcm::LookupTable::GetLUT (LookupTableType type, unsigned char * array, unsigned int & length) const`

25.164.4.8 `void gdcm::LookupTable::GetLUTDescriptor (LookupTableType type, unsigned short & length, unsigned short & subscript, unsigned short & bitsize) const`

25.164.4.9 `unsigned int gdcm::LookupTable::GetLUTLength (LookupTableType type) const`

25.164.4.10 `const unsigned char* gdcm::LookupTable::GetPointer () const`

return a raw pointer to the LUT

25.164.4.11 `void gdcm::LookupTable::InitializeBlueLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)`

25.164.4.12 `bool gdcm::LookupTable::Initialized () const`

return whether the LUT has been initialized

25.164.4.13 `void gdcm::LookupTable::InitializeGreenLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)`

25.164.4.14 `void gdcm::LookupTable::InitializeLUT (LookupTableType type, unsigned short length, unsigned short subscript, unsigned short bitsize)`

Generic interface:

25.164.4.15 `void gdcm::LookupTable::InitializeRedLUT (unsigned short length, unsigned short subscript, unsigned short bitsize)`

RED / GREEN / BLUE specific:

25.164.4.16 `void gdcm::LookupTable::Print (std::ostream &) const [inline],[virtual]`

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

25.164.4.17 void `gdcm::LookupTable::SetBlueLUT` (const unsigned char * *blue*, unsigned int *length*)

25.164.4.18 void `gdcm::LookupTable::SetGreenLUT` (const unsigned char * *green*, unsigned int *length*)

25.164.4.19 virtual void `gdcm::LookupTable::SetLUT` (`LookupTableType` *type*, const unsigned char * *array*, unsigned int *length*) [virtual]

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

25.164.4.20 void `gdcm::LookupTable::SetRedLUT` (const unsigned char * *red*, unsigned int *length*)

25.164.4.21 bool `gdcm::LookupTable::WriteBufferAsRGBA` (const unsigned char * *rgba*)

Write the LUT as RGBA.

25.164.5 Member Data Documentation

25.164.5.1 unsigned short `gdcm::LookupTable::BitSample` [protected]

25.164.5.2 bool `gdcm::LookupTable::IncompleteLUT` [protected]

25.164.5.3 `LookupTableInternal*` `gdcm::LookupTable::Internal` [protected]

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

- [gdcmLookupTable.h](#)

25.165 gdcm::Scanner::Itstr Struct Reference

```
#include <gdcmScanner.h>
```

Public Member Functions

- bool [operator\(\)](#) (const char **s1*, const char **s2*) const

25.165.1 Member Function Documentation

25.165.1.1 bool `gdcm::Scanner::Itstr::operator()` (const char * *s1*, const char * *s2*) const [inline]

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

- [gdcmScanner.h](#)

25.166 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](#) directly at root-level. See [Macro](#) for nested-included level.
- void [Clear](#) ()
- bool [FindMacroEntry](#) (const [Tag](#) &tag) const
- const [MacroEntry](#) & [GetMacroEntry](#) (const [Tag](#) &tag) const
- const char * [GetName](#) () const
- void [SetName](#) (const char *name)
- bool [Verify](#) (const [DataSet](#) &ds, [Usage](#) const &usage) const

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Macro](#) &_val)

25.166.1 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](#)

25.166.2 Member Typedef Documentation

25.166.2.1 typedef std::vector<std::string> [gdcm::Macro::ArrayIncludeMacrosType](#)

25.166.2.2 typedef std::map<[Tag](#), [MacroEntry](#)> [gdcm::Macro::MapModuleEntry](#)

25.166.3 Constructor & Destructor Documentation

25.166.3.1 [gdcm::Macro::Macro](#) () `[inline]`

25.166.4 Member Function Documentation

25.166.4.1 void [gdcm::Macro::AddMacroEntry](#) (const [Tag](#) & *tag*, const [MacroEntry](#) & *module*) `[inline]`

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

25.166.4.2 void `gdcmmacro::Macro::Clear ()` `[inline]`

25.166.4.3 bool `gdcmmacro::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.

25.166.4.4 const `MacroEntry&` `gdcmmacro::Macro::GetMacroEntry (const Tag & tag)` const

25.166.4.5 const char* `gdcmmacro::Macro::GetName ()` const `[inline]`

25.166.4.6 void `gdcmmacro::Macro::SetName (const char * name)` `[inline]`

25.166.4.7 bool `gdcmmacro::Macro::Verify (const DataSet & ds, Usage const & usage)` const

25.166.5 Friends And Related Function Documentation

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

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

- [gdcmmacro.h](#)

25.167 gdcmmacro::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)

25.167.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See Also

[Module](#)

Examples:

[TraverseModules.cxx](#).

25.167.2 Member Typedef Documentation

25.167.2.1 `typedef std::map<std::string, Macro> gdcm::Macros::ModuleMapType`

25.167.3 Constructor & Destructor Documentation

25.167.3.1 `gdcm::Macros::Macros ()` `[inline]`

25.167.4 Member Function Documentation

25.167.4.1 `void gdcm::Macros::AddMacro (const char * ref, const Macro & module)` `[inline]`

25.167.4.2 `void gdcm::Macros::Clear ()` `[inline]`

25.167.4.3 `const Macro& gdcm::Macros::GetMacro (const char * name) const` `[inline]`

25.167.4.4 `bool gdcm::Macros::IsEmpty () const` `[inline]`

25.167.5 Friends And Related Function Documentation

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

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

- [gdcmMacros.h](#)

25.168 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

25.168.1 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)

25.168.2 Constructor & Destructor Documentation

25.168.2.1 `gdcn::network::MaximumLengthSub::MaximumLengthSub ()`

25.168.3 Member Function Documentation

25.168.3.1 `uint32_t gdcn::network::MaximumLengthSub::GetMaximumLength () const` `[inline]`

25.168.3.2 `void gdcn::network::MaximumLengthSub::Print (std::ostream & os) const`

25.168.3.3 `std::istream& gdcn::network::MaximumLengthSub::Read (std::istream & is)`

25.168.3.4 `void gdcn::network::MaximumLengthSub::SetMaximumLength (uint32_t maximumlength)`

25.168.3.5 `size_t gdcn::network::MaximumLengthSub::Size () const`

25.168.3.6 `const std::ostream& gdcn::network::MaximumLengthSub::Write (std::ostream & os) const`

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

- [gdcnMaximumLengthSub.h](#)

25.169 gdcn::MD5 Class Reference

Class for [MD5](#).

```
#include <gdcnMD5.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])

25.169.1 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

25.169.2 Constructor & Destructor Documentation

25.169.2.1 `gdcm::MD5::MD5 ()`

25.169.2.2 `gdcm::MD5::~~MD5 ()`

25.169.3 Member Function Documentation

25.169.3.1 `static bool gdcm::MD5::Compute (const char * buffer, unsigned long buf_len, char digest_str[33])` `[static]`

25.169.3.2 `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](#)

25.170 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,`
 - `XRayRadiofluoroscopicImageStorage,`
 - `XRayAngiographicBiPlaneImageStorageRetired,`
 - `NuclearMedicineImageStorage,`
 - `RawDataStorage,`
 - `SpacialRegistrationStorage,`
 - `SpacialFiducialsStorage,`
 - `PETImageStorage,`
 - `RTImageStorage,`
 - `RTDoseStorage,`
 - `RTStructureSetStorage,`
 - `RTPlanStorage,`
 - `CSANonImageStorage,`
 - `Philips3D,`
 - `EnhancedSR,`
 - `BasicTextSR,`
 - `HardcopyGrayscaleImageStorage,`
 - `ComprehensiveSR,`
 - `DetachedStudyManagementSOPClass,`
 - `EncapsulatedPDFStorage,`
 - `EncapsulatedCDASStorage,`
 - `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)

25.170.1 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](#).

25.170.2 Member Enumeration Documentation

25.170.2.1 enum gdcm::MediaStorage::MSType

Enumerator

MediaStorageDirectoryStorage
ComputedRadiographylImageStorage
DigitalXRayImageStorageForPresentation
DigitalXRayImageStorageForProcessing
DigitalMammographylImageStorageForPresentation
DigitalMammographylImageStorageForProcessing
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
XRayRadiofluoroscopicImageStorage
XRayAngiographicBiPlaneImageStorageRetired
NuclearMedicineImageStorage
RawDataStorage
SpacialRegistrationStorage
SpacialFiducialsStorage
PETImageStorage
RTImageStorage
RTDoseStorage
RTStructureSetStorage
RTPlanStorage
CSANonImageStorage
Philips3D
EnhancedSR
BasicTextSR
HardcopyGrayscaleImageStorage
ComprehensiveSR
DetachedStudyManagementSOPClass
EncapsulatedPDFStorage
EncapsulatedCDASStorage
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
FujiPrivateCRIImageStorage
OphthalmicPhotography8BitImageStorage
OphthalmicTomographyImageStorage
MS_END

Examples:

[GenerateStandardSOPClasses.cxx](#).

25.170.2.2 enum gdcm::MediaStorage::ObjectType

Enumerator

NoObject
Video
Waveform
Audio
PDF
URI
Segmentation
ObjectEnd

25.170.3 Constructor & Destructor Documentation

25.170.3.1 `gdcm::MediaStorage::MediaStorage (MStype type = MS_END) [inline]`

25.170.4 Member Function Documentation

25.170.4.1 `const char* gdcm::MediaStorage::GetModality () const`

25.170.4.2 `unsigned int gdcm::MediaStorage::GetModalityDimension () const`

25.170.4.3 `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<<()`.

25.170.4.4 `static MStype gdcm::MediaStorage::GetMStype (const char * str) [static]`

Examples:

[TestReader.cxx](#).

25.170.4.5 `static unsigned int gdcm::MediaStorage::GetNumberOfModality () [static]`

25.170.4.6 `static unsigned int gdcm::MediaStorage::GetNumberOfMSString () [static]`

25.170.4.7 `static unsigned int gdcm::MediaStorage::GetNumberOfMStype () [static]`

25.170.4.8 `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](#).

25.170.4.9 `void gdcm::MediaStorage::GuessFromModality (const char * modality, unsigned int dimension = 2)`

25.170.4.10 `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

25.170.4.11 `bool gdcM::MediaStorage::IsUndefined () const [inline]`

Examples:

[TestReader.cxx](#).

25.170.4.12 `gdcM::MediaStorage::operator MType () const [inline]`

25.170.4.13 `bool gdcM::MediaStorage::SetFromDataSet (DataSet const & ds)`

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

25.170.4.14 `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](#).

25.170.4.15 `bool gdcM::MediaStorage::SetFromHeader (FileMetaInformation const & fmi)`

25.170.4.16 `bool gdcM::MediaStorage::SetFromModality (DataSet const & ds)`

25.170.4.17 `void gdcM::MediaStorage::SetFromSourceImageSequence (DataSet const & ds) [protected]`

25.170.5 Friends And Related Function Documentation

25.170.5.1 `std::ostream& operator<< (std::ostream & os, const MediaStorage & ms) [friend]`

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

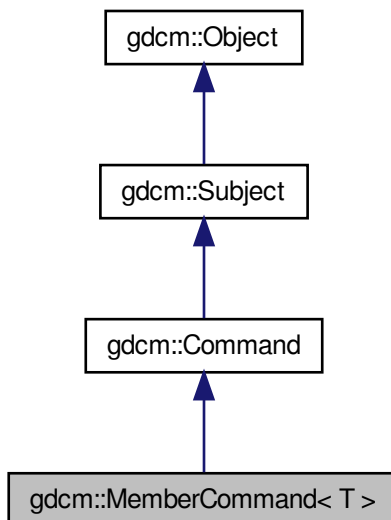
- [gdcMMediaStorage.h](#)

25.171 gdcM::MemberCommand< T > Class Template Reference

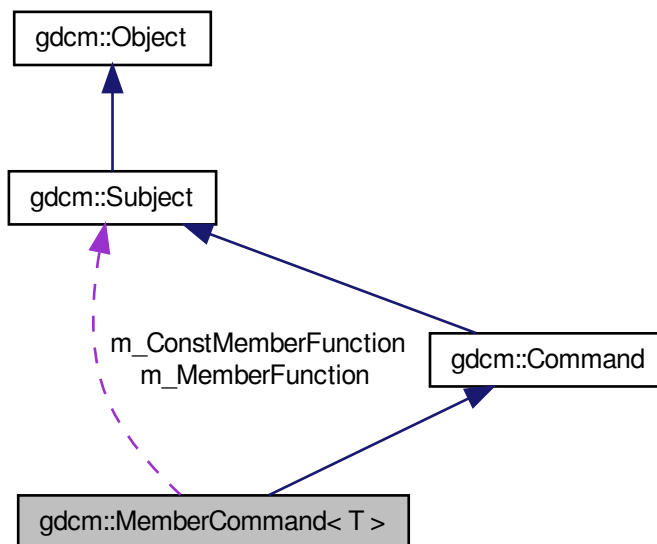
[Command](#) subclass that calls a pointer to a member function.

```
#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](#)

25.171.1 Detailed Description

```
template<class T>class gdcmmembercommand< 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](#).

25.171.2 Member Typedef Documentation

25.171.2.1 `template<class T > typedef MemberCommand gdcmmembercommand< T >::Self`

Standard class typedefs.

25.171.2.2 `template<class T> typedef void(T::* gdcmmembercommand< T >::TConstMemberFunctionPointer)(const Subject *, const Event &)`

25.171.2.3 `template<class T> typedef void(T::* gdcmmembercommand< T >::TMemberFunctionPointer)(Subject *, const Event &)`

pointer to a member function that takes a Subject* and the event

25.171.3 Constructor & Destructor Documentation

25.171.3.1 `template<class T> gdcmmembercommand< T >::MemberCommand () [inline], [protected]`

Referenced by gdcmmembercommand< T >::New().

25.171.3.2 `template<class T> virtual gdcmmembercommand< T >::~MemberCommand () [inline], [protected], [virtual]`

25.171.4 Member Function Documentation

25.171.4.1 `template<class T> virtual void gdcmmembercommand< T >::Execute (Subject * caller, const Event & event) [inline], [virtual]`

Invoke the member function.

Implements [gdcmmembercommand::Command](#).

References gdcmmembercommand< T >::m_MemberFunction.

25.171.4.2 `template<class T> virtual void gdcmmembercommand< T >::Execute (const Subject * caller, const Event & event) [inline], [virtual]`

Invoke the member function with a const object.

Implements [gdcmmembercommand::Command](#).

References gdcmmembercommand< T >::m_ConstMemberFunction.

25.171.4.3 `template<class T> static SmartPointer<MemberCommand> gdcmmembercommand< T >::New () [inline], [static]`

Method for creation through the object factory.

References gdcmmembercommand< T >::MemberCommand().

25.171.4.4 `template<class T> void gdcmmembercommand< 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 gdcmmembercommand< T >::m_MemberFunction, and gdcmmembercommand< T >::m_This.

25.171.4.5 `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`.

25.171.5 Member Data Documentation

25.171.5.1 `template<class T> TConstMemberFunctionPointer gdcm::MemberCommand< T >::m_ConstMemberFunction [protected]`

Referenced by `gdcm::MemberCommand< T >::Execute()`, and `gdcm::MemberCommand< T >::SetCallbackFunction()`.

25.171.5.2 `template<class T> TMemberFunctionPointer gdcm::MemberCommand< T >::m_MemberFunction [protected]`

Referenced by `gdcm::MemberCommand< T >::Execute()`, and `gdcm::MemberCommand< T >::SetCallbackFunction()`.

25.171.5.3 `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:

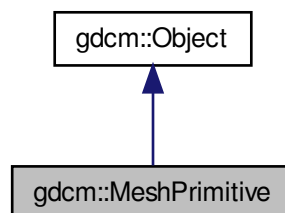
- [gdcmCommand.h](#)

25.172 gdcm::MeshPrimitive Class Reference

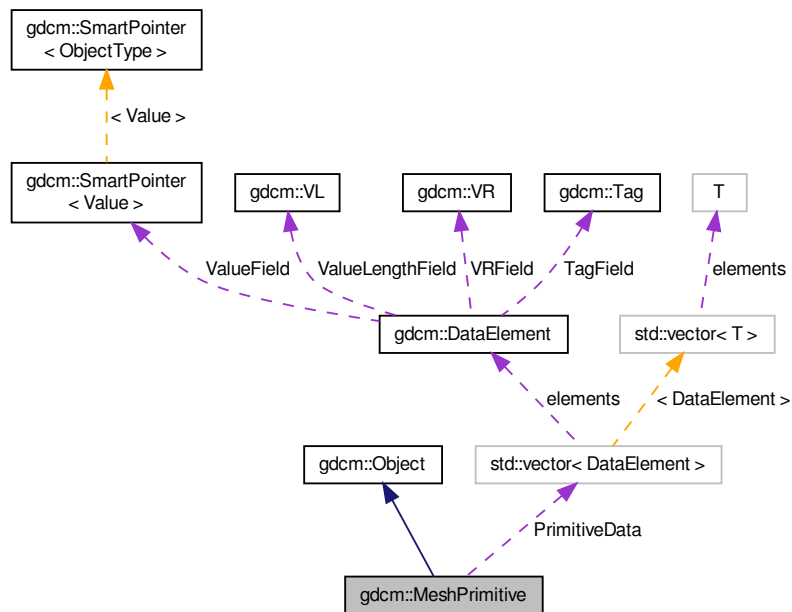
This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

```
#include <gdcmMeshPrimitive.h>
```

Inheritance diagram for `gdcm::MeshPrimitive`:



Collaboration diagram for gdcm::MeshPrimitive:



Public Types

- enum `MPTType` {
`VERTEX` = 0,
`EDGE`,
`TRIANGLE`,
`TRIANGLE_STRIP`,
`TRIANGLE_FAN`,
`LINE`,
`FACET`,
`MPTType_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](#) ()
- [MPTType](#) [GetPrimitiveType](#) () const
- void [SetPrimitiveData](#) ([DataElement](#) const &de)
- void [SetPrimitiveData](#) (const unsigned int idx, [DataElement](#) const &de)
- void [SetPrimitivesData](#) ([PrimitivesData](#) const &DEs)
- void [SetPrimitiveType](#) (const [MPTType](#) type)

Static Public Member Functions

- static [MPTType](#) [GetMPTType](#) (const char *type)
- static const char * [GetMPTTypeString](#) (const [MPTType](#) type)

Protected Attributes

- [PrimitivesData](#) [PrimitiveData](#)
- [MPTType](#) [PrimitiveType](#)

Additional Inherited Members

25.172.1 Detailed Description

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

See Also

PS 3.3 C.27.4

25.172.2 Member Typedef Documentation

25.172.2.1 `typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData`

25.172.3 Member Enumeration Documentation

25.172.3.1 `enum gdcm::MeshPrimitive::MPTType`

This enumeration defines primitive types.

See Also

PS 3.3 C.27.4.1

Enumerator

VERTEX
EDGE
TRIANGLE
TRIANGLE_STRIP
TRIANGLE_FAN
LINE
FACET
MPTType_END

25.172.4 Constructor & Destructor Documentation

25.172.4.1 `gdcm::MeshPrimitive::MeshPrimitive ()`

25.172.4.2 `virtual gdcm::MeshPrimitive::~~MeshPrimitive ()` `[virtual]`

25.172.5 Member Function Documentation

25.172.5.1 `void gdcm::MeshPrimitive::AddPrimitiveData (DataElement const & de)`

25.172.5.2 `static MPTYPE gdcm::MeshPrimitive::GetMPTYPE (const char * type)` `[static]`

25.172.5.3 `static const char* gdcm::MeshPrimitive::GetMPTYPEString (const MPTYPE type)` `[static]`

25.172.5.4 `unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData () const`

25.172.5.5 `const DataElement& gdcm::MeshPrimitive::GetPrimitiveData () const`

25.172.5.6 `DataElement& gdcm::MeshPrimitive::GetPrimitiveData ()`

25.172.5.7 `const DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int idx) const`

25.172.5.8 `DataElement& gdcm::MeshPrimitive::GetPrimitiveData (const unsigned int idx)`

25.172.5.9 `const PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData () const`

25.172.5.10 `PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ()`

25.172.5.11 `MPTYPE gdcm::MeshPrimitive::GetPrimitiveType () const`

25.172.5.12 `void gdcm::MeshPrimitive::SetPrimitiveData (DataElement const & de)`

25.172.5.13 `void gdcm::MeshPrimitive::SetPrimitiveData (const unsigned int idx, DataElement const & de)`

25.172.5.14 `void gdcm::MeshPrimitive::SetPrimitivesData (PrimitivesData const & DEs)`

25.172.5.15 `void gdcm::MeshPrimitive::SetPrimitiveType (const MPTYPE type)`

25.172.6 Member Data Documentation

25.172.6.1 `PrimitivesData gdcm::MeshPrimitive::PrimitiveData` `[protected]`

25.172.6.2 `MPTYPE gdcm::MeshPrimitive::PrimitiveType` `[protected]`

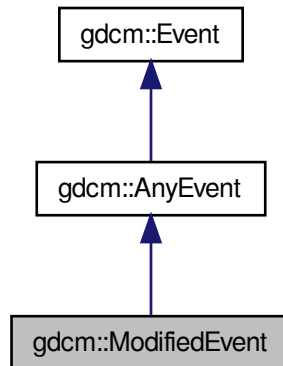
The documentation for this class was generated from the following file:

- [gdcmMeshPrimitive.h](#)

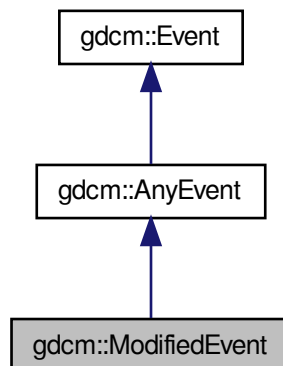
25.173 gdcm::ModifiedEvent Class Reference

```
#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](#)

25.174 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](#) directly at root-level. See [Macro](#) for nested-included level.
- void [Clear](#) ()
- bool [FindModuleEntryInMacros](#) ([Macros](#) const ¯os, const [Tag](#) &tag) const
- const [ModuleEntry](#) & [GetModuleEntryInMacros](#) ([Macros](#) const ¯os, 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)

25.174.1 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](#).

25.174.2 Member Typedef Documentation

25.174.2.1 `typedef std::vector<std::string> gdcmmodule::ArrayIncludeMacrosType`

25.174.2.2 `typedef std::map<Tag, ModuleEntry> gdcmmodule::MapModuleEntry`

25.174.3 Constructor & Destructor Documentation

25.174.3.1 `gdcmmodule::Module () [inline]`

25.174.4 Member Function Documentation

25.174.4.1 `void gdcmmodule::AddMacro (const char * include) [inline]`

25.174.4.2 `void gdcmmodule::AddModuleEntry (const Tag & tag, const ModuleEntry & module) [inline]`

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

25.174.4.3 `void gdcmmodule::Clear () [inline]`

25.174.4.4 `bool gdcmmodule::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](#).

25.174.4.5 `const ModuleEntry& gdcmmodule::GetModuleEntryInMacros (Macros const & macros, const Tag & tag) const`

Examples:

[TraverseModules.cxx](#).

25.174.4.6 `const char* gdcmmodule::GetName () const [inline]`

25.174.4.7 `void gdcmmodule::SetName (const char * name) [inline]`

25.174.4.8 `bool gdcmmodule::Verify (const DataSet & ds, Usage const & usage) const`

25.174.5 Friends And Related Function Documentation

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

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

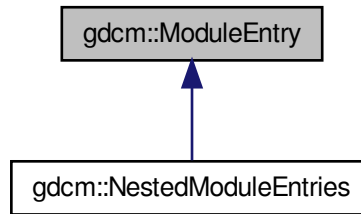
- [gdcmmodule.h](#)

25.175 gdcm::ModuleEntry Class Reference

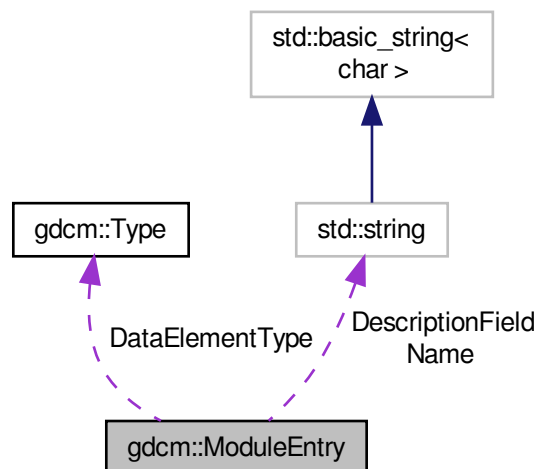
Class for representing a [ModuleEntry](#).

```
#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)

25.175.1 Detailed Description

Class for representing a [ModuleEntry](#).

Note

bla

See Also

[DictEntry](#)

Examples:

[TraverseModules.cxx](#).

25.175.2 Member Typedef Documentation

25.175.2.1 typedef std::string gdcm::ModuleEntry::Description

25.175.3 Constructor & Destructor Documentation

25.175.3.1 gdcm::ModuleEntry::ModuleEntry (const char * *name* = " ", const char * *type* = "3", const char * *description* = " ") [inline]

References [gdcm::Type::GetTypeType\(\)](#).

25.175.3.2 `virtual gdcm::ModuleEntry::~~ModuleEntry () [inline],[virtual]`

25.175.4 Member Function Documentation

25.175.4.1 `const Description& gdcm::ModuleEntry::GetDescription () const [inline]`

25.175.4.2 `const char* gdcm::ModuleEntry::GetName () const [inline]`

25.175.4.3 `const Type& gdcm::ModuleEntry::GetType () const [inline]`

Examples:

[TraverseModules.cxx](#).

25.175.4.4 `void gdcm::ModuleEntry::SetDescription (const char * d) [inline]`

25.175.4.5 `void gdcm::ModuleEntry::SetName (const char * name) [inline]`

25.175.4.6 `void gdcm::ModuleEntry::SetType (const Type & type) [inline]`

25.175.5 Friends And Related Function Documentation

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

25.175.6 Member Data Documentation

25.175.6.1 `Type gdcm::ModuleEntry::DataElementType [protected]`

Referenced by `gdcm::operator<<()`.

25.175.6.2 `Description gdcm::ModuleEntry::DescriptionField [protected]`

Referenced by `gdcm::operator<<()`.

25.175.6.3 `std::string gdcm::ModuleEntry::Name [protected]`

Referenced by `gdcm::operator<<()`.

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

- [gdcmModuleEntry.h](#)

25.176 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)

25.176.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See Also

[Module](#)

Examples:

[TraverseModules.cxx](#).

25.176.2 Member Typedef Documentation

25.176.2.1 typedef std::map<std::string, [Module](#)> [gdcm::Modules::ModuleMapType](#)

25.176.3 Constructor & Destructor Documentation

25.176.3.1 [gdcm::Modules::Modules](#) () [\[inline\]](#)

25.176.4 Member Function Documentation

25.176.4.1 void [gdcm::Modules::AddModule](#) (const char * *ref*, const [Module](#) & *module*) [\[inline\]](#)

25.176.4.2 void [gdcm::Modules::Clear](#) () [\[inline\]](#)

25.176.4.3 const [Module](#)& [gdcm::Modules::GetModule](#) (const char * *name*) const [\[inline\]](#)

25.176.4.4 bool [gdcm::Modules::IsEmpty](#) () const [\[inline\]](#)

25.176.5 Friends And Related Function Documentation

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

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

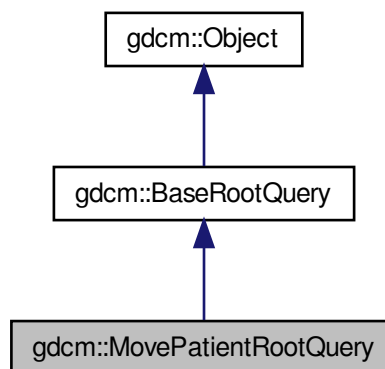
- [gdcmModules.h](#)

25.177 gdcm::MovePatientRootQuery Class Reference

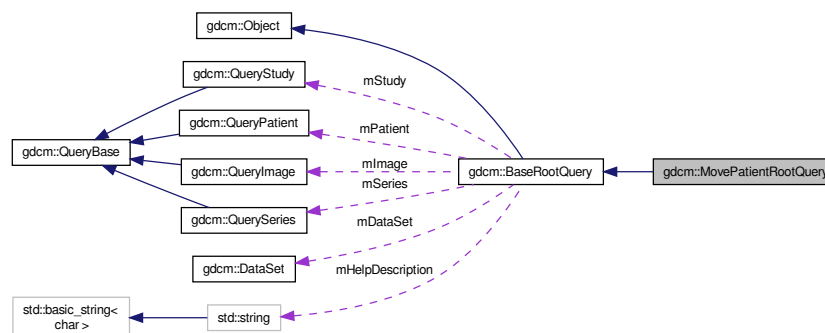
[MovePatientRootQuery](#) contains: the class which will produce a dataset for c-move with patient root.

```
#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

25.177.1 Detailed Description

[MovePatientRootQuery](#) contains: the class which will produce a dataset for c-move with patient root.

25.177.2 Constructor & Destructor Documentation

25.177.2.1 `gdcm::MovePatientRootQuery::MovePatientRootQuery ()`

25.177.3 Member Function Documentation

25.177.3.1 `UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID () const` [virtual]

Implements [gdcm::BaseRootQuery](#).

25.177.3.2 `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](#).

25.177.3.3 `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](#).

25.177.3.4 `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](#).

25.177.4 Friends And Related Function Documentation

25.177.4.1 friend class QueryFactory [friend]

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

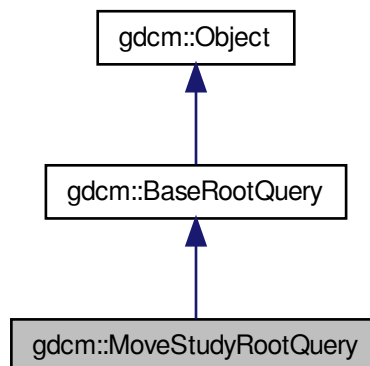
- [gdcmMovePatientRootQuery.h](#)

25.178 gdcm::MoveStudyRootQuery Class Reference

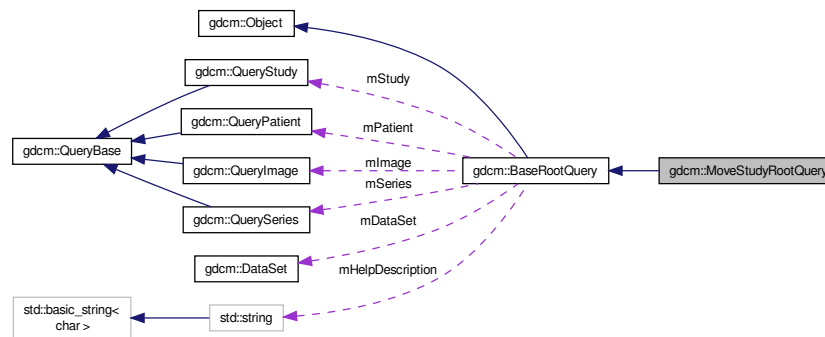
[MoveStudyRootQuery](#) contains: the class which will produce a dataset for C-MOVE with study root.

```
#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

25.178.1 Detailed Description

[MoveStudyRootQuery](#) contains: the class which will produce a dataset for C-MOVE with study root.

25.178.2 Constructor & Destructor Documentation

25.178.2.1 `gdcm::MoveStudyRootQuery::MoveStudyRootQuery ()`

25.178.3 Member Function Documentation

25.178.3.1 `UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID () const` `[virtual]`

Implements [gdcm::BaseRootQuery](#).

25.178.3.2 `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](#).

25.178.3.3 `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](#).

25.178.3.4 `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](#).

25.178.4 Friends And Related Function Documentation

25.178.4.1 `friend class QueryFactory` `[friend]`

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

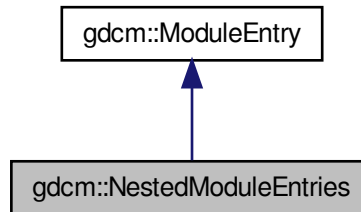
- [gdcmMoveStudyRootQuery.h](#)

25.179 gdcm::NestedModuleEntries Class Reference

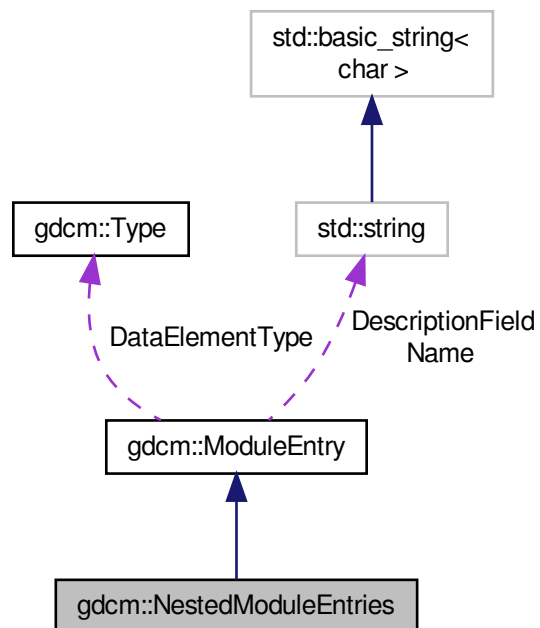
Class for representing a [NestedModuleEntries](#).

```
#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

25.179.1 Detailed Description

Class for representing a [NestedModuleEntries](#).

Note

bla

See Also

[ModuleEntry](#)

25.179.2 Member Typedef Documentation

25.179.2.1 typedef std::vector<[ModuleEntry](#)>::size_type gdcm::NestedModuleEntries::SizeType

25.179.3 Constructor & Destructor Documentation

25.179.3.1 gdcm::NestedModuleEntries::NestedModuleEntries (const char * *name* = " ", const char * *type* = "3", const char * *description* = " ") [\[inline\]](#)

25.179.4 Member Function Documentation

25.179.4.1 void gdcm::NestedModuleEntries::AddModuleEntry (const [ModuleEntry](#) & *me*) [\[inline\]](#)

25.179.4.2 const [ModuleEntry](#)& gdcm::NestedModuleEntries::GetModuleEntry ([SizeType](#) *idx*) const [\[inline\]](#)

25.179.4.3 [ModuleEntry](#)& gdcm::NestedModuleEntries::GetModuleEntry ([SizeType](#) *idx*) [\[inline\]](#)

25.179.4.4 [SizeType](#) gdcm::NestedModuleEntries::GetNumberOfModuleEntries () [\[inline\]](#)

25.179.5 Friends And Related Function Documentation

25.179.5.1 std::ostream& [operator<<](#) (std::ostream &_os, const [NestedModuleEntries](#) &_val) [\[friend\]](#)

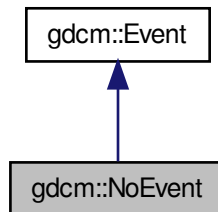
The documentation for this class was generated from the following file:

- [gdcmNestedModuleEntries.h](#)

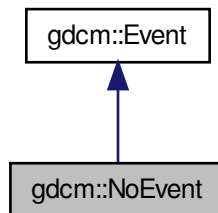
25.180 gdcM::NoEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcM::NoEvent:



Collaboration diagram for gdcM::NoEvent:



Additional Inherited Members

25.180.1 Detailed Description

Define some common GDCM events

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

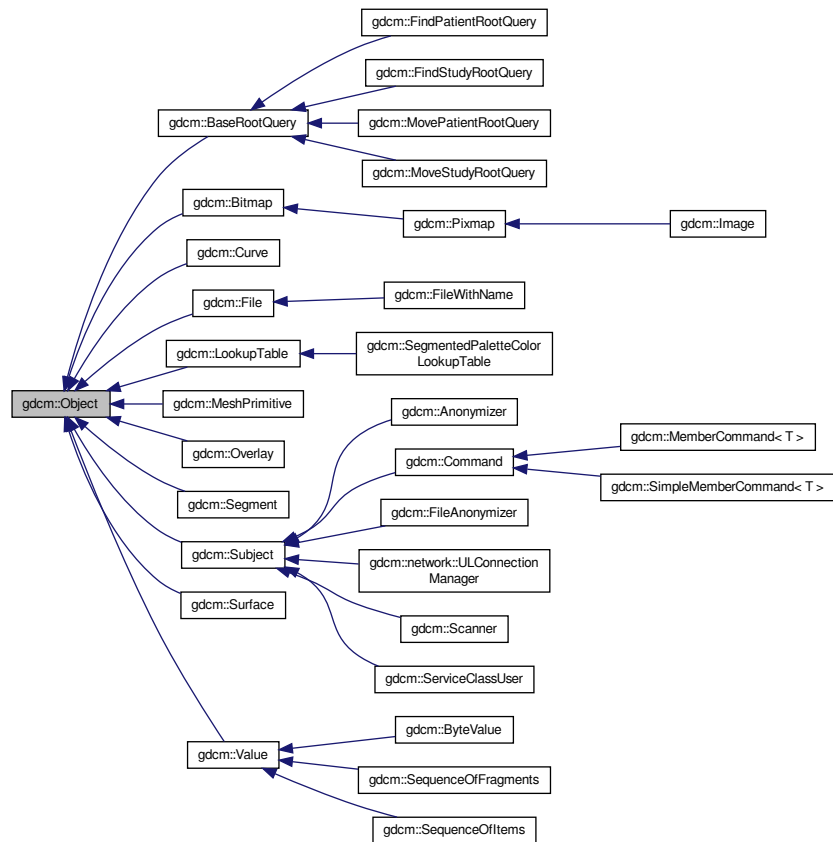
- [gdcMEvent.h](#)

25.181 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](#)

25.181.1 Detailed Description

[Object](#).

Note

main superclass for object that want to use [SmartPointer](#) invasive ref counting system

See Also

[SmartPointer](#)

25.181.2 Constructor & Destructor Documentation

25.181.2.1 `gdcM::Object::Object ()` `[inline]`

25.181.2.2 `virtual gdcM::Object::~~Object ()` `[inline], [virtual]`

25.181.2.3 `gdcM::Object::Object (const Object &)` `[inline]`

Special requirement for copy/cstor, assignment operator.

25.181.3 Member Function Documentation

25.181.3.1 `void gdcM::Object::operator= (const Object &)` `[inline]`

25.181.3.2 `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<<()`.

25.181.3.3 `void gdcM::Object::Register ()` `[inline], [protected]`

25.181.3.4 `void gdcM::Object::UnRegister ()` `[inline], [protected]`

25.181.4 Friends And Related Function Documentation

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

25.181.4.2 `template<class ObjectType > friend class SmartPointer` [`friend`]

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

- [gdcmObject.h](#)

25.182 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 dirsos[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)

25.182.1 Detailed Description

class to handle [Orientation](#)

25.182.2 Member Enumeration Documentation

25.182.2.1 enum `gdcm::Orientation::OrientationType`

Enumerator

UNKNOWN
AXIAL
CORONAL
SAGITTAL
OBLIQUE

25.182.3 Constructor & Destructor Documentation

25.182.3.1 `gdcm::Orientation::Orientation ()`

25.182.3.2 `gdcm::Orientation::~~Orientation ()`

25.182.4 Member Function Documentation

25.182.4.1 `static const char* gdcm::Orientation::GetLabel (OrientationType type)` [static]

Return the label of an [Orientation](#).

25.182.4.2 `static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine (double x, double y, double z)` [static], [protected]

25.182.4.3 `static double gdcm::Orientation::GetObliquityThresholdCosineValue ()` [static]

25.182.4.4 `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

25.182.4.5 `void gdcm::Orientation::Print (std::ostream &) const`

Print.

Referenced by `gdcm::operator<<()`.

25.182.4.6 `static void gdcm::Orientation::SetObliquityThresholdCosineValue (double val)` [static]

ObliquityThresholdCosineValue stuff.

25.182.5 Friends And Related Function Documentation

25.182.5.1 `std::ostream& operator<< (std::ostream &_os, const Orientation &o)` [friend]

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

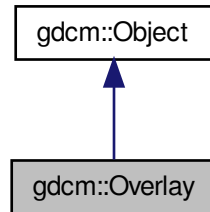
- [gdcmOrientation.h](#)

25.183 gdcm::Overlay Class Reference

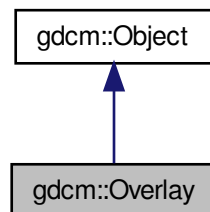
[Overlay](#) class.

```
#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 [Decode](#) (std::istream &is, std::ostream &os)
Do not use.
- 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
- bool [GetBuffer](#) (char *buffer) const
Get the raw (packed bits) Overlay Data:
- 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](#) (unsigned char *buffer) const
Do not use.
- bool [GetUnpackBuffer](#) (char *buffer, size_t len) const
- size_t [GetUnpackBufferLength](#) () const
- bool [GrabOverlayFromPixelData](#) ([DataSet](#) const &ds)
- bool [IsEmpty](#) () const
Return whether or not the Overlay is empty:
- bool [IsInPixelData](#) () const
return if the Overlay is stored in the pixel data or not
- void [IsInPixelData](#) (bool b)
Set whether or no the OverlayData is in the Pixel Data:
- bool [IsZero](#) () const
return true if all bits are set to 0
- 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 numberofframes)
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

25.183.1 Detailed Description

[Overlay](#) class.

Note

see [AreOverlaysInPixelData](#)

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

Example:

25.183.2 Member Enumeration Documentation

25.183.2.1 enum gdcm::Overlay::OverlayType

Enumerator

Invalid

Graphics

ROI

25.183.3 Constructor & Destructor Documentation

25.183.3.1 `gdcmm::Overlay::Overlay ()`

25.183.3.2 `gdcmm::Overlay::~~Overlay ()`

25.183.3.3 `gdcmm::Overlay::Overlay (Overlay const & ov)`

25.183.4 Member Function Documentation

25.183.4.1 `void gdcmm::Overlay::Decode (std::istream & is, std::ostream & os)`

Do not use.

25.183.4.2 `void gdcmm::Overlay::Decompress (std::ostream & os) const`

Decode the internal OverlayData (packed bits) into unpacked representation.

25.183.4.3 `unsigned short gdcmm::Overlay::GetBitPosition () const`

return bit position

25.183.4.4 `unsigned short gdcmm::Overlay::GetBitsAllocated () const`

return bits allocated

25.183.4.5 `bool gdcmm::Overlay::GetBuffer (char * buffer) const`

Get the raw (packed bits) [Overlay](#) Data:

25.183.4.6 `unsigned short gdcmm::Overlay::GetColumns () const`

get columns

25.183.4.7 `const char* gdcmm::Overlay::GetDescription () const`

get description

25.183.4.8 `unsigned short gdcmm::Overlay::GetGroup () const`

Get Group number.

25.183.4.9 `const signed short* gdcmm::Overlay::GetOrigin () const`

get origin

25.183.4.10 `const ByteValue& gdcm::Overlay::GetOverlayData () const`

Return the [Overlay](#) Data as [ByteValue](#): Not thread safe

25.183.4.11 `static const char* gdcm::Overlay::GetOverlayTypeAsString (OverlayType ot) [static]`

25.183.4.12 `static OverlayType gdcm::Overlay::GetOverlayTypeFromString (const char *) [static]`

25.183.4.13 `unsigned short gdcm::Overlay::GetRows () const`

get rows

25.183.4.14 `const char* gdcm::Overlay::GetType () const`

get type

25.183.4.15 `OverlayType gdcm::Overlay::GetTypeAsEnum () const`

25.183.4.16 `bool gdcm::Overlay::GetUnpackBuffer (unsigned char * buffer) const`

Do not use.

25.183.4.17 `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\(\)](#)

25.183.4.18 `size_t gdcm::Overlay::GetUnpackBufferLength () const`

Retrieve the size of the buffer needed to hold the [Overlay](#) as specified by Col & Row parameters

25.183.4.19 `bool gdcm::Overlay::GrabOverlayFromPixelData (DataSet const & ds)`

25.183.4.20 `bool gdcm::Overlay::IsEmpty () const`

Return whether or not the [Overlay](#) is empty:

25.183.4.21 `bool gdcm::Overlay::IsInPixelData () const`

return if the [Overlay](#) is stored in the pixel data or not

25.183.4.22 `void gdcm::Overlay::IsInPixelData (bool b)`

Set wether or no the OverlayData is in the Pixel Data:

25.183.4.23 `bool gdcm::Overlay::IsZero () const`

return true if all bits are set to 0

25.183.4.24 void `gdcmm::Overlay::Print (std::ostream &) const` `[virtual]`

Print.

Reimplemented from [gdcmm::Object](#).

25.183.4.25 void `gdcmm::Overlay::SetBitPosition (unsigned short bitposition)`

set bit position

25.183.4.26 void `gdcmm::Overlay::SetBitsAllocated (unsigned short bitsallocated)`

set bits allocated

25.183.4.27 void `gdcmm::Overlay::SetColumns (unsigned short columns)`

set columns

25.183.4.28 void `gdcmm::Overlay::SetDescription (const char * description)`

set description

25.183.4.29 void `gdcmm::Overlay::SetFrameOrigin (unsigned short frameorigin)`

set frame origin

25.183.4.30 void `gdcmm::Overlay::SetGroup (unsigned short group)`

Set Group number.

25.183.4.31 void `gdcmm::Overlay::SetNumberOfFrames (unsigned int numberofframes)`

set number of frames

25.183.4.32 void `gdcmm::Overlay::SetOrigin (const signed short origin[2])`

set origin

25.183.4.33 void `gdcmm::Overlay::SetOverlay (const char * array, size_t length)`

set overlay from byte array + length

25.183.4.34 void `gdcmm::Overlay::SetRows (unsigned short rows)`

set rows

25.183.4.35 void gdcm::Overlay::SetType (const char * *type*)

set type

25.183.4.36 void gdcm::Overlay::Update (const DataElement & *de*)

Update overlay from data element de:

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

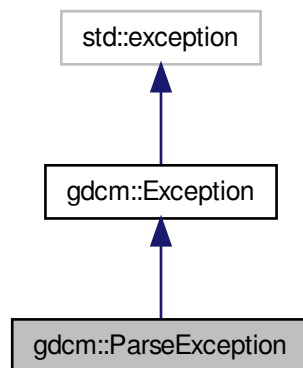
- [gdcmOverlay.h](#)

25.184 gdcm::ParseException Class Reference

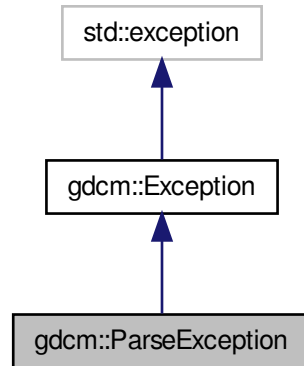
[ParseException](#) Standard exception handling object.

```
#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)

25.184.1 Detailed Description

[ParseException](#) Standard exception handling object.

25.184.2 Constructor & Destructor Documentation

25.184.2.1 `gdcm::ParseException::ParseException ()` `[inline]`

25.184.2.2 `virtual gdcm::ParseException::~~ParseException () throw ()` `[inline], [virtual]`

25.184.3 Member Function Documentation

25.184.3.1 `const DataElement& gdcm::ParseException::GetLastElement () const` `[inline]`

25.184.3.2 `ParseException& gdcm::ParseException::operator= (const ParseException & orig)` `[inline]`

Assignment operator.

25.184.3.3 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](#)

25.185 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](#) ()

25.185.1 Detailed Description

[Parser](#) ala XML_Parser from expat (SAX)

Detailed description here

Note

Simple API for DICOM

25.185.2 Member Typedef Documentation

25.185.2.1 `typedef void(* gdcmm::Parser::EndElementHandler)(void *userData, const Tag &name)`

25.185.2.2 `typedef void(* gdcmm::Parser::StartElementHandler)(void *userData, const Tag &tag, const char *atts[])`

25.185.3 Member Enumeration Documentation

25.185.3.1 `enum gdcmm::Parser::ErrorType`

Enumerator

NoError

NoMemoryError

SyntaxError

NoElementsError

TagMismatchError

DuplicateAttributeError

JunkAfterDocElementError

UndefinedEntityError

UnexpectedStateError

25.185.4 Constructor & Destructor Documentation

25.185.4.1 `gdcmm::Parser::Parser () [inline]`

25.185.4.2 `gdcmm::Parser::~~Parser () [inline]`

25.185.5 Member Function Documentation

25.185.5.1 `char* gdcmm::Parser::GetBuffer (int len) [protected]`

25.185.5.2 `unsigned long gdcmm::Parser::GetCurrentByteIndex () const`

25.185.5.3 `ErrorType gdcmm::Parser::GetErrorCode () const`

25.185.5.4 `static const char* gdcmm::Parser::GetErrorString (ErrorType const & err) [static]`

25.185.5.5 `void* gdcmm::Parser::GetUserData () const`

25.185.5.6 `bool gdcm::Parser::Parse (const char * s, int len, bool isFinal)`

25.185.5.7 `bool gdcm::Parser::ParseBuffer (int len, bool isFinal)` [protected]

25.185.5.8 `ErrorType gdcm::Parser::Process ()` [protected]

25.185.5.9 `void gdcm::Parser::SetElementHandler (StartElementHandler start, EndElementHandler end)`

25.185.5.10 `void gdcm::Parser::SetUserData (void * userData)`

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

- [gdcmParser.h](#)

25.186 gdcm::Patient Class Reference

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

```
#include <gdcmPatient.h>
```

Public Member Functions

- [Patient \(\)](#)

25.186.1 Detailed Description

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

25.186.2 Constructor & Destructor Documentation

25.186.2.1 `gdcm::Patient::Patient ()` [inline]

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

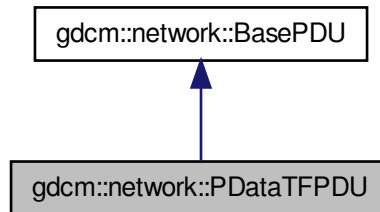
- [gdcmPatient.h](#)

25.187 gdcm::network::PDataTFPDU Class Reference

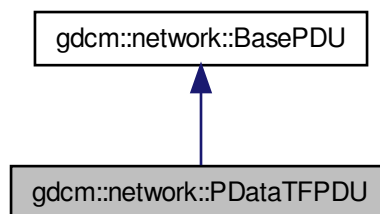
[PDataTFPDU Table](#) 9-22 P-DATA-TF PDU FIELDS.

```
#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)

25.187.1 Detailed Description

[PDataTFPDU Table](#) 9-22 P-DATA-TF PDU FIELDS.

25.187.2 Member Typedef Documentation

25.187.2.1 `typedef std::vector<PresentationDataValue>::size_type gdcm::network::PDataTFPDU::SizeType`

25.187.3 Constructor & Destructor Documentation

25.187.3.1 `gdcm::network::PDataTFPDU::PDataTFPDU ()`

25.187.4 Member Function Documentation

25.187.4.1 `void gdcm::network::PDataTFPDU::AddPresentationDataValue (PresentationDataValue const & pdv)`
[inline]

25.187.4.2 `SizeType gdcm::network::PDataTFPDU::GetNumberOfPresentationDataValues () const` [inline]

25.187.4.3 `PresentationDataValue const& gdcm::network::PDataTFPDU::GetPresentationDataValue (SizeType i) const`
[inline]

25.187.4.4 `bool gdcm::network::PDataTFPDU::IsLastFragment () const` [virtual]

Implements [gdcm::network::BasePDU](#).

25.187.4.5 `void gdcm::network::PDataTFPDU::Print (std::ostream & os) const` [virtual]

Implements [gdcm::network::BasePDU](#).

25.187.4.6 `std::istream& gdcm::network::PDataTFPDU::Read (std::istream & is)` [virtual]

Implements [gdcm::network::BasePDU](#).

25.187.4.7 `std::istream& gdcm::network::PDataTFPDU::ReadInto (std::istream & is, std::ostream & os)` [protected]

25.187.4.8 `size_t gdcm::network::PDataTFPDU::Size () const` [virtual]

Implements [gdcm::network::BasePDU](#).

25.187.4.9 `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:

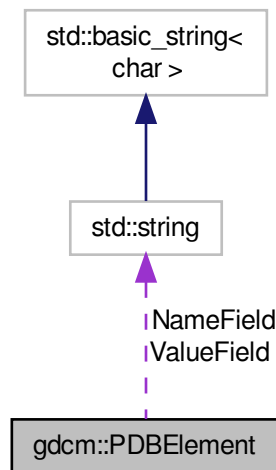
- [gdcmPDataTFPDU.h](#)

25.188 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)`

25.188.1 Detailed Description

Class to represent a PDB [Element](#).

See Also

[PDBHeader](#)

25.188.2 Constructor & Destructor Documentation

25.188.2.1 `gdcm::PDBelement::PDBelement ()` [\[inline\]](#)

25.188.3 Member Function Documentation

25.188.3.1 `const char* gdcm::PDBelement::GetName () const` [\[inline\]](#)

Set/Get Name.

25.188.3.2 `const char* gdcm::PDBelement::GetValue () const` [\[inline\]](#)

Set/Get [Value](#).

25.188.3.3 `bool gdcm::PDBelement::operator== (const PDBelement & de) const` [\[inline\]](#)

References NameField, and ValueField.

25.188.3.4 `void gdcm::PDBelement::SetName (const char * name)` [\[inline\]](#)

25.188.3.5 `void gdcm::PDBelement::SetValue (const char * value)` [\[inline\]](#)

25.188.4 Friends And Related Function Documentation

25.188.4.1 `std::ostream& operator<< (std::ostream & os, const PDBelement & val)` [\[friend\]](#)

25.188.5 Member Data Documentation

25.188.5.1 `std::string gdcm::PDBelement::NameField` [\[protected\]](#)

Referenced by `gdcm::operator<<()`, and `operator==()`.

25.188.5.2 `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](#)

25.189 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)

25.189.1 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](#)

25.189.2 Constructor & Destructor Documentation

25.189.2.1 `gdcm::PDBHeader::PDBHeader ()` `[inline]`

25.189.2.2 `gdcm::PDBHeader::~~PDBHeader ()` `[inline]`

25.189.3 Member Function Documentation

25.189.3.1 `bool gdcm::PDBHeader::FindPDBElementByName (const char * name)`

Return true if the PDB element matching name is found or not.

25.189.3.2 `const PDBElement& gdcm::PDBHeader::GetPDBEnd () const` `[protected]`

25.189.3.3 `const PDBElement& gdcm::PDBHeader::GetPDBElementByName (const char * name)`

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

25.189.3.4 `static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag ()` `[static]`

Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).

25.189.3.5 `bool gdcm::PDBHeader::LoadFromDataElement (DataElement const & de)`

Load the PDB Header from a [DataElement](#) of a [DataSet](#).

25.189.3.6 `void gdcm::PDBHeader::Print (std::ostream & os) const`

Print.

Referenced by `gdcm::operator<<()`.

25.189.4 Friends And Related Function Documentation

25.189.4.1 `std::ostream& operator<< (std::ostream & _os, const PDBHeader & d)` `[friend]`

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

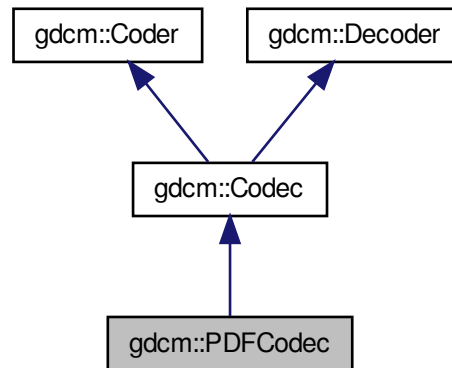
- [gdcmPDBHeader.h](#)

25.190 gdcm::PDFCodec Class Reference

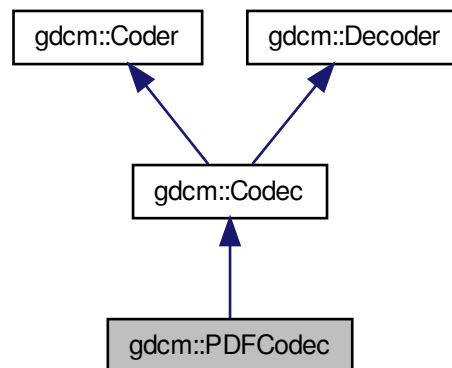
[PDFCodec](#) class.

```
#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

25.190.1 Detailed Description

[PDFCodec](#) class.

25.190.2 Constructor & Destructor Documentation

25.190.2.1 `gdcm::PDFCodec::PDFCodec ()`

25.190.2.2 `gdcm::PDFCodec::~~PDFCodec ()`

25.190.3 Member Function Documentation

25.190.3.1 `bool gdcm::PDFCodec::CanCode (TransferSyntax const &) const` `[inline], [virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

25.190.3.2 `bool gdcm::PDFCodec::CanDecode (TransferSyntax const &) const` `[inline], [virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

25.190.3.3 `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](#)

25.191 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)

25.191.1 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.

25.191.2 Member Function Documentation

- 25.191.2.1 static [BasePDU](#)* [gdcmm::network::PDUFactory::ConstructAbortPDU](#) () [static]
- 25.191.2.2 static [BasePDU](#)* [gdcmm::network::PDUFactory::ConstructPDU](#) (uint8_t *itemtype*) [static]
- 25.191.2.3 static [BasePDU](#)* [gdcmm::network::PDUFactory::ConstructReleasePDU](#) () [static]
- 25.191.2.4 static std::vector<[BasePDU](#)*> [gdcmm::network::PDUFactory::CreateCEchoPDU](#) (const [ULConnection](#) & *inConnection*) [static]
- 25.191.2.5 static std::vector<[BasePDU](#)*> [gdcmm::network::PDUFactory::CreateCFindPDU](#) (const [ULConnection](#) & *inConnection*, const [BaseRootQuery](#) * *inRootQuery*) [static]
- 25.191.2.6 static std::vector<[BasePDU](#)*> [gdcmm::network::PDUFactory::CreateCMovePDU](#) (const [ULConnection](#) & *inConnection*, const [BaseRootQuery](#) * *inRootQuery*) [static]
- 25.191.2.7 static std::vector<[BasePDU](#)*> [gdcmm::network::PDUFactory::CreateCStoreRQPDU](#) (const [ULConnection](#) & *inConnection*, const [File](#) & *file*) [static]
- 25.191.2.8 static std::vector<[BasePDU](#)*> [gdcmm::network::PDUFactory::CreateCStoreRSPPDU](#) (const [DataSet](#) * *inDataSet*, const [BasePDU](#) * *inPC*) [static]
- 25.191.2.9 static [EEventID](#) [gdcmm::network::PDUFactory::DetermineEventByPDU](#) (const [BasePDU](#) * *inPDU*) [static]
- 25.191.2.10 static std::vector<[PresentationDataValue](#)> [gdcmm::network::PDUFactory::GetPDVs](#) (const std::vector< [BasePDU](#) * > & *inDataPDUs*) [static]

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

- [gdcmmPDUFactory.h](#)

25.192 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](#) = '^'

25.192.1 Detailed Description

[PersonName](#) class.

25.192.2 Member Function Documentation

25.192.2.1 unsigned int gdcm::PersonName::GetMaxLength () const [inline]

25.192.2.2 unsigned int gdcm::PersonName::GetNumberOfComponents () const [inline]

25.192.2.3 void gdcm::PersonName::Print (std::ostream & os) const [inline]

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

25.192.2.5 void gdcm::PersonName::SetComponents (const char * *comp1* = " ", const char * *comp2* = " ", const char * *comp3* = " ", const char * *comp4* = " ", const char * *comp5* = " ") [inline]

25.192.2.6 void gdcm::PersonName::SetComponents (const char * *components*[]) [inline]

25.192.3 Member Data Documentation

25.192.3.1 `char gdcM::PersonName::Component[MaxNumberOfComponents][MaxLength+1]`

25.192.3.2 `const unsigned int gdcM::PersonName::MaxLength = 64` `[static]`

25.192.3.3 `const unsigned int gdcM::PersonName::MaxNumberOfComponents = 5` `[static]`

25.192.3.4 `const char gdcM::PersonName::Padding = ''` `[static]`

25.192.3.5 `const char gdcM::PersonName::Separator = '^'` `[static]`

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

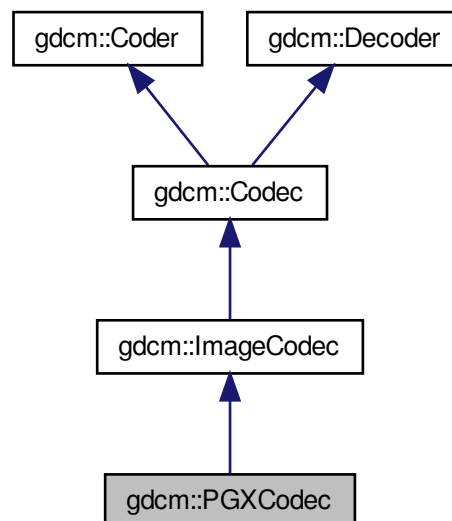
- [gdcMPersonName.h](#)

25.193 gdcM::PGXCodec Class Reference

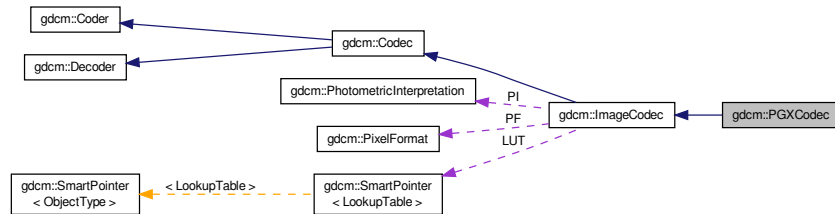
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

```
#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)
- 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

25.193.1 Detailed Description

Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

25.193.2 Constructor & Destructor Documentation

25.193.2.1 `gdcm::PGXCodec::PGXCodec ()`

25.193.2.2 `gdcm::PGXCodec::~~PGXCodec ()`

25.193.3 Member Function Documentation

25.193.3.1 `bool gdcm::PGXCodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.193.3.2 `bool gdcm::PGXCodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.193.3.3 `bool gdcM::PGXCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` [virtual]

Reimplemented from [gdcM::ImageCodec](#).

25.193.3.4 `bool gdcM::PGXCodec::Read (const char * filename, DataElement & out) const`

25.193.3.5 `bool gdcM::PGXCodec::Write (const char * filename, const DataElement & out) const`

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

- [gdcM_PGXCodec.h](#)

25.194 gdcM::PhotometricInterpretation Class Reference

Class to represent an [PhotometricInterpretation](#).

```
#include <gdcMPhotometricInterpretation.h>
```

Public Types

- enum [PType](#) {
[UNKNOWN](#) = 0,
[MONOCHROME1](#),
[MONOCHROME2](#),
[PALETTE_COLOR](#),
[RGB](#),
[HSV](#),
[ARGB](#),
[CMYK](#),
[YBR_FULL](#),
[YBR_FULL_422](#),
[YBR_PARTIAL_422](#),
[YBR_PARTIAL_420](#),
[YBR_ICT](#),
[YBR_RCT](#),
[PI_END](#) }

Public Member Functions

- [PhotometricInterpretation](#) ([PType](#) pi=[UNKNOWN](#))
- unsigned short [GetSamplesPerPixel](#) () const
return the value for Sample Per Pixel associated with a particular Photometric Interpretation
- const char * [GetString](#) () const
- [PType](#) [GetType](#) () const
- bool [IsLossless](#) () const
- bool [IsLossy](#) () const
- bool [IsSameColorSpace](#) ([PhotometricInterpretation](#) const &pi) const
- operator [PType](#) () 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)

25.194.1 Detailed Description

Class to represent an [PhotometricInterpretation](#).

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [HelloVizWorld.cxx](#), and [iU22tomultisc.cxx](#).

25.194.2 Member Enumeration Documentation

25.194.2.1 enum gdcm::PhotometricInterpretation::PIType

Enumerator

UNKNOWN
MONOCHROME1
MONOCHROME2
PALETTE_COLOR
RGB
HSV
ARGB
CMYK
YBR_FULL
YBR_FULL_422
YBR_PARTIAL_422
YBR_PARTIAL_420
YBR_ICT
YBR_RCT
PI_END

25.194.3 Constructor & Destructor Documentation

25.194.3.1 `gdcm::PhotometricInterpretation::PhotometricInterpretation (PIType pi = UNKNOWN)` `[inline]`

25.194.4 Member Function Documentation

25.194.4.1 `static const char* gdcm::PhotometricInterpretation::GetPIString (PIType pi)` `[static]`

Referenced by `gdcm::operator<<()`.

25.194.4.2 `static PType gdcm::PhotometricInterpretation::GetPType (const char * pi)` `[static]`

25.194.4.3 `unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel () const`

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

25.194.4.4 `const char* gdcm::PhotometricInterpretation::GetString () const`

25.194.4.5 `PType gdcm::PhotometricInterpretation::GetType () const` `[inline]`

25.194.4.6 `bool gdcm::PhotometricInterpretation::IsLossless () const`

25.194.4.7 `bool gdcm::PhotometricInterpretation::IsLossy () const`

25.194.4.8 `static bool gdcm::PhotometricInterpretation::IsRetired (PType pi)` `[static]`

25.194.4.9 `bool gdcm::PhotometricInterpretation::IsSameColorSpace (PhotometricInterpretation const & pi) const`

25.194.4.10 `gdcm::PhotometricInterpretation::operator PType () const` `[inline]`

25.194.5 Friends And Related Function Documentation

25.194.5.1 `std::ostream& operator<< (std::ostream & os, const PhotometricInterpretation & pi)` `[friend]`

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

- [gdcmPhotometricInterpretation.h](#)

25.195 gdcm::PixelFormat Class Reference

[PixelFormat](#).

```
#include <gdcmPixelFormat.h>
```

Public Types

- enum [ScalarType](#) {
[UINT8](#),
[INT8](#),
[UINT12](#),
[INT12](#),
[UINT16](#),
[INT16](#),
[UINT32](#),
[INT32](#),
[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)
- [~PixelFormat](#) ()
- 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 [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)

25.195.1 Detailed Description

[PixelFormat](#).

Note

By default the Pixel [Type](#) will be instantiated with the following parameters:

- SamplesPerPixel : 1
- BitsAllocated : 8
- BitsStored : 8
- HighBit : 7
- PixelRepresentation : 0

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSample-Precision.cxx](#), [iU22tomultisc.cxx](#), and [threadgdcm.cxx](#).

25.195.2 Member Enumeration Documentation

25.195.2.1 enum gdcm::PixelFormat::ScalarType

Enumerator

UINT8
INT8
UINT12
INT12
UINT16
INT16
UINT32
INT32
FLOAT16
FLOAT32
FLOAT64
SINGLEBIT
UNKNOWN

25.195.3 Constructor & Destructor Documentation

25.195.3.1 `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]`

25.195.3.2 `gdcm::PixelFormat::PixelFormat (ScalarType st)`

25.195.3.3 `gdcm::PixelFormat::~~PixelFormat ()` `[inline]`

25.195.4 Member Function Documentation

25.195.4.1 unsigned short gdcm::PixelFormat::GetBitsAllocated () const [inline]

BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.

Examples:

[GetJPEGSamplePrecision.cxx](#).

25.195.4.2 unsigned short gdcm::PixelFormat::GetBitsStored () const [inline]

BitsStored see [Tag](#) (0028,0101) US Bits Stored.

Examples:

[GetJPEGSamplePrecision.cxx](#).

25.195.4.3 unsigned short gdcm::PixelFormat::GetHighBit () const [inline]

HighBit see [Tag](#) (0028,0102) US High Bit.

25.195.4.4 int64_t gdcm::PixelFormat::GetMax () const

return the max possible of the pixel

25.195.4.5 int64_t gdcm::PixelFormat::GetMin () const

return the min possible of the pixel

25.195.4.6 unsigned short gdcm::PixelFormat::GetPixelRepresentation () const [inline]

PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.

25.195.4.7 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](#).

25.195.4.8 `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](#).

25.195.4.9 `ScalarType gdcm::PixelFormat::GetScalarType () const`

ScalarType does not take into account the sample per pixel.

25.195.4.10 `const char* gdcm::PixelFormat::GetScalarTypeAsString () const`

25.195.4.11 `bool gdcm::PixelFormat::IsValid () const`

return IsValid

25.195.4.12 `gdcm::PixelFormat::operator ScalarType () const` `[inline]`

25.195.4.13 `bool gdcm::PixelFormat::operator!= (ScalarType st) const` `[inline]`

25.195.4.14 `bool gdcm::PixelFormat::operator!= (const PixelFormat & pf) const` `[inline]`

25.195.4.15 `bool gdcm::PixelFormat::operator== (ScalarType st) const` `[inline]`

25.195.4.16 `bool gdcm::PixelFormat::operator== (const PixelFormat & pf) const` `[inline]`

25.195.4.17 `void gdcm::PixelFormat::Print (std::ostream & os) const`

Print.

Referenced by `gdcm::operator<<()`.

25.195.4.18 `void gdcm::PixelFormat::SetBitsAllocated (unsigned short ba)` `[inline]`

25.195.4.19 `void gdcm::PixelFormat::SetBitsStored (unsigned short bs)` `[inline]`

25.195.4.20 `void gdcm::PixelFormat::SetHighBit (unsigned short hb)` `[inline]`

25.195.4.21 `void gdcm::PixelFormat::SetPixelRepresentation (unsigned short pr)` `[inline]`

25.195.4.22 `void gdcm::PixelFormat::SetSamplesPerPixel (unsigned short spp)` `[inline]`

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [GenFakeImage.cxx](#).

References `gdcmAssertMacro`.

25.195.4.23 void gdcm::PixelFormat::SetScalarType (ScalarType st)

Set [PixelFormat](#) based only on the ScalarType

Warning

: You need to call SetScalarType *before* SetSamplesPerPixel

25.195.4.24 bool gdcm::PixelFormat::Validate () [protected]

When image with 24/24/23 was read, need to validate.

Referenced by gdcm::Bitmap::SetPixelFormat().

25.195.5 Friends And Related Function Documentation

25.195.5.1 friend class Bitmap [friend]

25.195.5.2 std::ostream& operator<< (std::ostream &_os, const PixelFormat & pf) [friend]

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

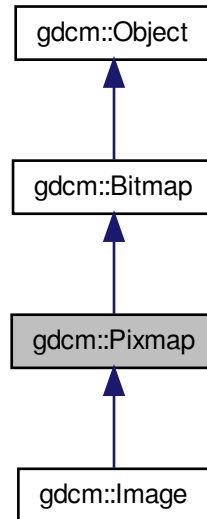
- [gdcmPixelFormat.h](#)

25.196 gdcm::Pixmap Class Reference

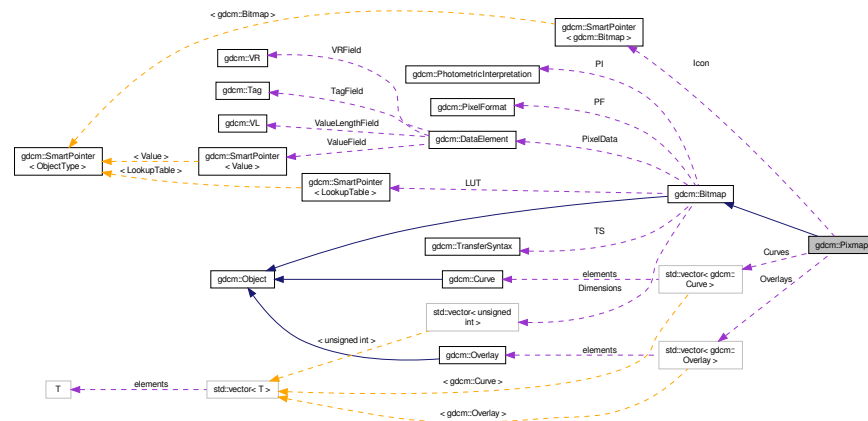
[Pixmap](#) class A bitmap based image. Used as parent for both IconImage and the main Pixel Data [Image](#) It does not contains any World Space information (IPP, IOP)

```
#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

25.196.1 Detailed Description

[Pixmap](#) class A bitmap based image. Used as parent for both [IconImage](#) and the main Pixel Data [Image](#) It does not contains any World Space information (IPP, IOP)

See Also

[PixmapReader](#)

Examples:

[StandardizeFiles.cs.](#)

25.196.2 Constructor & Destructor Documentation

25.196.2.1 [gdcm::Pixmap::Pixmap](#) ()

25.196.2.2 [gdcm::Pixmap::~~Pixmap](#) ()

25.196.3 Member Function Documentation

25.196.3.1 [bool gdcm::Pixmap::AreOverlaysInPixelData](#) () const [virtual]

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from [gdcm::Bitmap](#).

25.196.3.2 `Curve& gdcm::Pixmap::GetCurve (size_t i = 0)` [inline]

[Curve](#): group 50xx.

25.196.3.3 `const Curve& gdcm::Pixmap::GetCurve (size_t i = 0) const` [inline]

25.196.3.4 `const IconImage& gdcm::Pixmap::GetIconImage () const` [inline]

Set/Get Icon [Image](#).

25.196.3.5 `IconImage& gdcm::Pixmap::GetIconImage ()` [inline]

25.196.3.6 `size_t gdcm::Pixmap::GetNumberOfCurves () const` [inline]

25.196.3.7 `size_t gdcm::Pixmap::GetNumberOfOverlays () const` [inline]

25.196.3.8 `Overlay& gdcm::Pixmap::GetOverlay (size_t i = 0)` [inline]

[Overlay](#): group 60xx.

25.196.3.9 `const Overlay& gdcm::Pixmap::GetOverlay (size_t i = 0) const` [inline]

25.196.3.10 `void gdcm::Pixmap::Print (std::ostream &) const` [virtual]

Reimplemented from [gdcm::Bitmap](#).

25.196.3.11 `void gdcm::Pixmap::RemoveOverlay (size_t i)` [inline]

25.196.3.12 `void gdcm::Pixmap::SetIconImage (IconImage const & ii)` [inline]

25.196.3.13 `void gdcm::Pixmap::SetNumberOfCurves (size_t n)` [inline]

25.196.3.14 `void gdcm::Pixmap::SetNumberOfOverlays (size_t n)` [inline]

25.196.4 Member Data Documentation

25.196.4.1 `std::vector<Curve> gdcm::Pixmap::Curves` [protected]

25.196.4.2 `SmartPointer<IconImage> gdcm::Pixmap::Icon` [protected]

25.196.4.3 `std::vector<Overlay> gdcm::Pixmap::Overlays` [protected]

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

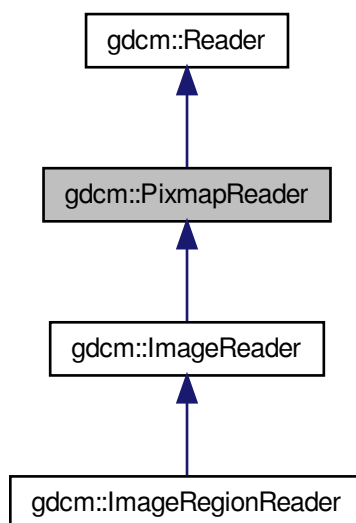
- [gdcmPixmap.h](#)

25.197 gdcm::PixmapReader Class Reference

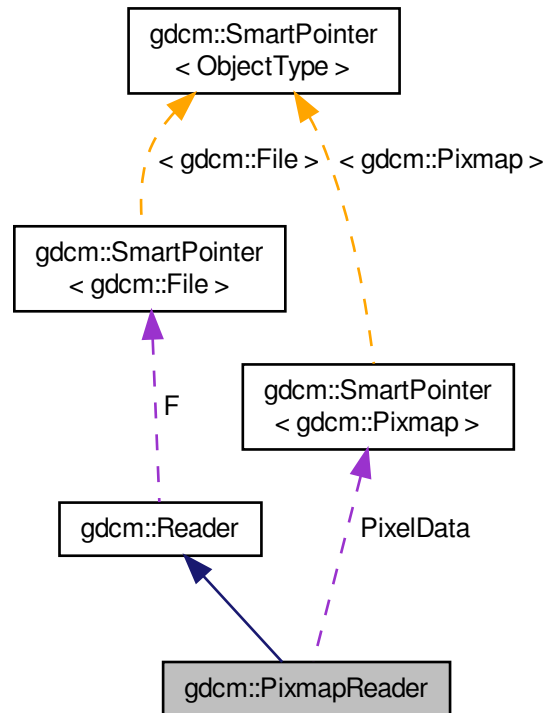
[PixmapReader](#).

```
#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 [ReadACRNEMAIImage](#) ()
- virtual bool [ReadImage](#) ([MediaStorage](#) const &ms)
- bool [ReadImageInternal](#) ([MediaStorage](#) const &ms, bool handlepixeldata=true)

Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

25.197.1 Detailed Description

[PixmapReader](#).

Note

its role is to convert the DICOM [DataSet](#) into a [gdcmm::Pixmap](#) representation By default it is also loading the lookup table and overlay when found as they impact the rendering of the image

See PS 3.3-2008, [Table C.7-11b IMAGE PIXEL MACRO ATTRIBUTES](#) for the list of attribute that belong to what gdcmm calls a 'Pixmap'

Warning

the API `ReadUpToTag` and `ReadSelectedTag`

See Also

[Pixmap](#)

25.197.2 Constructor & Destructor Documentation

25.197.2.1 `gdcmm::PixmapReader::PixmapReader ()`

25.197.2.2 `virtual gdcmm::PixmapReader::~~PixmapReader () [virtual]`

25.197.3 Member Function Documentation

25.197.3.1 `const Pixmap& gdcmm::PixmapReader::GetPixmap () const`

Return the read image (need to call [Read\(\)](#) first)

25.197.3.2 `Pixmap& gdcmm::PixmapReader::GetPixmap ()`

25.197.3.3 `virtual bool gdcmm::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 [gdcmm::Reader](#).

Reimplemented in [gdcmm::ImageRegionReader](#), and [gdcmm::ImageReader](#).

25.197.3.4 `virtual bool gdcmm::PixmapReader::ReadACRNEMAIImage () [protected], [virtual]`

Reimplemented in [gdcmm::ImageReader](#).

25.197.3.5 `virtual bool gdcmm::PixmapReader::ReadImage (MediaStorage const & ms) [protected], [virtual]`

Reimplemented in [gdcmm::ImageReader](#).

25.197.3.6 `bool gdcm::PixmapReader::ReadImageInternal (MediaStorage const & ms, bool handlepixeldata = true)`
`[protected]`

25.197.4 Member Data Documentation

25.197.4.1 `SmartPointer<Pixmap> gdcm::PixmapReader::PixelData` `[protected]`

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

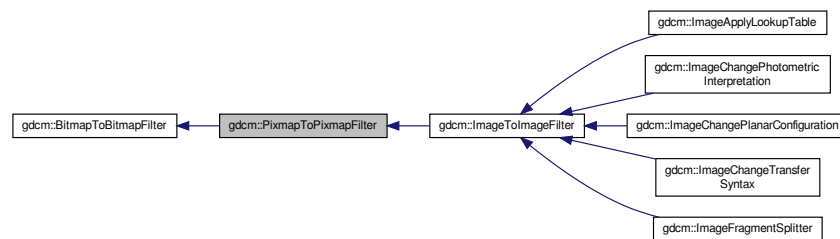
- [gdcmPixmapReader.h](#)

25.198 gdcm::PixmapToPixmapFilter Class Reference

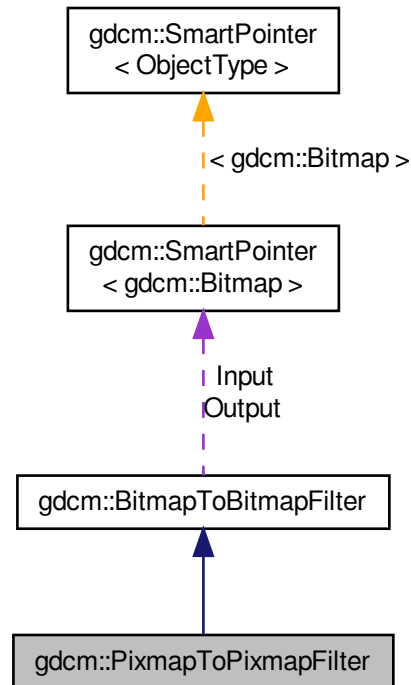
[PixmapToPixmapFilter](#) class Super class for all filter taking an image and producing an output image.

```
#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

25.198.1 Detailed Description

[PixmapToPixmapFilter](#) class Super class for all filter taking an image and producing an output image.

25.198.2 Constructor & Destructor Documentation

25.198.2.1 `gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ()`

25.198.2.2 `gdcm::PixmapToPixmapFilter::~~PixmapToPixmapFilter () [inline]`

25.198.3 Member Function Documentation

25.198.3.1 `Pixmap& gdcm::PixmapToPixmapFilter::GetInput ()`

25.198.3.2 `const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput () const`

Get Output image.

25.198.3.3 `const Pixmap& gdcm::PixmapToPixmapFilter::GetOutputAsPixmap () const`

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

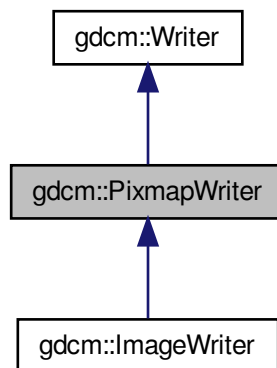
- [gdcmPixmapToPixmapFilter.h](#)

25.199 gdcm::PixmapWriter Class Reference

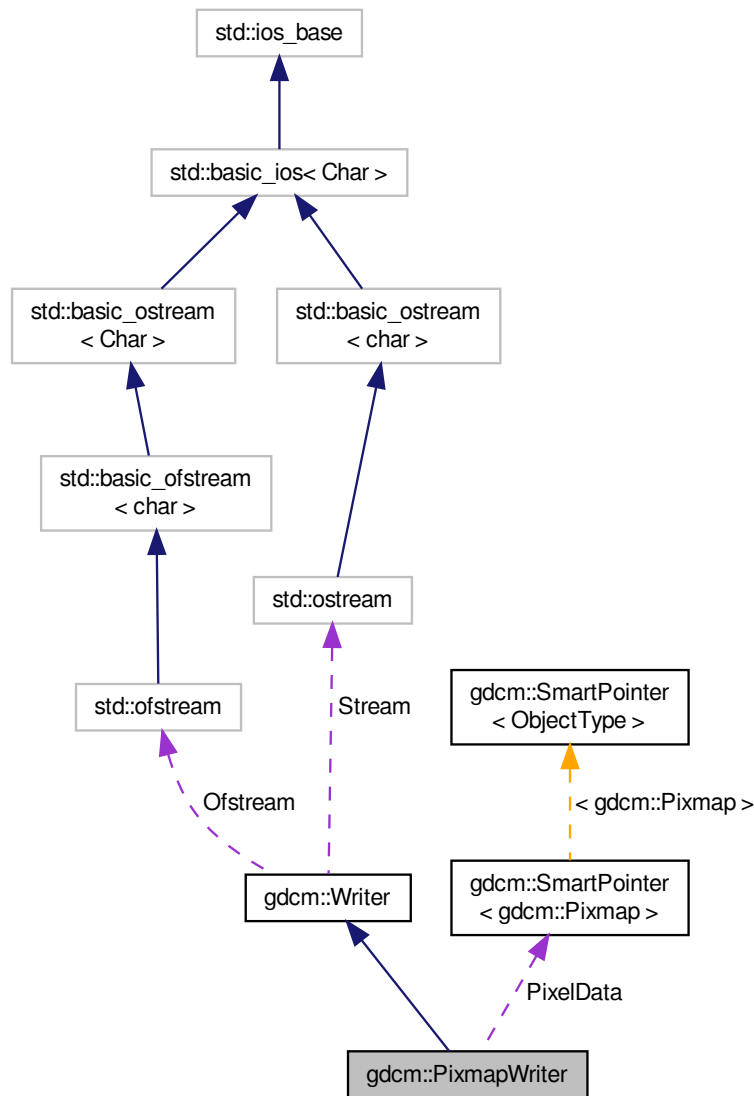
[PixmapWriter](#) This class will takes two inputs:

```
#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 [DolconImage](#) ([DataSet](#) &ds, [Pixmap](#) const &image)
- bool [PrepareWrite](#) ()

Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

25.199.1 Detailed Description

[PixmapWriter](#) This class will takes two inputs:

1. The DICOM [DataSet](#)
2. The [Image](#) input It will override any info from the [Image](#) over the [DataSet](#).

For instance when one read in a lossy compressed image and write out as unencapsulated (ie implicitly lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

25.199.2 Constructor & Destructor Documentation

25.199.2.1 `gdcm::PixmapWriter::PixmapWriter ()`

25.199.2.2 `gdcm::PixmapWriter::~~PixmapWriter ()`

25.199.3 Member Function Documentation

25.199.3.1 `void gdcm::PixmapWriter::DolconImage (DataSet & ds, Pixmap const & image)` `[protected]`

25.199.3.2 `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](#).

25.199.3.3 `virtual Pixmap& gdcm::PixmapWriter::GetImage ()` `[inline],[virtual]`

Reimplemented in [gdcm::ImageWriter](#).

25.199.3.4 `const Pixmap& gdcm::PixmapWriter::GetPixmap () const` [inline]

25.199.3.5 `Pixmap& gdcm::PixmapWriter::GetPixmap ()` [inline]

25.199.3.6 `bool gdcm::PixmapWriter::PrepareWrite ()` [protected]

25.199.3.7 `virtual void gdcm::PixmapWriter::SetImage (Pixmap const & img)` [virtual]

Examples:

[CompressImage.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), and [MergeTwoFiles.cxx](#).

25.199.3.8 `void gdcm::PixmapWriter::SetPixmap (Pixmap const & img)`

25.199.3.9 `bool gdcm::PixmapWriter::Write ()` [virtual]

Write.

Reimplemented from [gdcm::Writer](#).

25.199.4 Member Data Documentation

25.199.4.1 `SmartPointer<Pixmap> gdcm::PixmapWriter::PixelData` [protected]

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

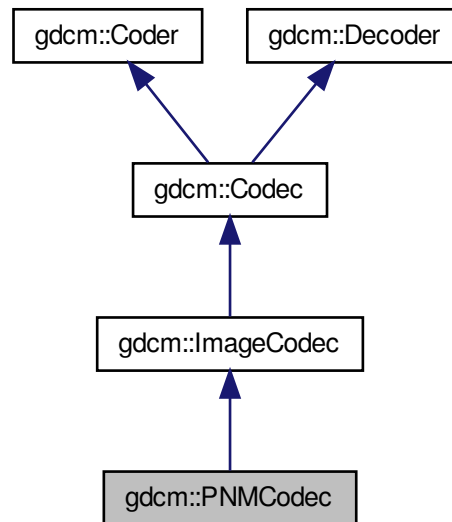
- [gdcmPixmapWriter.h](#)

25.200 gdcm::PNMCodec Class Reference

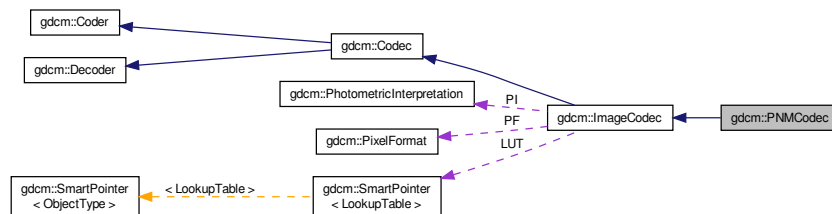
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.

```
#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)
- 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

25.200.1 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](#).

25.200.2 Constructor & Destructor Documentation

25.200.2.1 `gdcm::PNMCodec::PNMCodec ()`

25.200.2.2 `gdcm::PNMCodec::~~PNMCodec ()`

25.200.3 Member Function Documentation

25.200.3.1 `bool gdcm::PNMCodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.200.3.2 `bool gdcm::PNMCodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.200.3.3 `unsigned long gdcm::PNMCodec::GetBufferLength () const` `[inline]`

25.200.3.4 `bool gdcm::PNMCodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts)` `[virtual]`

Reimplemented from [gdcm::ImageCodec](#).

25.200.3.5 `bool gdcm::PNMCodec::Read (const char * filename, DataElement & out) const`

25.200.3.6 `void gdcm::PNMCodec::SetBufferLength (unsigned long l)` `[inline]`

25.200.3.7 `bool gdcmm::PNMCodec::Write (const char * filename, const DataElement & out) const`

Examples:

[ExtractIconFromFile.cxx](#).

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

- [gdcmmPNMCodec.h](#)

25.201 gdcmm::Preamble Class Reference

DICOM [Preamble](#) (Part 10)

```
#include <gdcmmPreamble.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)

25.201.1 Detailed Description

DICOM [Preamble](#) (Part 10)

25.201.2 Constructor & Destructor Documentation

25.201.2.1 `gdcm::Preamble::Preamble ()`

25.201.2.2 `gdcm::Preamble::~~Preamble ()`

25.201.2.3 `gdcm::Preamble::Preamble (Preamble const &)` `[inline]`

25.201.3 Member Function Documentation

25.201.3.1 `void gdcm::Preamble::Clear ()`

25.201.3.2 `void gdcm::Preamble::Create ()`

25.201.3.3 `const char* gdcm::Preamble::GetInternal () const` `[inline]`

25.201.3.4 `VL gdcm::Preamble::GetLength () const` `[inline]`

25.201.3.5 `bool gdcm::Preamble::IsEmpty () const` `[inline]`

25.201.3.6 `bool gdcm::Preamble::IsValid () const` `[inline]`, `[protected]`

25.201.3.7 `Preamble& gdcm::Preamble::operator= (Preamble const &)` `[inline]`

25.201.3.8 `void gdcm::Preamble::Print (std::ostream & os) const`

25.201.3.9 `std::istream& gdcm::Preamble::Read (std::istream & is)`

25.201.3.10 `void gdcm::Preamble::Remove ()`

25.201.3.11 `void gdcm::Preamble::Valid ()`

25.201.3.12 `std::ostream const& gdcm::Preamble::Write (std::ostream & os) const`

25.201.4 Friends And Related Function Documentation

25.201.4.1 `std::ostream& operator<< (std::ostream & _os, const Preamble & _val)` `[friend]`

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

- [gdcmPreamble.h](#)

25.202 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::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
- void [AddTransferSyntax](#) (const char *tsstr)
- const char * [GetAbstractSyntax](#) () const
- [SizeType](#) [GetNumberOfTransferSyntaxes](#) () const
- uint8_t [GetPresentationContextID](#) () const
- const char * [GetTransferSyntax](#) ([SizeType](#) i) const
- bool [operator==](#) (const [PresentationContext](#) &pc) const
- void [Print](#) (std::ostream &os) const
- void [SetAbstractSyntax](#) (const char *as)
- void [SetPresentationContextID](#) (uint8_t id)

25.202.1 Detailed Description

[PresentationContext](#).

See Also

[PresentationContextAC](#) [PresentationContextRQ](#)

25.202.2 Member Typedef Documentation

25.202.2.1 typedef TransferSyntaxArrayType::size_type [gdcm::PresentationContext::SizeType](#)

25.202.2.2 typedef std::vector<std::string> [gdcm::PresentationContext::TransferSyntaxArrayType](#)

25.202.3 Constructor & Destructor Documentation

25.202.3.1 [gdcm::PresentationContext::PresentationContext](#) ()

25.202.3.2 [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).

25.202.4 Member Function Documentation

25.202.4.1 void [gdcm::PresentationContext::AddTransferSyntax](#) (const char * tsstr)

- 25.202.4.2 `const char* gdcm::PresentationContext::GetAbstractSyntax () const` [inline]
- 25.202.4.3 `SizeType gdcm::PresentationContext::GetNumberOfTransferSyntaxes () const` [inline]
- 25.202.4.4 `uint8_t gdcm::PresentationContext::GetPresentationContextID () const`
- 25.202.4.5 `const char* gdcm::PresentationContext::GetTransferSyntax (SizeType i) const` [inline]
- 25.202.4.6 `bool gdcm::PresentationContext::operator== (const PresentationContext & pc) const` [inline]
- 25.202.4.7 `void gdcm::PresentationContext::Print (std::ostream & os) const`
- 25.202.4.8 `void gdcm::PresentationContext::SetAbstractSyntax (const char * as)` [inline]
- 25.202.4.9 `void gdcm::PresentationContext::SetPresentationContextID (uint8_t id)`

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

- [gdcmPresentationContext.h](#)

25.203 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`

25.203.1 Detailed Description

[PresentationContextAC Table](#) 9-18 PRESENTATION CONTEXT ITEM FIELDS.

See Also

[PresentationContext](#)

25.203.2 Constructor & Destructor Documentation

25.203.2.1 `gdcm::network::PresentationContextAC::PresentationContextAC ()`

25.203.3 Member Function Documentation

25.203.3.1 `uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID () const` `[inline]`

25.203.3.2 `uint8_t gdcm::network::PresentationContextAC::GetReason () const` `[inline]`

25.203.3.3 `TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax () const` `[inline]`

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

25.203.3.5 `std::istream& gdcm::network::PresentationContextAC::Read (std::istream & is)`

25.203.3.6 `void gdcm::network::PresentationContextAC::SetPresentationContextID (uint8_t id)`

25.203.3.7 `void gdcm::network::PresentationContextAC::SetReason (uint8_t r)` `[inline]`

25.203.3.8 `void gdcm::network::PresentationContextAC::SetTransferSyntax (TransferSyntaxSub const & ts)`

25.203.3.9 `size_t gdcm::network::PresentationContextAC::Size () const`

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

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

- [gdcmPresentationContextAC.h](#)

25.204 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

25.204.1 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-S-TORE (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](#).

25.204.2 Member Typedef Documentation

25.204.2.1 `typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType`

25.204.2.2 `typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType`

25.204.3 Constructor & Destructor Documentation

25.204.3.1 `gdcm::PresentationContextGenerator::PresentationContextGenerator ()`

25.204.4 Member Function Documentation

25.204.4.1 `bool gdcM::PresentationContextGenerator::AddPresentationContext (const char * as, const char * ts)`
`[protected]`

25.204.4.2 `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-STO-RE operations

Examples:

[CStoreQtProgress.cxx](#).

25.204.4.3 `bool gdcM::PresentationContextGenerator::GenerateFromUID (UIDs::TSName asname)`

Generate the [PresentationContext](#) array from a UID (eg. VerificationSOPClass)

25.204.4.4 `const char* gdcM::PresentationContextGenerator::GetDefaultTransferSyntax () const` `[protected]`

25.204.4.5 `PresentationContextArrayType const& gdcM::PresentationContextGenerator::GetPresentationContexts ()`
`[inline]`

Examples:

[CStoreQtProgress.cxx](#).

25.204.4.6 `void gdcM::PresentationContextGenerator::SetDefaultTransferSyntax (const TransferSyntax & ts)`

Not implemented for now. GDCM internally uses Implicit Little Endian.

25.204.4.7 `void gdcM::PresentationContextGenerator::SetMergeModeToAbstractSyntax ()`

25.204.4.8 `void gdcM::PresentationContextGenerator::SetMergeModeToTransferSyntax ()`

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

- [gdcMPresentationContextGenerator.h](#)

25.205 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::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM)
- [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

25.205.1 Detailed Description

[PresentationContextRQ](#) Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

See Also

[PresentationContextAC](#)

25.205.2 Member Typedef Documentation

25.205.2.1 typedef std::vector<[TransferSyntaxSub](#)>::size_type gdcm::network::PresentationContextRQ::SizeType

25.205.3 Constructor & Destructor Documentation

25.205.3.1 gdcm::network::PresentationContextRQ::PresentationContextRQ ()

25.205.3.2 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 (default to Implicit [VR](#) LittleEndian when not specified).

25.205.3.3 gdcm::network::PresentationContextRQ::PresentationContextRQ (const [PresentationContext](#) & pc)

25.205.4 Member Function Documentation

25.205.4.1 void gdcm::network::PresentationContextRQ::AddTransferSyntax ([TransferSyntaxSub](#) const & ts)

- 25.205.4.2 **AbstractSyntax** const& gdcmm::network::PresentationContextRQ::GetAbstractSyntax () const [inline]
- 25.205.4.3 **AbstractSyntax&** gdcmm::network::PresentationContextRQ::GetAbstractSyntax () [inline]
- 25.205.4.4 **SizeType** gdcmm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes () const [inline]
- 25.205.4.5 **uint8_t** gdcmm::network::PresentationContextRQ::GetPresentationContextID () const
- 25.205.4.6 **TransferSyntaxSub** const& gdcmm::network::PresentationContextRQ::GetTransferSyntax (**SizeType** *i*) const [inline]
- 25.205.4.7 **TransferSyntaxSub&** gdcmm::network::PresentationContextRQ::GetTransferSyntax (**SizeType** *i*) [inline]
- 25.205.4.8 **std::vector<TransferSyntaxSub>** const& gdcmm::network::PresentationContextRQ::GetTransferSyntaxes () const [inline]
- 25.205.4.9 **bool** gdcmm::network::PresentationContextRQ::operator== (const **PresentationContextRQ** & *pc*) const [inline]
- 25.205.4.10 **void** gdcmm::network::PresentationContextRQ::Print (**std::ostream** & *os*) const
- 25.205.4.11 **std::istream&** gdcmm::network::PresentationContextRQ::Read (**std::istream** & *is*)
- 25.205.4.12 **void** gdcmm::network::PresentationContextRQ::SetAbstractSyntax (**AbstractSyntax** const & *as*)
- 25.205.4.13 **void** gdcmm::network::PresentationContextRQ::SetPresentationContextID (**uint8_t** *id*)
- 25.205.4.14 **size_t** gdcmm::network::PresentationContextRQ::Size () const
- 25.205.4.15 **const std::ostream&** gdcmm::network::PresentationContextRQ::Write (**std::ostream** & *os*) const

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

- [gdcmmPresentationContextRQ.h](#)

25.206 gdcmm::network::PresentationDataValue Class Reference

[PresentationDataValue Table](#) 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

```
#include <gdcmmPresentationDataValue.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)

25.206.1 Detailed Description

[PresentationDataValue Table](#) 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

25.206.2 Constructor & Destructor Documentation

25.206.2.1 `gdcm::network::PresentationDataValue::PresentationDataValue ()`

25.206.3 Member Function Documentation

25.206.3.1 `static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs (const std::vector< PresentationDataValue > & inPDVs) [static]`

Warning

[DataSet](#) will be read as Implicit Little Endian TS

25.206.3.2 `const std::string& gdcm::network::PresentationDataValue::GetBlob () const`

25.206.3.3 `bool gdcm::network::PresentationDataValue::GetIsCommand () const`

25.206.3.4 `bool gdcm::network::PresentationDataValue::GetIsLastFragment () const`

25.206.3.5 `uint8_t gdcm::network::PresentationDataValue::GetMessageHeader () const [inline]`

25.206.3.6 `uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID () const [inline]`

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

25.206.3.8 `std::istream& gdcm::network::PresentationDataValue::Read (std::istream & is)`

25.206.3.9 `std::istream& gdcm::network::PresentationDataValue::ReadInto (std::istream & is, std::ostream & os)`

25.206.3.10 `void gdcm::network::PresentationDataValue::SetBlob (const std::string & partialblob)`

25.206.3.11 `void gdcmm::network::PresentationDataValue::SetCommand (bool inCommand)`

25.206.3.12 `void gdcmm::network::PresentationDataValue::SetDataSet (const DataSet & ds)`

Set [DataSet](#). Write [DataSet](#) in implicit.

Warning

size of dataset should be below maxpdusize

25.206.3.13 `void gdcmm::network::PresentationDataValue::SetLastFragment (bool inLast)`

25.206.3.14 `void gdcmm::network::PresentationDataValue::SetMessageHeader (uint8_t messageheader) [inline]`

25.206.3.15 `void gdcmm::network::PresentationDataValue::SetPresentationContextID (uint8_t id) [inline]`

25.206.3.16 `size_t gdcmm::network::PresentationDataValue::Size () const`

25.206.3.17 `const std::ostream& gdcmm::network::PresentationDataValue::Write (std::ostream & os) const`

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

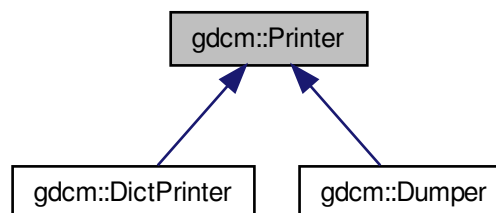
- [gdcmmPresentationDataValue.h](#)

25.207 gdcmm::Printer Class Reference

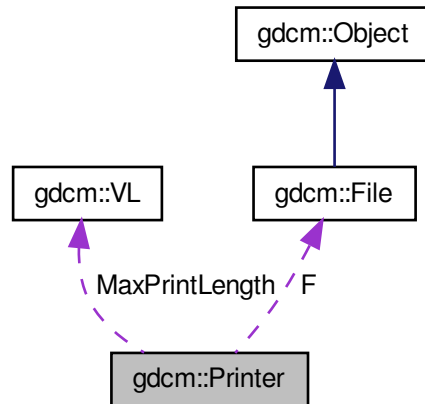
[Printer](#) class.

```
#include <gdcmmPrinter.h>
```

Inheritance diagram for gdcmm::Printer:



Collaboration diagram for gdcmm::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::ostream &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

25.207.1 Detailed Description

[Printer](#) class.

25.207.2 Member Enumeration Documentation

25.207.2.1 enum gdcm::Printer::PrintStyles

Enumerator

VERBOSE_STYLE
CONDENSED_STYLE
XML

25.207.3 Constructor & Destructor Documentation

25.207.3.1 gdcm::Printer::Printer ()

25.207.3.2 gdcm::Printer::~~Printer ()

25.207.4 Member Function Documentation

25.207.4.1 [PrintStyles](#) gdcm::Printer::GetPrintStyle () const [inline]

Get PrintStyle value.

25.207.4.2 void gdcm::Printer::Print (std::ostream & os)

Print.

25.207.4.3 VR gdcm::Printer::PrintDataElement (std::ostream & os, const [Dicts](#) & *dicts*, const [DataSet](#) & *ds*, const [DataElement](#) & *de*, std::ostream & *out*, std::string const & *indent*) [protected]

25.207.4.4 void gdcm::Printer::PrintDataSet (const [DataSet](#) & *ds*, std::ostream & *os*, const std::string & *s* = " ")

Print an individual dataset.

25.207.4.5 void gdcm::Printer::PrintSQ (const SequenceOfItems * *sqi*, std::ostream & *os*, std::string const & *indent*)
[protected]

25.207.4.6 void gdcm::Printer::SetColor (bool *c*)

Set color mode or not.

25.207.4.7 void gdcm::Printer::SetFile (File const & *f*) [inline]

Set file.

25.207.4.8 void gdcm::Printer::SetStyle (PrintStyles *ps*) [inline]

Set PrintStyle value.

25.207.5 Member Data Documentation

25.207.5.1 const File* gdcm::Printer::F [protected]

25.207.5.2 VL gdcm::Printer::MaxPrintLength [protected]

25.207.5.3 PrintStyles gdcm::Printer::PrintStyle [protected]

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

- [gdcmPrinter.h](#)

25.208 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)`

25.208.1 Detailed Description

Private [Dict](#).

25.208.2 Constructor & Destructor Documentation

25.208.2.1 `gdcm::PrivateDict::PrivateDict ()` `[inline]`

25.208.2.2 `gdcm::PrivateDict::~~PrivateDict ()` `[inline]`

25.208.3 Member Function Documentation

25.208.3.1 `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`.

25.208.3.2 `bool gdcm::PrivateDict::FindDictEntry (const PrivateTag &tag) const` `[inline]`

25.208.3.3 `const DictEntry& gdcm::PrivateDict::GetDictEntry (const PrivateTag &tag) const` `[inline]`

25.208.3.4 `bool gdcm::PrivateDict::IsEmpty () const` `[inline]`

25.208.3.5 `void gdcm::PrivateDict::LoadDefault ()` `[protected]`

25.208.3.6 `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()`.

25.208.3.7 `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.

25.208.4 Friends And Related Function Documentation

25.208.4.1 `friend class Dicts` `[friend]`

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

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

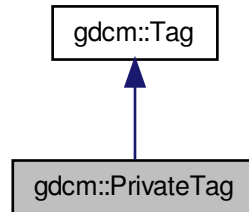
- [gdcmDict.h](#)

25.209 gdcm::PrivateTag Class Reference

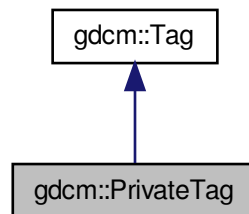
Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)

```
#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="")
- 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)

25.209.1 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:

[csa2img.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [ELSCINT1WaveToText.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [MrProtocol.cxx](#), [pmsct_rgb1.cxx](#), [PublicDict.cxx](#), [ReadGEMSS-DO.cxx](#), and [rle2img.cxx](#).

25.209.2 Constructor & Destructor Documentation

25.209.2.1 `gdcm::PrivateTag::PrivateTag (uint16_t group = 0, uint16_t element = 0, const char * owner = " ") [inline]`

25.209.3 Member Function Documentation

25.209.3.1 `const char* gdcm::PrivateTag::GetOwner () const [inline]`

Examples:

[PublicDict.cxx](#).

Referenced by `gdcm::PrivateDict::PrintXML()`.

25.209.3.2 `bool gdcm::PrivateTag::operator< (const PrivateTag & _val) const`

25.209.3.3 `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")

25.209.3.4 `void gdcm::PrivateTag::SetOwner (const char * owner) [inline]`

25.209.4 Friends And Related Function Documentation

25.209.4.1 `std::ostream& operator<< (std::ostream & _os, const PrivateTag & _val) [friend]`

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

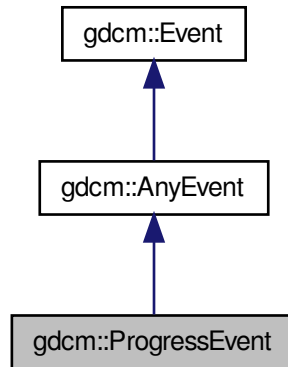
- [gdcmPrivateTag.h](#)

25.210 gdcm::ProgressEvent Class Reference

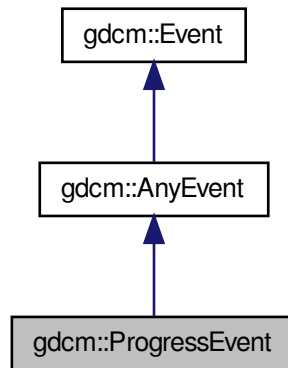
[ProgressEvent](#) Special type of event triggered during.

```
#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 [::gdcmm::Event](#) *e) const
- virtual const char * [GetEventName](#) () const
- double [GetProgress](#) () const
- virtual [::gdcmm::Event](#) * [MakeObject](#) () const
- void [SetProgress](#) (double p)

25.210.1 Detailed Description

[ProgressEvent](#) Special type of event triggered during.

See Also

[AnyEvent](#)

25.210.2 Member Typedef Documentation

25.210.2.1 `typedef ProgressEvent gdcmm::ProgressEvent::Self`

25.210.2.2 `typedef AnyEvent gdcmm::ProgressEvent::Superclass`

25.210.3 Constructor & Destructor Documentation

25.210.3.1 `gdcmm::ProgressEvent::ProgressEvent (double p = 0) [inline]`

25.210.3.2 `virtual gdcmm::ProgressEvent::~~ProgressEvent () [inline],[virtual]`

25.210.3.3 `gdcmm::ProgressEvent::ProgressEvent (const Self & s) [inline]`

25.210.4 Member Function Documentation

25.210.4.1 `virtual bool gdcmm::ProgressEvent::CheckEvent (const ::gdcmm::Event * e) const [inline],[virtual]`

25.210.4.2 `virtual const char* gdcmm::ProgressEvent::GetEventName () const [inline],[virtual]`

Return the StringName associated with the event.

Implements [gdcmm::Event](#).

25.210.4.3 `double gdcmm::ProgressEvent::GetProgress () const [inline]`

25.210.4.4 `virtual ::gdcmm::Event* gdcmm::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 [gdcmm::Event](#).

25.210.4.5 `void gdcmm::ProgressEvent::SetProgress (double p) [inline]`

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

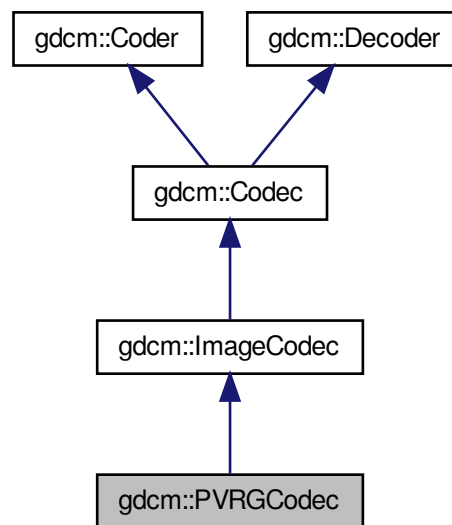
- [gdcmProgressEvent.h](#)

25.211 gdcm::PVRGCodec Class Reference

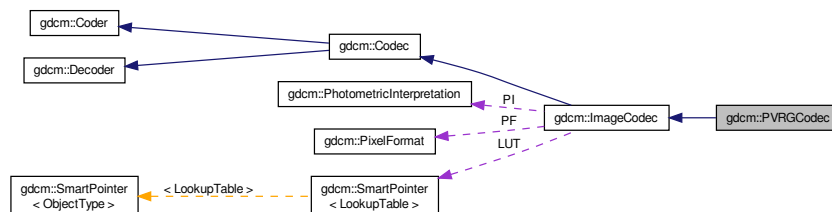
[PVRGCodec](#).

```
#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)
- bool [Code](#) ([DataElement](#) const &in, [DataElement](#) &out)
Code.
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)
Decode.

Additional Inherited Members

25.211.1 Detailed Description

[PVRGCodec](#).

Note

pvr 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_Gyrosan-12-Jpeg_Extended_Process_2_4.dcm

we have to...

25.211.2 Constructor & Destructor Documentation

25.211.2.1 `gdcm::PVRGCodec::PVRGCodec ()`

25.211.2.2 `gdcm::PVRGCodec::~~PVRGCodec ()`

25.211.3 Member Function Documentation

25.211.3.1 `bool gdcm::PVRGCodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.211.3.2 `bool gdcm::PVRGCodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.211.3.3 `bool gdcm::PVRGCodec::Code (DataElement const & in_, DataElement & out_)` `[virtual]`

Code.

Reimplemented from [gdcm::Coder](#).

25.211.3.4 `bool gdcm::PVRGCodec::Decode (DataElement const & , DataElement &) [virtual]`

Decode.

Reimplemented from [gdcm::ImageCodec](#).

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

- [gdcmPVRGCodec.h](#)

25.212 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)

25.212.1 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.

25.212.2 Constructor & Destructor Documentation

25.212.2.1 `gdcm::PythonFilter::PythonFilter ()`

25.212.2.2 `gdcm::PythonFilter::~~PythonFilter ()`

25.212.3 Member Function Documentation

25.212.3.1 `File& gdcm::PythonFilter::GetFile () [inline]`

25.212.3.2 `const File& gdcm::PythonFilter::GetFile () const [inline]`

25.212.3.3 `void gdcm::PythonFilter::SetDicts (const Dicts &dicts)`

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

25.212.3.5 `PyObject* gdcm::PythonFilter::ToPyObject (const Tag &t) const`

25.212.3.6 void `gdcm::PythonFilter::UseDictAlways` (bool *use*) [inline]

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

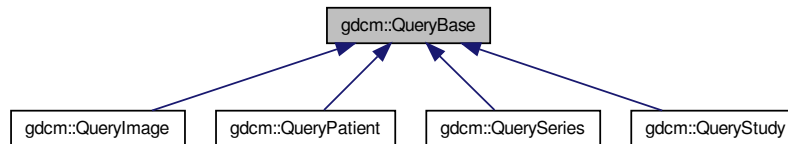
- [gdcmPythonFilter.h](#)

25.213 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](#) > [GetHierarchicalSearchTags](#) (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

25.213.1 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.

25.213.2 Constructor & Destructor Documentation

25.213.2.1 `virtual gdcm::QueryBase::~QueryBase () [inline],[virtual]`

25.213.3 Member Function Documentation

25.213.3.1 `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

25.213.3.2 `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

25.213.3.3 `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](#).

25.213.3.4 `virtual const char* gdcm::QueryBase::GetName () const [pure virtual]`

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

25.213.3.5 `virtual std::vector<Tag> gdcm::QueryBase::GetOptionalTags (const ERootType & inRootType) const [pure virtual]`

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

25.213.3.6 `virtual DataElement gdcm::QueryBase::GetQueryLevel () const [pure virtual]`

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

25.213.3.7 `virtual std::vector<Tag> gdcm::QueryBase::GetRequiredTags (const ERootType & inRootType) const [pure virtual]`

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

25.213.3.8 `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](#)

25.214 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](#) inQueryLevel)

25.214.1 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

25.214.2 Member Function Documentation

25.214.2.1 `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()).

25.214.2.2 `static void gdcM::QueryFactory::ListCharSets (std::ostream & os)` [static]

List all possible CharSet.

25.214.2.3 `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

25.214.2.4 `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:

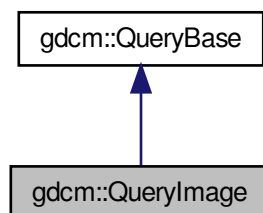
- [gdcmQueryFactory.h](#)

25.215 gdcm::QueryImage Class Reference

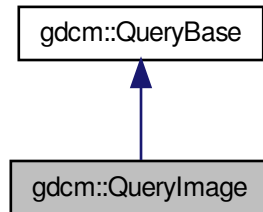
[QueryImage](#) contains: class to construct an image-based query for C-FIND and C-MOVE.

```
#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`

25.215.1 Detailed Description

[QueryImage](#) contains: class to construct an image-based query for C-FIND and C-MOVE.

25.215.2 Member Function Documentation

25.215.2.1 `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](#).

25.215.2.2 `const char* gdcm::QueryImage::GetName () const` `[virtual]`

Implements [gdcm::QueryBase](#).

25.215.2.3 `std::vector<Tag> gdcm::QueryImage::GetOptionalTags (const ERootType & inRootType) const` `[virtual]`

Implements [gdcm::QueryBase](#).

25.215.2.4 `DataElement gdcm::QueryImage::GetQueryLevel () const` [virtual]

Implements [gdcm::QueryBase](#).

25.215.2.5 `std::vector<Tag> gdcm::QueryImage::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

25.215.2.6 `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:

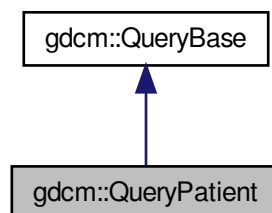
- [gdcmQueryImage.h](#)

25.216 gdcm::QueryPatient Class Reference

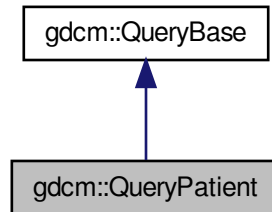
[QueryPatient](#) contains: class to construct a patient-based query for c-find and c-move.

```
#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`

25.216.1 Detailed Description

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

25.216.2 Member Function Documentation

25.216.2.1 `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`.

25.216.2.2 `const char* gdcm::QueryPatient::GetName () const` `[virtual]`

Implements `gdcm::QueryBase`.

25.216.2.3 `std::vector<Tag> gdcm::QueryPatient::GetOptionalTags (const ERootType & inRootType) const` `[virtual]`

Implements `gdcm::QueryBase`.

25.216.2.4 `DataElement` `gdcm::QueryPatient::GetQueryLevel () const` [virtual]

Implements [gdcm::QueryBase](#).

25.216.2.5 `std::vector<Tag>` `gdcm::QueryPatient::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

25.216.2.6 `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:

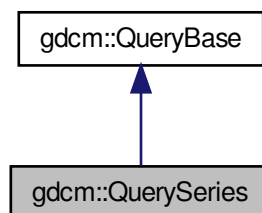
- [gdcmQueryPatient.h](#)

25.217 gdcm::QuerySeries Class Reference

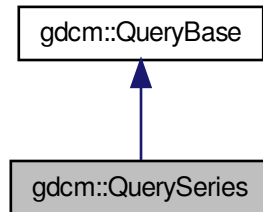
[QuerySeries](#) contains: class to construct a series-based query for c-find and c-move.

```
#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`

25.217.1 Detailed Description

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

25.217.2 Member Function Documentation

25.217.2.1 `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`.

25.217.2.2 `const char* gdcm::QuerySeries::GetName () const` `[virtual]`

Implements `gdcm::QueryBase`.

25.217.2.3 `std::vector<Tag> gdcm::QuerySeries::GetOptionalTags (const ERootType & inRootType) const` `[virtual]`

Implements `gdcm::QueryBase`.

25.217.2.4 `DataElement` `gdcm::QuerySeries::GetQueryLevel () const` [virtual]

Implements [gdcm::QueryBase](#).

25.217.2.5 `std::vector<Tag>` `gdcm::QuerySeries::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

25.217.2.6 `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:

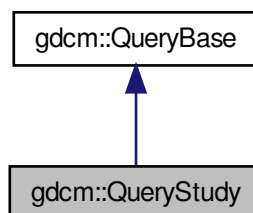
- [gdcmQuerySeries.h](#)

25.218 gdcm::QueryStudy Class Reference

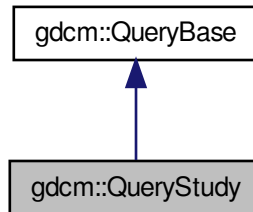
QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

```
#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`

25.218.1 Detailed Description

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

25.218.2 Member Function Documentation

25.218.2.1 `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`.

25.218.2.2 `const char* gdcm::QueryStudy::GetName () const` `[virtual]`

Implements `gdcm::QueryBase`.

25.218.2.3 `std::vector<Tag> gdcm::QueryStudy::GetOptionalTags (const ERootType & inRootType) const` `[virtual]`

Implements `gdcm::QueryBase`.

25.218.2.4 `DataElement gdcm::QueryStudy::GetQueryLevel () const` [virtual]

Implements [gdcm::QueryBase](#).

25.218.2.5 `std::vector<Tag> gdcm::QueryStudy::GetRequiredTags (const ERootType & inRootType) const` [virtual]

Implements [gdcm::QueryBase](#).

25.218.2.6 `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:

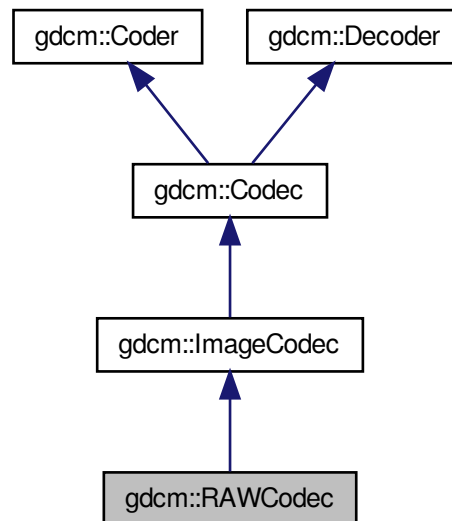
- [gdcmQueryStudy.h](#)

25.219 gdcm::RAWCodec Class Reference

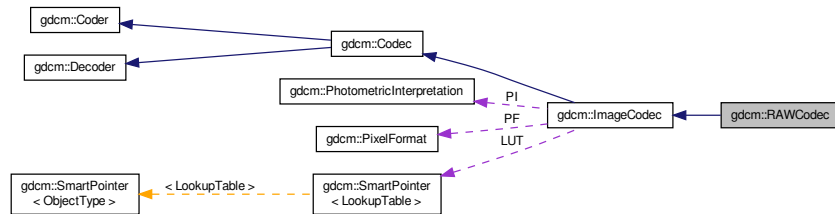
[RAWCodec](#) class.

```
#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)
- 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

25.219.1 Detailed Description

[RAWCodec](#) class.

25.219.2 Constructor & Destructor Documentation

25.219.2.1 `gdcm::RAWCodec::RAWCodec ()`

25.219.2.2 `gdcm::RAWCodec::~~RAWCodec ()`

25.219.3 Member Function Documentation

25.219.3.1 `bool gdcm::RAWCodec::CanCode (TransferSyntax const &) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.219.3.2 `bool gdcm::RAWCodec::CanDecode (TransferSyntax const &) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.219.3.3 `bool gdcm::RAWCodec::Code (DataElement const & in_, DataElement & out_)` [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

25.219.3.4 `bool gdcm::RAWCodec::Decode (DataElement const &, DataElement &)` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

25.219.3.5 `bool gdcm::RAWCodec::DecodeByStreams (std::istream & is, std::ostream & os)` [protected], [virtual]

Reimplemented from [gdcm::ImageCodec](#).

25.219.3.6 `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.

25.219.3.7 `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:

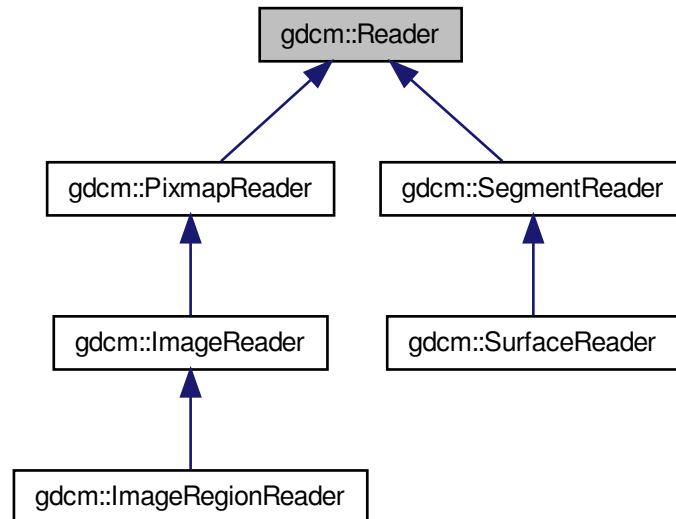
- [gdcmRAWCodec.h](#)

25.220 gdcm::Reader Class Reference

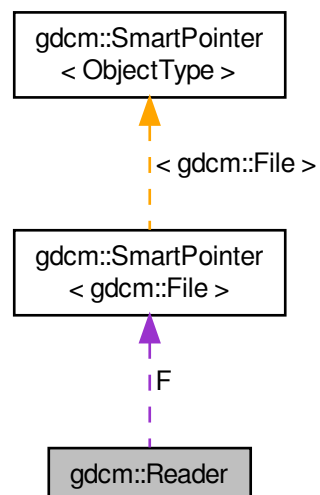
[Reader](#) ala DOM (Document [Object](#) Model)

```
#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.
- virtual bool [Read](#) ()
Main function to read a file.
- bool [ReadSelectedTags](#) (std::set< [Tag](#) > const &tags)
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](#)

25.220.1 Detailed Description

[Reader](#) ala DOM (Document [Object](#) Model)

This class is a non-validating reader, it will only performs well- formedness check only, and to some extent catch known error (non well-formed document).

Detailed description here

A [DataSet](#) DOES NOT contains group 0x0002 (see [FileMetaInformation](#))

This is really a [DataSet](#) reader. This will not make sure the dataset conform to any [IOD](#) at all. This is a completely different step. The reasoning was that user could control the [IOD](#) there lib would handle and thus we would not be able to read a [DataSet](#) if the [IOD](#) was not found Instead we separate the reading from the validation.

Note

From GDCM1.x. Users will realize that one feature is missing from this DOM implementation. In GDCM 1.x user used to be able to control the size of the [Value](#) to be read. By default it was 0xffff. 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 unordered (non-alphabetical order).

See Also

[Writer FileMetaInformation DataSet File](#)

Examples:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumplImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), [ScanDirectory.java](#), [SimplePrintPatientName.cs](#), and [TestReader.cxx](#).

25.220.2 Constructor & Destructor Documentation

25.220.2.1 `gdcmm::Reader::Reader () [inline]`

25.220.2.2 `virtual gdcmm::Reader::~~Reader () [virtual]`

25.220.3 Member Function Documentation

25.220.3.1 `bool gdcmm::Reader::CanRead () const`

Test whether this is a DICOM file

Warning

need to call either `SetFileName` or `SetStream` first

Examples:

[ReadUTF8QtDir.cxx](#).

25.220.3.2 `const File& gdcmm::Reader::GetFile () const [inline]`

Set/Get [File](#).

Examples:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumplImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), [ScanDirectory.java](#), [SimplePrintPatientName.cs](#), and [TestReader.cxx](#).

[cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

25.220.3.3 **File**& `gdcm::Reader::GetFile ()` `[inline]`

Set/Get [File](#).

25.220.3.4 `std::istream*` `gdcm::Reader::GetStreamPtr () const` `[inline]`, `[protected]`

25.220.3.5 `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:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

25.220.3.6 `bool` `gdcm::Reader::ReadDataSet ()` `[protected]`

25.220.3.7 `bool` `gdcm::Reader::ReadMetaInformation ()` `[protected]`

25.220.3.8 `bool` `gdcm::Reader::ReadPreamble ()` `[protected]`

25.220.3.9 `bool` `gdcm::Reader::ReadSelectedTags (std::set< Tag > const & tags)`

Will only read the specified selected tags.

25.220.3.10 `bool` `gdcm::Reader::ReadUpToTag (const Tag & tag, std::set< Tag > const & skiptags = std::set< Tag > ())`

Will read only up to [Tag](#)

Parameters

<i>tag</i>	and skipping any tag specified in
<i>skiptags</i>	

25.220.3.11 `void` `gdcm::Reader::SetFile (File & file)` `[inline]`

Set/Get [File](#).

25.220.3.12 void gdcmm::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:

[ChangeSequenceUltrasound.cxx](#), [CheckBigEndianBug.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadMultiTimesException.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), [TestReader.cxx](#), and [threadgdcmm.cxx](#).

25.220.3.13 void gdcmm::Reader::SetStream (std::istream & *input_stream*) [inline]

Set the open-ed stream directly.

Examples:

[ReadUTF8QtDir.cxx](#).

25.220.4 Friends And Related Function Documentation

25.220.4.1 friend class StreamImageReader [friend]

25.220.5 Member Data Documentation

25.220.5.1 SmartPointer<File> gdcmm::Reader::F [protected]

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

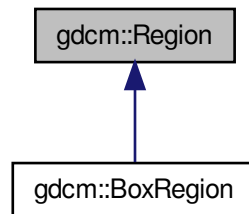
- [gdcmmReader.h](#)

25.221 gdcmm::Region Class Reference

Class for manipulation region.

```
#include <gdcmmRegion.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.

25.221.1 Detailed Description

Class for manipulation region.

25.221.2 Constructor & Destructor Documentation

25.221.2.1 `gdcm::Region::Region ()`

25.221.2.2 `virtual gdcm::Region::~~Region ()` [virtual]

25.221.3 Member Function Documentation

25.221.3.1 `virtual size_t gdcm::Region::Area () const` [pure virtual]

compute the area

Implemented in [gdcm::BoxRegion](#).

25.221.3.2 `virtual Region* gdcm::Region::Clone () const` [pure virtual]

Implemented in [gdcm::BoxRegion](#).

25.221.3.3 `virtual BoxRegion gdcm::Region::ComputeBoundingBox ()` [pure virtual]

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in [gdcm::BoxRegion](#).

25.221.3.4 `virtual bool gdcm::Region::Empty () const` [pure virtual]

return whether this domain is empty:

Implemented in [gdcm::BoxRegion](#).

25.221.3.5 `virtual bool gdcm::Region::IsValid () const` [pure virtual]

return whether this is valid domain

Implemented in [gdcm::BoxRegion](#).

25.221.3.6 `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](#)

25.222 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 ComputeInterceptSlopePixelFormat \(\)](#)

- [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 [SetMinMaxForPixelFormat](#) (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 [SetTargetPixelFormat](#) ([PixelFormat](#) const &targetst)
- void [SetUseTargetPixelFormat](#) (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)

25.222.1 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.SetMinMaxForPixelFormat( ((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](#)

25.222.2 Constructor & Destructor Documentation

25.222.2.1 `gdcm::Rescaler::Rescaler ()` `[inline]`

25.222.2.2 `gdcm::Rescaler::~~Rescaler ()` `[inline]`

25.222.3 Member Function Documentation

25.222.3.1 `PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelType ()`

Compute the Pixel Format of the output data Used for direct transformation

25.222.3.2 `PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ()`

Compute the Pixel Format of the output data Used for inverse transformation

25.222.3.3 `double gdcm::Rescaler::GetIntercept () const` `[inline]`

25.222.3.4 `double gdcm::Rescaler::GetSlope () const` `[inline]`

25.222.3.5 `bool gdcm::Rescaler::InverseRescale (char * out, const char * in, size_t n)`

Inverse transform.

25.222.3.6 `template<typename TIn > void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit (char * out, const TIn * in, size_t n)` `[protected]`

25.222.3.7 `bool gdcm::Rescaler::Rescale (char * out, const char * in, size_t n)`

Direct transform.

25.222.3.8 `template<typename TIn > void gdcm::Rescaler::RescaleFunctionIntoBestFit (char * out, const TIn * in, size_t n)` `[protected]`

25.222.3.9 `void gdcm::Rescaler::SetIntercept (double i)` `[inline]`

Set Intercept: used for both direct&inverse transformation.

25.222.3.10 `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

25.222.3.11 void gdcm::Rescaler::SetPixelFormat (PixelFormat const & *pf*) [inline]

Set Pixel Format of input data.

25.222.3.12 void gdcm::Rescaler::SetSlope (double *s*) [inline]

Set Slope: user for both direct&inverse transformation.

25.222.3.13 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](#)

25.222.3.14 void gdcm::Rescaler::SetUseTargetPixelType (bool *b*)

Override default behavior of Rescale.

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

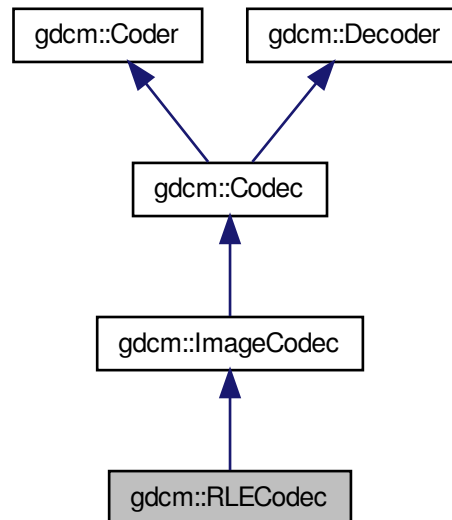
- [gdcmRescaler.h](#)

25.223 gdcm::RLECodec Class Reference

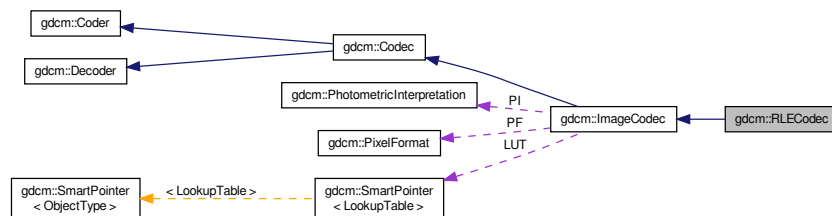
Class to do RLE.

```
#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)
- 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 [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)

Friends

- class [ImageRegionReader](#)

Additional Inherited Members

25.223.1 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).

25.223.2 Constructor & Destructor Documentation

25.223.2.1 `gdcm::RLECodec::RLECodec ()`

25.223.2.2 `gdcm::RLECodec::~~RLECodec ()`

25.223.3 Member Function Documentation

25.223.3.1 `bool gdcm::RLECodec::CanCode (TransferSyntax const &) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

25.223.3.2 `bool gdcm::RLECodec::CanDecode (TransferSyntax const &) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

25.223.3.3 `bool gdcM::RLECodec::Code (DataElement const & in_, DataElement & out_) [virtual]`

Code.

Reimplemented from [gdcM::Coder](#).

25.223.3.4 `bool gdcM::RLECodec::Decode (DataElement const &, DataElement &) [virtual]`

Decode.

Reimplemented from [gdcM::ImageCodec](#).

25.223.3.5 `bool gdcM::RLECodec::DecodeByStreams (std::istream & is, std::ostream & os) [protected],[virtual]`

Reimplemented from [gdcM::ImageCodec](#).

25.223.3.6 `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]`

25.223.3.7 `unsigned long gdcM::RLECodec::GetBufferLength () const [inline]`

25.223.3.8 `bool gdcM::RLECodec::GetHeaderInfo (std::istream & is, TransferSyntax & ts) [virtual]`

Reimplemented from [gdcM::ImageCodec](#).

25.223.3.9 `void gdcM::RLECodec::SetBufferLength (unsigned long l) [inline]`

25.223.3.10 `void gdcM::RLECodec::SetLength (unsigned long l) [inline]`

25.223.4 Friends And Related Function Documentation

25.223.4.1 `friend class ImageRegionReader [friend]`

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

- [gdcMRLECodec.h](#)

25.224 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`

25.224.1 Detailed Description

[RoleSelectionSub](#) PS 3.7 [Table D.3-9](#) SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

25.224.2 Constructor & Destructor Documentation

25.224.2.1 `gdcm::network::RoleSelectionSub::RoleSelectionSub ()`

25.224.3 Member Function Documentation

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

25.224.3.2 `std::istream& gdcm::network::RoleSelectionSub::Read (std::istream & is)`

25.224.3.3 `void gdcm::network::RoleSelectionSub::SetTuple (const char * uid, uint8_t scurole, uint8_t scprole)`

25.224.3.4 `size_t gdcm::network::RoleSelectionSub::Size () const`

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

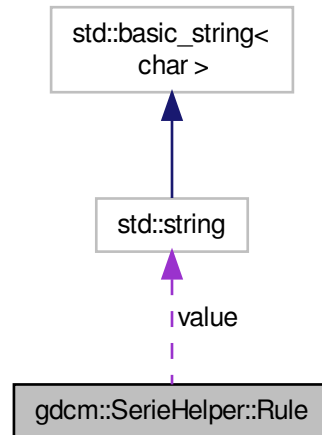
The documentation for this class was generated from the following file:

- [gdcmRoleSelectionSub.h](#)

25.225 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](#)

25.225.1 Member Data Documentation

25.225.1.1 `uint16_t` `gdcm::SerieHelper::Rule::elem`

25.225.1.2 `uint16_t` `gdcm::SerieHelper::Rule::group`

25.225.1.3 `int` `gdcm::SerieHelper::Rule::op`

25.225.1.4 `std::string` `gdcm::SerieHelper::Rule::value`

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

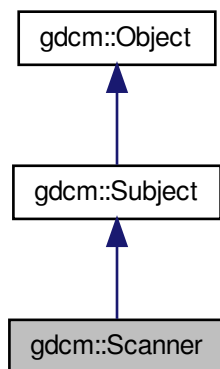
- [gdcmSerieHelper.h](#)

25.226 gdcm::Scanner Class Reference

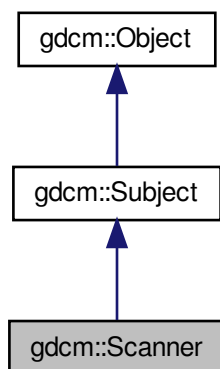
[Scanner](#) This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).


```
#include <gdcmScanner.h>
```

Inheritance diagram for gdcm::Scanner:



Collaboration diagram for gdcm::Scanner:



Classes

- struct [ltstr](#)

Public Types

- typedef MappingType::const_iterator [ConstIterator](#)

- typedef std::map< const char *, [TagToValue](#), [ltstr](#) > [MappingType](#)
- typedef std::map< [Tag](#), const char * > [TagToValue](#)
- typedef [TagToValue](#)::value_type [TagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

Public Member Functions

- [Scanner](#) ()
- [~Scanner](#) ()
- 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: instantiate a reference counted object

Protected Member Functions

- void [ProcessPublicTag](#) ([StringFilter](#) &sf, const char *filename)

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [Scanner](#) &s)

25.226.1 Detailed Description

[Scanner](#) This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

This filter is dealing with both VRASCII and VRBINARY element, thanks to the help of [gdcm::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](#).

25.226.2 Member Typedef Documentation

25.226.2.1 `typedef MappingType::const_iterator gdcm::Scanner::ConstIterator`

25.226.2.2 `typedef std::map<const char *, TagToValue, Itstr> gdcm::Scanner::MappingType`

25.226.2.3 `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)

25.226.2.4 `typedef TagToValue::value_type gdcmm::Scanner::TagToValueValueType`

25.226.2.5 `typedef std::set< std::string > gdcmm::Scanner::ValuesType`

25.226.3 Constructor & Destructor Documentation

25.226.3.1 `gdcmm::Scanner::Scanner ()` `[inline]`

25.226.3.2 `gdcmm::Scanner::~~Scanner ()`

25.226.4 Member Function Documentation

25.226.4.1 `void gdcmm::Scanner::AddPrivateTag (PrivateTag const & t)`

25.226.4.2 `void gdcmm::Scanner::AddSkipTag (Tag const & t)`

Add a tag that will need to be skipped. Those are root level skip tags.

25.226.4.3 `void gdcmm::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](#).

25.226.4.4 `ConstIterator gdcmm::Scanner::Begin () const` `[inline]`

25.226.4.5 `void gdcmm::Scanner::ClearSkipTags ()`

25.226.4.6 `void gdcmm::Scanner::ClearTags ()`

25.226.4.7 `ConstIterator gdcmm::Scanner::End () const` `[inline]`

25.226.4.8 `Directory::FileNamesType gdcmm::Scanner::GetAllFileNamesFromTagToValue (Tag const & t, const char * valuref) const`

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valuref'

25.226.4.9 `const char* gdcmm::Scanner::GetFilenameFromTagToValue (Tag const & t, const char * valuref) const`

Will loop over all files and return the first file where value match the reference value 'valuref'

25.226.4.10 `Directory::FileNamesType const& gdcmm::Scanner::GetFileNames () const` `[inline]`

25.226.4.11 `Directory::FileNamesType gdcmm::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](#).

25.226.4.12 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](#).

25.226.4.13 TagToValue const& gdcm::Scanner::GetMappingFromTagToValue (Tag const & *t*, const char * *value*) const

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

25.226.4.14 MappingType const& gdcm::Scanner::GetMappings () const [inline]

Mappings are the mapping from a particular tag to the map, mapping filename to value:

25.226.4.15 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)

25.226.4.16 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 !

25.226.4.17 ValuesType const& gdcm::Scanner::GetValues () const [inline]

Get all the values found (in lexicographic order)

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

25.226.4.18 ValuesType gdcm::Scanner::GetValues (Tag const & *t*) const

Get all the values found (in lexicographic order) associated with Tag 't'.

25.226.4.19 `bool gdcM::Scanner::IsKey (const char * filename) const`

Check if filename is a key in the Mapping table. returns true only if file can be found, which means the file was indeed a DICOM file that could be processed

Examples:

[DumpToSQLITE3.cxx](#), and [SimpleScanner.cxx](#).

25.226.4.20 `static SmartPointer<Scanner> gdcM::Scanner::New () [inline],[static]`

for wrapped language: instantiate a reference counted object

25.226.4.21 `void gdcM::Scanner::Print (std::ostream & os) const [virtual]`

Print result.

Reimplemented from [gdcM::Object](#).

Referenced by `gdcM::operator<<()`.

25.226.4.22 `void gdcM::Scanner::ProcessPublicTag (StringFilter & sf, const char * filename) [protected]`

25.226.4.23 `bool gdcM::Scanner::Scan (Directory::FileNamesType const & filenames)`

Start the scan !

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

25.226.5 Friends And Related Function Documentation

25.226.5.1 `std::ostream& operator<< (std::ostream & _os, const Scanner & s) [friend]`

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

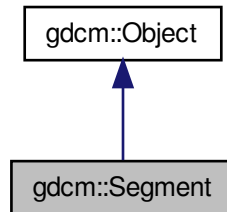
- [gdcMScanner.h](#)

25.227 gdcM::Segment Class Reference

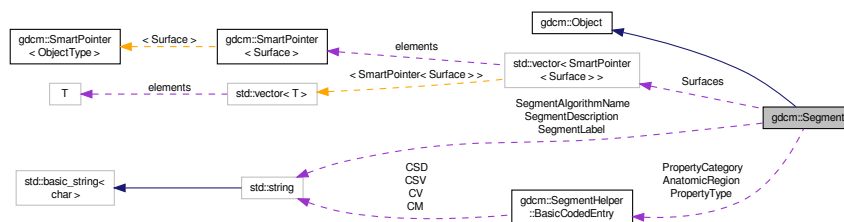
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

```
#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

25.227.1 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

25.227.2 Member Typedef Documentation

25.227.2.1 `typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector`

25.227.3 Member Enumeration Documentation

25.227.3.1 `enum gdcm::Segment::ALGOType`

Enumerator

MANUAL

AUTOMATIC

ALGOType_END

25.227.4 Constructor & Destructor Documentation

25.227.4.1 `gdcm::Segment::Segment ()`

25.227.4.2 `virtual gdcm::Segment::~~Segment () [virtual]`

25.227.5 Member Function Documentation

25.227.5.1 `void gdcm::Segment::AddSurface (SmartPointer< Surface > surface)`

25.227.5.2 `static ALGOType gdcm::Segment::GetALGOType (const char * type) [static]`

25.227.5.3 `static const char* gdcm::Segment::GetALGOTypeString (ALGOType type) [static]`

25.227.5.4 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion () const`

25.227.5.5 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ()`

25.227.5.6 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory () const`

25.227.5.7 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ()`

25.227.5.8 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType () const`

25.227.5.9 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ()`

25.227.5.10 `const char* gdcm::Segment::GetSegmentAlgorithmName () const`

25.227.5.11 `ALGOType gdcm::Segment::GetSegmentAlgorithmType () const`

25.227.5.12 `const char* gdcm::Segment::GetSegmentDescription () const`

25.227.5.13 `const char* gdcm::Segment::GetSegmentLabel () const`

25.227.5.14 `unsigned short gdcm::Segment::GetSegmentNumber () const`

25.227.5.15 `SmartPointer< Surface > gdcm::Segment::GetSurface (const unsigned int idx = 0) const`

- 25.227.5.16 `unsigned long gdcM::Segment::GetSurfaceCount ()`
- 25.227.5.17 `SurfaceVector const& gdcM::Segment::GetSurfaces () const`
- 25.227.5.18 `SurfaceVector& gdcM::Segment::GetSurfaces ()`
- 25.227.5.19 `void gdcM::Segment::SetAnatomicRegion (SegmentHelper::BasicCodedEntry const & BSE)`
- 25.227.5.20 `void gdcM::Segment::SetPropertyCategory (SegmentHelper::BasicCodedEntry const & BSE)`
- 25.227.5.21 `void gdcM::Segment::SetPropertyType (SegmentHelper::BasicCodedEntry const & BSE)`
- 25.227.5.22 `void gdcM::Segment::SetSegmentAlgorithmName (const char * name)`
- 25.227.5.23 `void gdcM::Segment::SetSegmentAlgorithmType (ALGOType type)`
- 25.227.5.24 `void gdcM::Segment::SetSegmentAlgorithmType (const char * typeStr)`
- 25.227.5.25 `void gdcM::Segment::SetSegmentDescription (const char * description)`
- 25.227.5.26 `void gdcM::Segment::SetSegmentLabel (const char * label)`
- 25.227.5.27 `void gdcM::Segment::SetSegmentNumber (const unsigned short num)`
- 25.227.5.28 `void gdcM::Segment::SetSurfaceCount (const unsigned long nb)`

25.227.6 Member Data Documentation

- 25.227.6.1 `SegmentHelper::BasicCodedEntry gdcM::Segment::AnatomicRegion` [protected]
- 25.227.6.2 `SegmentHelper::BasicCodedEntry gdcM::Segment::PropertyCategory` [protected]
- 25.227.6.3 `SegmentHelper::BasicCodedEntry gdcM::Segment::PropertyType` [protected]
- 25.227.6.4 `std::string gdcM::Segment::SegmentAlgorithmName` [protected]
- 25.227.6.5 `ALGOType gdcM::Segment::SegmentAlgorithmType` [protected]
- 25.227.6.6 `std::string gdcM::Segment::SegmentDescription` [protected]
- 25.227.6.7 `std::string gdcM::Segment::SegmentLabel` [protected]
- 25.227.6.8 `unsigned short gdcM::Segment::SegmentNumber` [protected]
- 25.227.6.9 `unsigned long gdcM::Segment::SurfaceCount` [protected]
- 25.227.6.10 `SurfaceVector gdcM::Segment::Surfaces` [protected]

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

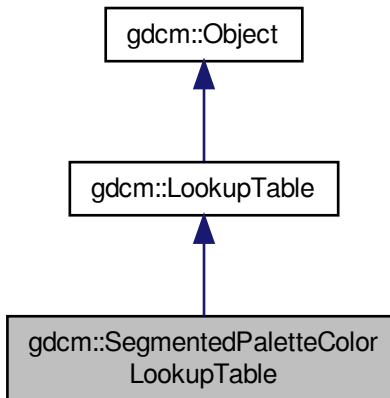
- [gdcMSegment.h](#)

25.228 gdcm::SegmentedPaletteColorLookupTable Class Reference

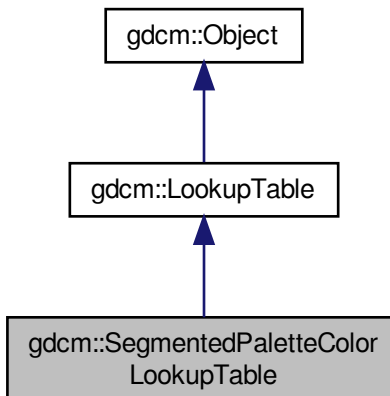
[SegmentedPaletteColorLookupTable](#) class.

```
#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

25.228.1 Detailed Description

[SegmentedPaletteColorLookupTable](#) class.

25.228.2 Constructor & Destructor Documentation

25.228.2.1 `gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ()`

25.228.2.2 `gdcm::SegmentedPaletteColorLookupTable::~~SegmentedPaletteColorLookupTable ()`

25.228.3 Member Function Documentation

25.228.3.1 `void gdcm::SegmentedPaletteColorLookupTable::Print (std::ostream &) const` `[inline]`, `[virtual]`

Reimplemented from [gdcm::LookupTable](#).

25.228.3.2 `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:

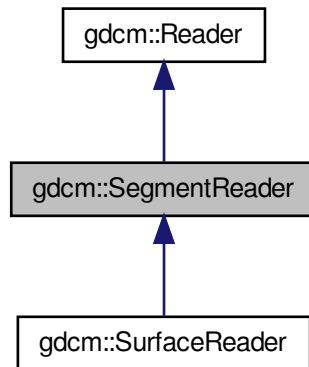
- [gdcmSegmentedPaletteColorLookupTable.h](#)

25.229 gdcm::SegmentReader Class Reference

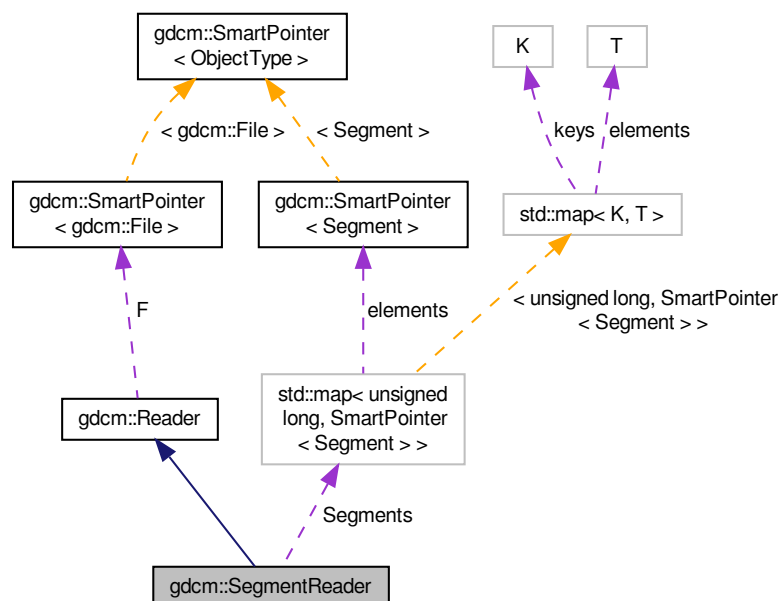
This class defines a segment reader. It reads attributes of group 0x0062.

```
#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](#)

25.229.1 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

25.229.2 Member Typedef Documentation

25.229.2.1 typedef std::map< unsigned long, [SmartPointer](#)< [Segment](#) > > [gdcm::SegmentReader::SegmentMap](#)
[protected]

25.229.2.2 typedef std::vector< [SmartPointer](#)< [Segment](#) > > [gdcm::SegmentReader::SegmentVector](#)

25.229.3 Constructor & Destructor Documentation

25.229.3.1 [gdcm::SegmentReader::SegmentReader](#) ()

25.229.3.2 virtual [gdcm::SegmentReader::~~SegmentReader](#) () [virtual]

25.229.4 Member Function Documentation

25.229.4.1 `const SegmentVector gdcm::SegmentReader::GetSegments () const`

25.229.4.2 `SegmentVector gdcm::SegmentReader::GetSegments ()`

25.229.4.3 `virtual bool gdcm::SegmentReader::Read () [virtual]`

Read.

Reimplemented from [gdcm::Reader](#).

Reimplemented in [gdcm::SurfaceReader](#).

25.229.4.4 `bool gdcm::SegmentReader::ReadSegment (const Item & segmentItem, const unsigned int idx) [protected]`

25.229.4.5 `bool gdcm::SegmentReader::ReadSegments () [protected]`

25.229.5 Member Data Documentation

25.229.5.1 `SegmentMap gdcm::SegmentReader::Segments [protected]`

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

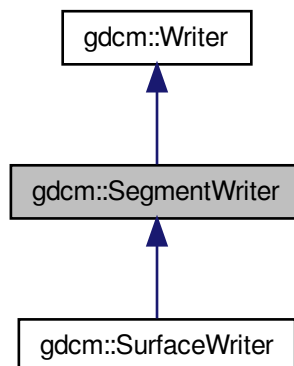
- [gdcmSegmentReader.h](#)

25.230 gdcm::SegmentWriter Class Reference

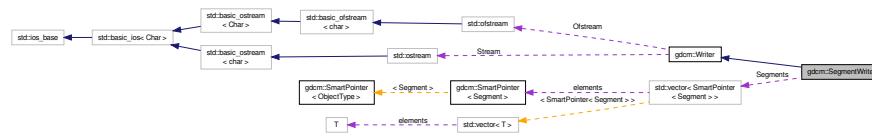
This class defines a segment writer. It writes attributes of group 0x0062.

```
#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](#)

25.230.1 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

25.230.2 Member Typedef Documentation

25.230.2.1 `typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector`

25.230.3 Constructor & Destructor Documentation

25.230.3.1 `gdcm::SegmentWriter::SegmentWriter ()`

25.230.3.2 `virtual gdcm::SegmentWriter::~~SegmentWriter () [virtual]`

25.230.4 Member Function Documentation

25.230.4.1 `void gdcm::SegmentWriter::AddSegment (SmartPointer< Segment > segment)`

25.230.4.2 `unsigned int gdcm::SegmentWriter::GetNumberOfSegments () const`

25.230.4.3 `SmartPointer< Segment > gdcm::SegmentWriter::GetSegment (const unsigned int idx = 0) const`

25.230.4.4 `const SegmentVector& gdcm::SegmentWriter::GetSegments () const`

25.230.4.5 `SegmentVector& gdcm::SegmentWriter::GetSegments ()`

25.230.4.6 `bool gdcm::SegmentWriter::PrepareWrite () [protected]`

25.230.4.7 `void gdcm::SegmentWriter::SetNumberOfSegments (const unsigned int size)`

25.230.4.8 `void gdcm::SegmentWriter::SetSegments (SegmentVector & segments)`

25.230.4.9 `bool gdcm::SegmentWriter::Write () [virtual]`

Write.

Reimplemented from [gdcm::Writer](#).

Reimplemented in [gdcm::SurfaceWriter](#).

25.230.5 Member Data Documentation

25.230.5.1 `SegmentVector gdcm::SegmentWriter::Segments [protected]`

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

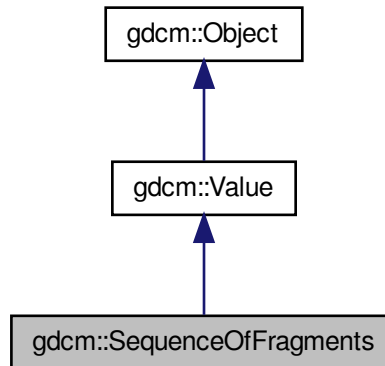
- [gdcmSegmentWriter.h](#)

25.231 gdcm::SequenceOfFragments Class Reference

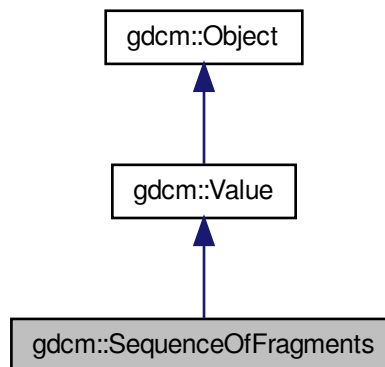
Class to represent a Sequence Of Fragments.

```
#include <gdcmSequenceOfFragments.h>
```

Inheritance diagram for `gdc::SequenceOfFragments`:



Collaboration diagram for `gdc::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)
- template<typename TSwap >
std::istream & [ReadPreValue](#) (std::istream &is)
- template<typename TSwap >
std::istream & [ReadValue](#) (std::istream &is)
- 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

25.231.1 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](#).

25.231.2 Member Typedef Documentation

25.231.2.1 `typedef FragmentVector::const_iterator gdcmm::SequenceOfFragments::ConstIterator`

25.231.2.2 `typedef std::vector<Fragment> gdcmm::SequenceOfFragments::FragmentVector`

25.231.2.3 `typedef FragmentVector::iterator gdcmm::SequenceOfFragments::Iterator`

25.231.2.4 `typedef FragmentVector::size_type gdcmm::SequenceOfFragments::SizeType`

25.231.3 Constructor & Destructor Documentation

25.231.3.1 `gdcmm::SequenceOfFragments::SequenceOfFragments () [inline]`

constructor (UndefinedLength by default)

25.231.4 Member Function Documentation

25.231.4.1 `void gdcmm::SequenceOfFragments::AddFragment (Fragment const & item)`

Appends a [Fragment](#) to the already added ones.

Examples:

[FixBrokenJ2K.cxx](#).

25.231.4.2 `Iterator gdcmm::SequenceOfFragments::Begin () [inline]`

25.231.4.3 `ConstIterator gdcmm::SequenceOfFragments::Begin () const [inline]`

25.231.4.4 `void gdcmm::SequenceOfFragments::Clear () [virtual]`

Clear.

Implements [gdcmm::Value](#).

25.231.4.5 `unsigned long gdcmm::SequenceOfFragments::ComputeByteLength () const`

25.231.4.6 `VL gdcmm::SequenceOfFragments::ComputeLength () const`

25.231.4.7 `Iterator gdcmm::SequenceOfFragments::End () [inline]`

25.231.4.8 `ConstIterator gdcmm::SequenceOfFragments::End () const [inline]`

25.231.4.9 `bool gdcmm::SequenceOfFragments::GetBuffer (char * buffer, unsigned long length) const`

25.231.4.10 `bool gdcmm::SequenceOfFragments::GetFragBuffer (unsigned int fragNb, char * buffer, unsigned long & length) const`

25.231.4.11 `const Fragment& gdcm::SequenceOfFragments::GetFragment (SizeType num) const`

Examples:

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGs.cxx](#).

25.231.4.12 `VL gdcm::SequenceOfFragments::GetLength () const [inline],[virtual]`

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

25.231.4.13 `SizeType gdcm::SequenceOfFragments::GetNumberOfFragments () const`

Examples:

[FixJAIBugJPEGs.cxx](#).

25.231.4.14 `const BasicOffsetTable& gdcm::SequenceOfFragments::GetTable () const [inline]`

25.231.4.15 `BasicOffsetTable& gdcm::SequenceOfFragments::GetTable () [inline]`

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

25.231.4.17 `bool gdcm::SequenceOfFragments::operator== (const Value & val) const [inline],[virtual]`

Implements [gdcm::Value](#).

25.231.4.18 `void gdcm::SequenceOfFragments::Print (std::ostream & os) const [inline],[virtual]`

Reimplemented from [gdcm::Object](#).

25.231.4.19 `template<typename TSwap > std::istream& gdcm::SequenceOfFragments::Read (std::istream & is) [inline]`

25.231.4.20 `template<typename TSwap > std::istream& gdcm::SequenceOfFragments::ReadPreValue (std::istream & is)
[inline]`

References [gdcmDebugMacro](#), and [gdcm::DataElement::SetByteValue\(\)](#).

25.231.4.21 `template<typename TSwap > std::istream& gdcm::SequenceOfFragments::ReadValue (std::istream & is)
[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\(\)](#).

25.231.4.22 `void gdcm::SequenceOfFragments::SetLength (VL length) [inline],[virtual]`

Sets the actual SQ length.

Implements [gdcm::Value](#).

25.231.4.23 `template<typename TSwap > std::ostream const& gdcm::SequenceOfFragments::Write (std::ostream & os) const [inline]`

References [gdcm::VL::Write\(\)](#), and [gdcm::Tag::Write\(\)](#).

25.231.4.24 `bool gdcm::SequenceOfFragments::WriteBuffer (std::ostream & os) const`

Examples:

[GetJPEGSamplePrecision.cxx](#).

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

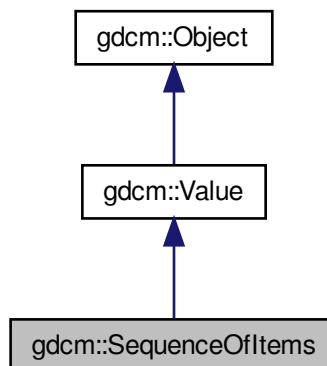
- [gdcmSequenceOfFragments.h](#)

25.232 gdcm::SequenceOfItems Class Reference

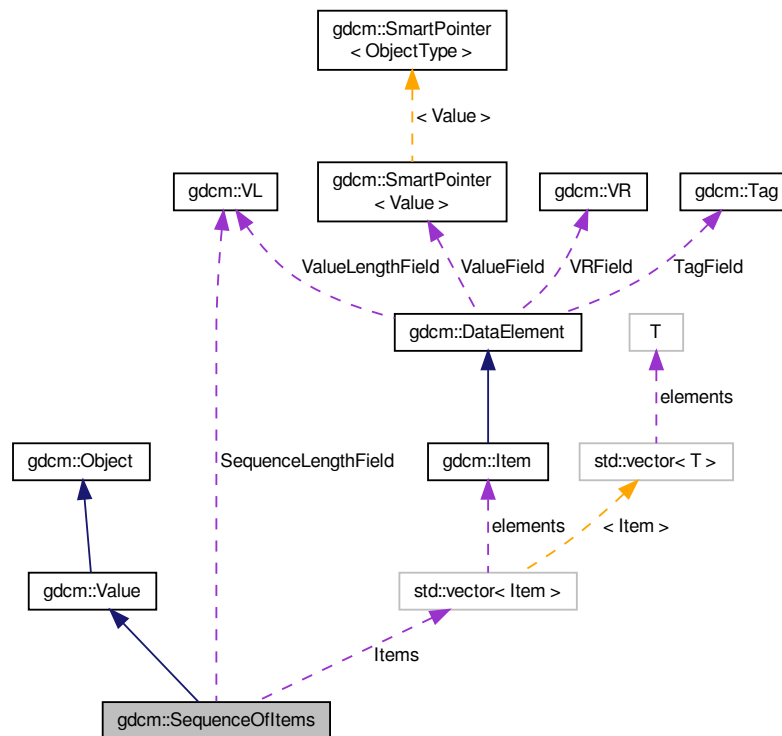
Class to represent a Sequence Of Items (value representation : SQ)

`#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](#) ()
- 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)
- 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

25.232.1 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:

[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](#).

25.232.2 Member Typedef Documentation

25.232.2.1 `typedef ItemVector::const_iterator gdcm::SequenceOfItems::ConstIterator`

25.232.2.2 `typedef std::vector< Item > gdcm::SequenceOfItems::ItemVector`

25.232.2.3 `typedef ItemVector::iterator gdcm::SequenceOfItems::Iterator`

25.232.2.4 `typedef ItemVector::size_type gdcm::SequenceOfItems::SizeType`

25.232.3 Constructor & Destructor Documentation

25.232.3.1 `gdcm::SequenceOfItems::SequenceOfItems () [inline]`

constructor (UndefinedLength by default)

25.232.4 Member Function Documentation

25.232.4.1 `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](#).

25.232.4.2 `Iterator gdcm::SequenceOfItems::Begin () [inline]`

25.232.4.3 `ConstIterator gdcm::SequenceOfItems::Begin () const [inline]`

25.232.4.4 `void gdcm::SequenceOfItems::Clear () [inline],[virtual]`

Implements [gdcm::Value](#).

25.232.4.5 `template<typename TDE > VL gdcm::SequenceOfItems::ComputeLength () const`

25.232.4.6 `Iterator gdcm::SequenceOfItems::End () [inline]`

25.232.4.7 **ConstIterator** `gdcmm::SequenceOfItems::End () const` `[inline]`

25.232.4.8 **bool** `gdcmm::SequenceOfItems::FindDataElement (const Tag & t) const`

25.232.4.9 **const Item&** `gdcmm::SequenceOfItems::GetItem (SizeType position) const`

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [LargeVRDSExplicit.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

25.232.4.10 **Item&** `gdcmm::SequenceOfItems::GetItem (SizeType position)`

25.232.4.11 **VL** `gdcmm::SequenceOfItems::GetLength () const` `[inline],[virtual]`

Returns the SQ length, as read from disk.

Implements [gdcmm::Value](#).

25.232.4.12 **SizeType** `gdcmm::SequenceOfItems::GetNumberOfItems () const` `[inline]`

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpGEMSMovieGroup.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), and [LargeVRDSExplicit.cxx](#).

25.232.4.13 **bool** `gdcmm::SequenceOfItems::IsUndefinedLength () const` `[inline]`

return if [Value](#) Length if of undefined length

25.232.4.14 **static SmartPointer<SequenceOfItems>** `gdcmm::SequenceOfItems::New ()` `[inline],[static]`

25.232.4.15 **SequenceOfItems&** `gdcmm::SequenceOfItems::operator= (const SequenceOfItems & val)` `[inline]`

References Items, and SequenceLengthField.

25.232.4.16 **bool** `gdcmm::SequenceOfItems::operator==(const Value & val) const` `[inline],[virtual]`

Implements [gdcmm::Value](#).

References Items, and SequenceLengthField.

25.232.4.17 **void** `gdcmm::SequenceOfItems::Print (std::ostream & os) const` `[inline],[virtual]`

Reimplemented from [gdcmm::Object](#).

25.232.4.18 `template<typename TDE , typename TSwap > std::istream& gdcmm::SequenceOfItems::Read (std::istream & is)`
`[inline]`

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

References `gdcmm::Item::Clear()`, `gdcmmDebugMacro`, `gdcmmWarningMacro`, `gdcmm::Exception::GetDescription()`, `gdcmm::Item::GetNestedDataSet()`, `gdcmm::DataElement::GetTag()`, `gdcmm::DataElement::GetVL()`, `gdcmm::Item::Read()`, and `gdcmm::DataSet::Size()`.

25.232.4.19 `void gdcmm::SequenceOfItems::SetLength (VL length)` `[inline]`, `[virtual]`

Sets the actual SQ length.

Implements [gdcmm::Value](#).

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

25.232.4.20 `void gdcmm::SequenceOfItems::SetLengthToUndefined ()`

Properly set the Sequence of [Item](#) to be undefined length.

Examples:

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllIVR.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

25.232.4.21 `void gdcmm::SequenceOfItems::SetNumberOfItems (SizeType n)` `[inline]`

25.232.4.22 `template<typename TDE , typename TSwap > std::ostream const& gdcmm::SequenceOfItems::Write (std::ostream & os) const` `[inline]`

References `gdcmm::VL::Write()`, and `gdcmm::Tag::Write()`.

25.232.5 Member Data Documentation

25.232.5.1 ItemVector `gdcmm::SequenceOfItems::Items`

Vector of Sequence Items.

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

25.232.5.2 VL `gdcmm::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:

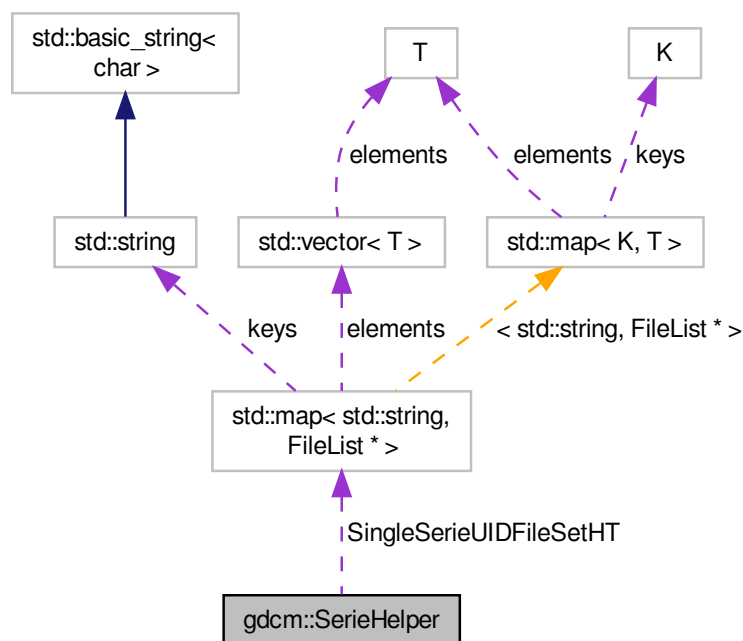
- [gdcmSequenceOfItems.h](#)

25.233 gdcm::SerieHelper Class Reference

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

```
#include <gdcmSerieHelper.h>
```

Collaboration diagram for gdcm::SerieHelper:



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`

25.233.1 Detailed Description

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

Instead see [gdcm::ImageHelper](#) or [gdcm::IPPSorter](#)

25.233.2 Member Typedef Documentation

25.233.2.1 `typedef std::vector<Rule> gdcm::SerieHelper::SerieRestrictions` [protected]

25.233.2.2 `typedef std::map<std::string, FileList *> gdcm::SerieHelper::SingleSerieUIDFileSetmap` [protected]

25.233.3 Constructor & Destructor Documentation

25.233.3.1 `gdcm::SerieHelper::SerieHelper ()`

25.233.3.2 `gdcm::SerieHelper::~~SerieHelper ()`

25.233.4 Member Function Documentation

- 25.233.4.1 `bool gdcM::SerieHelper::AddFile (FileWithName & header)` [protected]
- 25.233.4.2 `void gdcM::SerieHelper::AddFileName (std::string const & filename)` [protected]
- 25.233.4.3 `void gdcM::SerieHelper::AddRestriction (const std::string & tag)`
- 25.233.4.4 `void gdcM::SerieHelper::AddRestriction (uint16_t group, uint16_t elem, std::string const & value, int op)`
- 25.233.4.5 `void gdcM::SerieHelper::AddRestriction (const Tag & tag)` [protected]
- 25.233.4.6 `void gdcM::SerieHelper::Clear ()`
- 25.233.4.7 `void gdcM::SerieHelper::CreateDefaultUniqueSeriesIdentifier ()`
- 25.233.4.8 `std::string gdcM::SerieHelper::CreateUniqueSeriesIdentifier (File * inFile)`
- 25.233.4.9 `bool gdcM::SerieHelper::FileNameOrdering (FileList * fileList)` [protected]
- 25.233.4.10 `FileList* gdcM::SerieHelper::GetFirstSingleSerieUIDFileSet ()`
- 25.233.4.11 `FileList* gdcM::SerieHelper::GetNextSingleSerieUIDFileSet ()`
- 25.233.4.12 `bool gdcM::SerieHelper::ImagePositionPatientOrdering (FileList * fileSet)` [protected]
- 25.233.4.13 `void gdcM::SerieHelper::OrderFileList (FileList * fileSet)`
- 25.233.4.14 `void gdcM::SerieHelper::SetDirectory (std::string const & dir, bool recursive = false)`
- 25.233.4.15 `void gdcM::SerieHelper::SetLoadMode (int)` [inline]
- 25.233.4.16 `void gdcM::SerieHelper::SetUseSeriesDetails (bool useSeriesDetails)`
- 25.233.4.17 `bool gdcM::SerieHelper::UserOrdering (FileList * fileSet)` [protected]

25.233.5 Member Data Documentation

- 25.233.5.1 `SingleSerieUIDFileSetmap::iterator gdcM::SerieHelper::ItFileSetHt` [protected]
- 25.233.5.2 `SingleSerieUIDFileSetmap gdcM::SerieHelper::SingleSerieUIDFileSetHT` [protected]

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

- [gdcMSerieHelper.h](#)

25.234 gdcM::Series Class Reference

[Series.](#)

```
#include <gdcMSeries.h>
```

Public Member Functions

- [Series](#) ()

25.234.1 Detailed Description

[Series](#).

25.234.2 Constructor & Destructor Documentation

25.234.2.1 gdcm::Series::Series () [inline]

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

- [gdcmSeries.h](#)

25.235 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

25.235.1 Detailed Description

PS 3.4 [Table](#) B.3-1 SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

25.235.2 Constructor & Destructor Documentation

25.235.2.1 gdcm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ()

25.235.3 Member Function Documentation

25.235.3.1 void gdcm::network::ServiceClassApplicationInformation::Print (std::ostream & os) const

25.235.3.2 std::istream& gdcm::network::ServiceClassApplicationInformation::Read (std::istream & is)

25.235.3.3 void gdcm::network::ServiceClassApplicationInformation::SetTuple (uint8_t *levelofsupport*, uint8_t *levelofdigitalsig*, uint8_t *elementcoercion*)

25.235.3.4 `size_t gdcmm::network::ServiceClassApplicationInformation::Size () const`

25.235.3.5 `const std::ostream& gdcmm::network::ServiceClassApplicationInformation::Write (std::ostream & os) const`

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

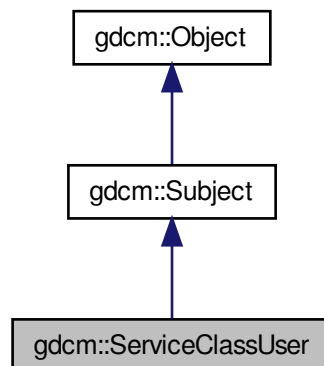
- [gdcmmServiceClassApplicationInformation.h](#)

25.236 gdcmm::ServiceClassUser Class Reference

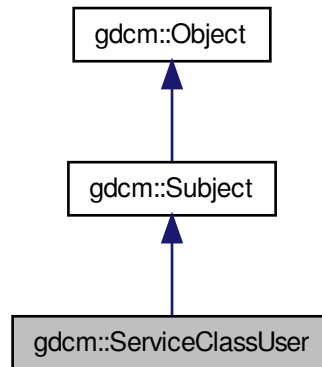
[ServiceClassUser](#).

```
#include <gdcmmServiceClassUser.h>
```

Inheritance diagram for gdcmm::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.

Additional Inherited Members

25.236.1 Detailed Description

[ServiceClassUser](#).

Examples:

[CStoreQtProgress.cxx](#).

25.236.2 Constructor & Destructor Documentation

25.236.2.1 gdcm::ServiceClassUser::ServiceClassUser ()

Construct a SCU with default:

- hostname = localhost
- port = 104

25.236.2.2 gdcm::ServiceClassUser::~~ServiceClassUser ()

25.236.3 Member Function Documentation

25.236.3.1 const char* gdcm::ServiceClassUser::GetAETitle () const

25.236.3.2 const char* gdcm::ServiceClassUser::GetCalledAETitle () const

25.236.3.3 double gdcm::ServiceClassUser::GetTimeout () const

25.236.3.4 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](#).

25.236.3.5 `bool gdcm::ServiceClassUser::IsPresentationContextAccepted (const PresentationContext & pc) const`

Return if the passed in presentation was accepted during association negotiation.

25.236.3.6 `bool gdcm::ServiceClassUser::SendEcho ()`

C-ECHO.

25.236.3.7 `bool gdcm::ServiceClassUser::SendFind (const BaseRootQuery * query, std::vector< DataSet > & retDatasets)`

C-FIND a query, return result are in retDatasets.

25.236.3.8 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, const char * outputdir)`

Execute a C-MOVE, based on query, return files are written in outputdir.

25.236.3.9 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, std::vector< DataSet > & retDatasets)`

Execute a C-MOVE, based on query, returned dataset are Implicit.

25.236.3.10 `bool gdcm::ServiceClassUser::SendMove (const BaseRootQuery * query, std::vector< File > & retFile)`

Execute a C-MOVE, based on query, returned Files are stored in vector.

25.236.3.11 `bool gdcm::ServiceClassUser::SendStore (const char * filename)`

Execute a C-STORE on file on disk, named filename.

Examples:

[CStoreQtProgress.cxx](#).

25.236.3.12 `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.

25.236.3.13 `bool gdcm::ServiceClassUser::SendStore (DataSet const & ds)`

Execute a C-STORE on a [DataSet](#), the transfer syntax used will be Implicit.

25.236.3.14 `void gdcm::ServiceClassUser::SetAETitle (const char * aetitle)`

set calling ae title

25.236.3.15 void gdcm::ServiceClassUser::SetCalledAETitle (const char * *aetitle*)

set called ae title

Examples:

[CStoreQtProgress.cxx](#).

25.236.3.16 void gdcm::ServiceClassUser::SetHostname (const char * *hostname*)

Set the name of the called hostname (hostname or IP address)

Examples:

[CStoreQtProgress.cxx](#).

25.236.3.17 void gdcm::ServiceClassUser::SetPort (uint16_t *port*)

Set port of remote host (called application)

Examples:

[CStoreQtProgress.cxx](#).

25.236.3.18 void gdcm::ServiceClassUser::SetPortSCP (uint16_t *portscp*)

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

25.236.3.19 void gdcm::ServiceClassUser::SetPresentationContexts (std::vector< **PresentationContext** > const & *pcs*)

Set the Presentation Context used for the Association.

Examples:

[CStoreQtProgress.cxx](#).

25.236.3.20 void gdcm::ServiceClassUser::SetTimeout (double *t*)

set/get Timeout

Examples:

[CStoreQtProgress.cxx](#).

25.236.3.21 bool gdcm::ServiceClassUser::StartAssociation ()

Start the association. Need to call SetPresentationContexts before.

Examples:

[CStoreQtProgress.cxx](#).

25.236.3.22 bool gdcm::ServiceClassUser::StopAssociation ()

Stop the running association.

Examples:

[CStoreQtProgress.cxx](#).

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

- [gdcmServiceClassUser.h](#)

25.237 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])

25.237.1 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

25.237.2 Constructor & Destructor Documentation

25.237.2.1 gdcm::SHA1::SHA1 ()

25.237.2.2 gdcm::SHA1::~~SHA1 ()

25.237.3 Member Function Documentation

25.237.3.1 `static bool gdcM::SHA1::Compute (const char * buffer, unsigned long buf_len, char digest_str[20*2+1])`
[static]

25.237.3.2 `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:

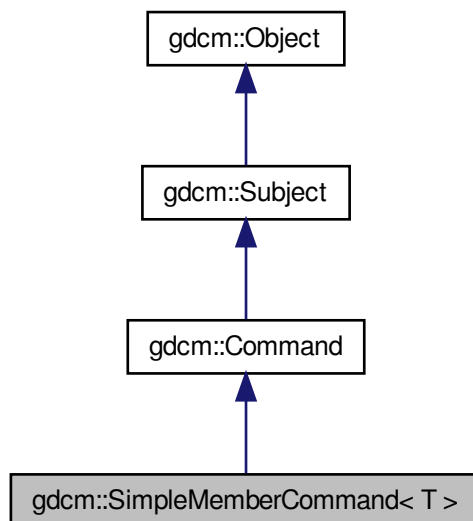
- [gdcMSHA1.h](#)

25.238 gdcM::SimpleMemberCommand< T > Class Template Reference

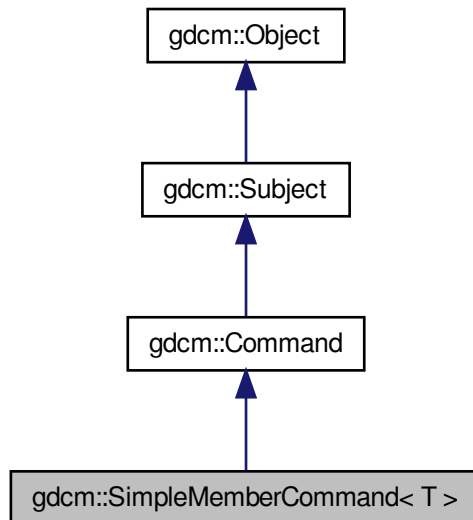
[Command](#) subclass that calls a pointer to a member function.

```
#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](#)

25.238.1 Detailed Description

`template<typename T>class gdcmm::SimpleMemberCommand< T >`

[Command](#) subclass that calls a pointer to a member function.

[SimpleMemberCommand](#) calls a pointer to a member function with no arguments.

25.238.2 Member Typedef Documentation

25.238.2.1 `template<typename T > typedef SimpleMemberCommand gdcmm::SimpleMemberCommand< T >::Self`

Standard class typedefs.

25.238.2.2 `template<typename T > typedef void(T::* gdcmm::SimpleMemberCommand< T >::TMemberFunctionPointer)()`

A method callback.

25.238.3 Constructor & Destructor Documentation

25.238.3.1 `template<typename T > gdcmm::SimpleMemberCommand< T >::SimpleMemberCommand ()`
`[inline], [protected]`

Referenced by `gdcmm::SimpleMemberCommand< T >::New()`.

25.238.3.2 `template<typename T > virtual gdcmm::SimpleMemberCommand< T >::~~SimpleMemberCommand ()`
`[inline], [protected], [virtual]`

25.238.4 Member Function Documentation

25.238.4.1 `template<typename T > virtual void gdcmm::SimpleMemberCommand< T >::Execute (Subject *, const Event &)` `[inline], [virtual]`

Invoke the callback function.

Implements [gdcmm::Command](#).

References `gdcmm::SimpleMemberCommand< T >::m_MemberFunction`.

25.238.4.2 `template<typename T > virtual void gdcmm::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 [gdcmm::Command](#).

References `gdcm::SimpleMemberCommand< T >::m_MemberFunction`.

25.238.4.3 `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()`.

25.238.4.4 `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`.

25.238.5 Member Data Documentation

25.238.5.1 `template<typename T > TMemberFunctionPointer gdcm::SimpleMemberCommand< T >::m_MemberFunction [protected]`

Referenced by `gdcm::SimpleMemberCommand< T >::Execute()`, and `gdcm::SimpleMemberCommand< T >::SetCallbackFunction()`.

25.238.5.2 `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](#)

25.239 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 [ShowIteration](#) ()
- virtual void [ShowProgress](#) ([Subject](#) *caller, const [Event](#) &evt)
- virtual void [StartFilter](#) ()
- void [TestAbortOff](#) ()
- void [TestAbortOn](#) ()

25.239.1 Detailed Description

[SimpleSubjectWatcher](#) This is a typical [Subject](#) Watcher class. It will observe all events.

25.239.2 Constructor & Destructor Documentation

25.239.2.1 `gdcmm::SimpleSubjectWatcher::SimpleSubjectWatcher (Subject * s, const char * comment = " ")`

25.239.2.2 `virtual gdcmm::SimpleSubjectWatcher::~SimpleSubjectWatcher ()` [virtual]

25.239.3 Member Function Documentation

25.239.3.1 `virtual void gdcmm::SimpleSubjectWatcher::EndFilter ()` [protected],[virtual]

25.239.3.2 `virtual void gdcmm::SimpleSubjectWatcher::ShowAbort ()` [protected],[virtual]

25.239.3.3 `virtual void gdcmm::SimpleSubjectWatcher::ShowAnonymization (Subject * caller, const Event & evt)`
[protected],[virtual]

25.239.3.4 `virtual void gdcmm::SimpleSubjectWatcher::ShowData (Subject * caller, const Event & evt)` [protected],
[virtual]

25.239.3.5 `virtual void gdcmm::SimpleSubjectWatcher::ShowDataSet (Subject * caller, const Event & evt)` [protected],
[virtual]

25.239.3.6 `virtual void gdcmm::SimpleSubjectWatcher::ShowIteration ()` [protected],[virtual]

25.239.3.7 `virtual void gdcmm::SimpleSubjectWatcher::ShowProgress (Subject * caller, const Event & evt)` [protected],
[virtual]

25.239.3.8 `virtual void gdcmm::SimpleSubjectWatcher::StartFilter ()` [protected],[virtual]

25.239.3.9 `void gdcmm::SimpleSubjectWatcher::TestAbortOff ()` [protected]

25.239.3.10 `void gdcmm::SimpleSubjectWatcher::TestAbortOn ()` [protected]

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

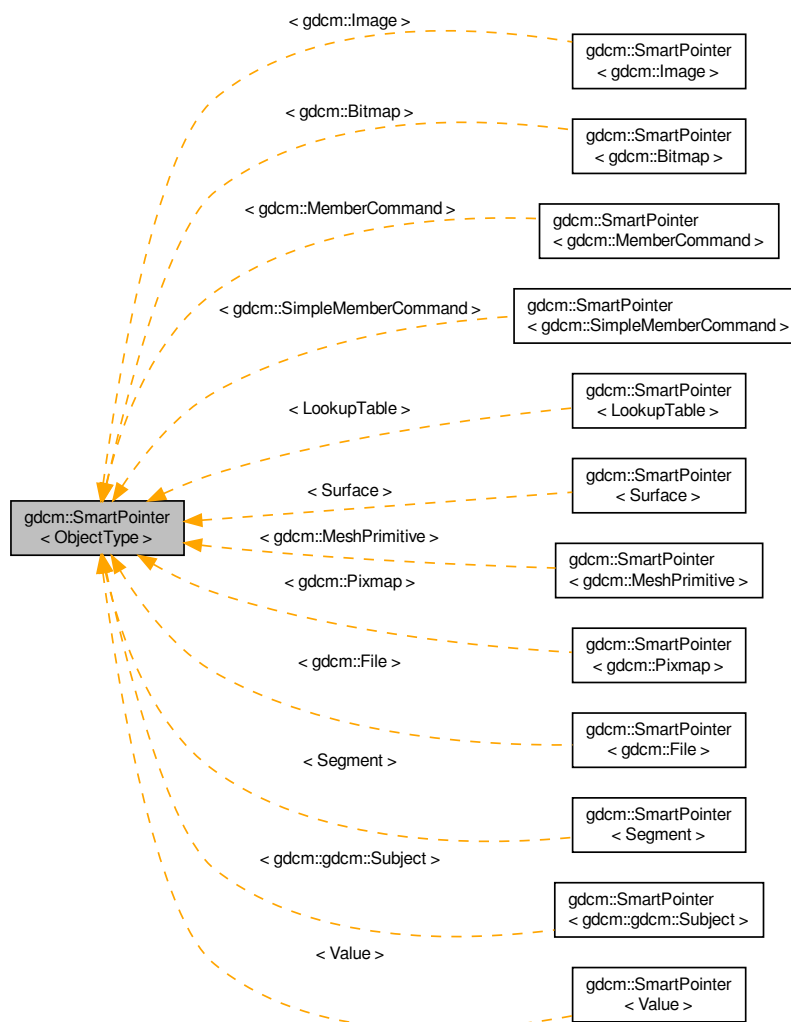
- [gdcmmSimpleSubjectWatcher.h](#)

25.240 `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)`

25.240.1 Detailed Description

```
template<class ObjectType>class gdcmm::SmartPointer< ObjectType >
```

Class for Smart Pointer.

Will only work for subclass of `gdcmm::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](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDS-Explicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), and [ReadExplicitLengthSQIVR.cxx](#).

25.240.2 Constructor & Destructor Documentation

25.240.2.1 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer () [inline]`

25.240.2.2 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer (const SmartPointer< ObjectType > & p) [inline]`

25.240.2.3 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer (ObjectType * p) [inline]`

25.240.2.4 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer (ObjectType const & p) [inline]`

25.240.2.5 `template<class ObjectType> gdcm::SmartPointer< ObjectType >::~SmartPointer () [inline]`

25.240.3 Member Function Documentation

25.240.3.1 `template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::GetPointer () const [inline]`

Explicit function to retrieve the pointer.

25.240.3.2 `template<class ObjectType> gdcm::SmartPointer< ObjectType >::operator ObjectType * () const [inline]`

Return pointer to object.

25.240.3.3 `template<class ObjectType> ObjectType& gdcm::SmartPointer< ObjectType >::operator* () const [inline]`

25.240.3.4 `template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::operator-> () const [inline]`

Overload operator ->

25.240.3.5 `template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (SmartPointer< ObjectType > const & r) [inline]`

Overload operator assignment.

Referenced by `gdcm::SmartPointer< Value >::operator=()`.

25.240.3.6 `template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= (ObjectType * r) [inline]`

Overload operator assignment.

25.240.3.7 `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](#)

25.241 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

25.241.1 Detailed Description

[SOPClassExtendedNegociationSub](#) PS 3.7 [Table D.3-11](#) SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

25.241.2 Constructor & Destructor Documentation

25.241.2.1 `gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ()`

25.241.3 Member Function Documentation

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

25.241.3.2 `std::istream& gdcm::network::SOPClassExtendedNegociationSub::Read (std::istream & is)`

25.241.3.3 `void gdcm::network::SOPClassExtendedNegociationSub::SetTuple (const char * uid, uint8_t levelofsupport = 3, uint8_t levelofdigitalsig = 0, uint8_t elementcoercion = 2)`

25.241.3.4 `size_t gdcm::network::SOPClassExtendedNegociationSub::Size () const`

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

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

- [gdcmSOPClassExtendedNegociationSub.h](#)

25.242 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](#) ()

25.242.1 Detailed Description

Class convert a class SOP Class UID into [IOD](#).

Reference PS 3.4 [Table B.5-1](#) STANDARD SOP CLASSES

25.242.2 Member Typedef Documentation

25.242.2.1 `typedef const char* gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]`

25.242.3 Member Function Documentation

25.242.3.1 `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](#).

25.242.3.2 `static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID (const char * sopclassuid)` [`static`]

25.242.3.3 `static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ()` [`static`]

Return the number of SOP Class UID listed internally.

25.242.3.4 `static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD (const char * iod)` [`static`]

25.242.3.5 `static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD (unsigned int i)` [`static`]

25.242.3.6 `static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ()` [`static`]

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

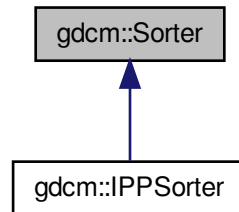
- [gdcmSOPClassUIDToIOD.h](#)

25.243 gdcm::Sorter Class Reference

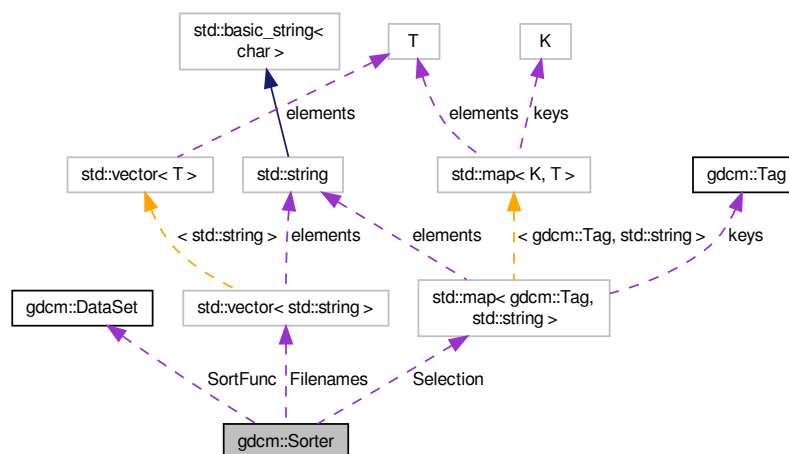
[Sorter](#) General class to do sorting using a custom function You simply need to provide a function of type: [Sorter::Sort-Function](#).

```
#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 `gdcm::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)`

25.243.1 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](#).

25.243.2 Member Typedef Documentation

25.243.2.1 `typedef std::map<Tag,std::string> gdcm::Sorter::SelectionMap` [protected]

25.243.2.2 `typedef bool(* gdcm::Sorter::SortFunction)(DataSet const &, DataSet const &)`

Set the sort function which compares one dataset to the other.

25.243.3 Constructor & Destructor Documentation

25.243.3.1 `gdcmm::Sorter::Sorter ()`

25.243.3.2 `virtual gdcmm::Sorter::~~Sorter ()` `[virtual]`

25.243.4 Member Function Documentation

25.243.4.1 `bool gdcmm::Sorter::AddSelect (Tag const & tag, const char * value)`

UNSUPPORTED FOR NOW.

25.243.4.2 `const std::vector<std::string>& gdcmm::Sorter::GetFileNames () const` `[inline]`

Return the list of filenames as sorted by the specific algorithm used. Empty by default (before [Sort\(\)](#) is called)

Examples:

[gdcmmorthoplanes.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

25.243.4.3 `void gdcmm::Sorter::Print (std::ostream & os) const`

Print.

Examples:

[gdcmmorthoplanes.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

Referenced by `gdcmm::operator<<()`.

25.243.4.4 `void gdcmm::Sorter::SetSortFunction (SortFunction f)`

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

25.243.4.5 `virtual bool gdcmm::Sorter::Sort (std::vector< std::string > const & filenames)` `[virtual]`

Typically the output of `gdcmm::Directory::GetFileNames()`

Reimplemented in `gdcmm::IPPSorter`.

Examples:

[SortImage.cxx](#).

25.243.4.6 `virtual bool gdcmm::Sorter::StableSort (std::vector< std::string > const & filenames)` `[virtual]`

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

25.243.5 Friends And Related Function Documentation

25.243.5.1 `std::ostream& operator<< (std::ostream & _os, const Sorter & s)` [*friend*]

25.243.6 Member Data Documentation

25.243.6.1 `std::vector<std::string> gdcmm::Sorter::FileNames` [*protected*]

25.243.6.2 `std::map<Tag,std::string> gdcmm::Sorter::Selection` [*protected*]

25.243.6.3 **SortFunction** `gdcmm::Sorter::SortFunc` [*protected*]

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

- [gdcmmSorter.h](#)

25.244 gdcmm::Spacing Class Reference

Class for [Spacing](#).

```
#include <gdcmmSpacing.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)

25.244.1 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

25.244.2 Member Enumeration Documentation

25.244.2.1 enum gdcm::Spacing::SpacingType

Enumerator

DETECTOR
MAGNIFIED
CALIBRATED
UNKNOWN

25.244.3 Constructor & Destructor Documentation

25.244.3.1 gdcm::Spacing::Spacing ()

25.244.3.2 gdcm::Spacing::~~Spacing ()

25.244.4 Member Function Documentation

25.244.4.1 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](#)

25.245 gdcm::Spectroscopy Class Reference

[Spectroscopy](#) class.

```
#include <gdcmSpectroscopy.h>
```

Public Member Functions

- [Spectroscopy](#) ()

25.245.1 Detailed Description

[Spectroscopy](#) class.

25.245.2 Constructor & Destructor Documentation

25.245.2.1 [gdcm::Spectroscopy::Spectroscopy](#) () [inline]

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

- [gdcmSpectroscopy.h](#)

25.246 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.

25.246.1 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.

25.246.2 Constructor & Destructor Documentation

25.246.2.1 `gdcm::SplitMosaicFilter::SplitMosaicFilter ()`

25.246.2.2 `gdcm::SplitMosaicFilter::~~SplitMosaicFilter ()`

25.246.3 Member Function Documentation

25.246.3.1 `bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions (unsigned int dims[3])`

Compute the new dimensions according to private information stored in the MOSAIC header.

25.246.3.2 `File& gdcm::SplitMosaicFilter::GetFile ()` `[inline]`

25.246.3.3 `const File& gdcm::SplitMosaicFilter::GetFile () const` `[inline]`

25.246.3.4 `const Image& gdcm::SplitMosaicFilter::GetImage () const` `[inline]`

25.246.3.5 `Image& gdcm::SplitMosaicFilter::GetImage ()` `[inline]`

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

25.246.3.7 `void gdcm::SplitMosaicFilter::SetImage (const Image & image)`

25.246.3.8 `bool gdcm::SplitMosaicFilter::Split ()`

Split the SIEMENS MOSAIC image.

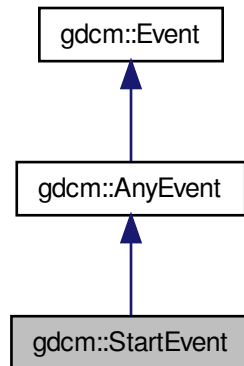
The documentation for this class was generated from the following file:

- [gdcmSplitMosaicFilter.h](#)

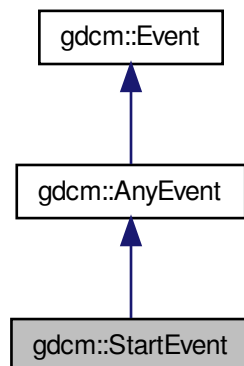
25.247 gdcm::StartEvent Class Reference

```
#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](#)

25.248 `gdcm::static_assert_test< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

25.249 `gdcm::STATIC_ASSERTION_FAILURE< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

25.250 `gdcm::STATIC_ASSERTION_FAILURE< true >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

Public Types

- enum { [value](#) = 1 }

25.250.1 Member Enumeration Documentation

25.250.1.1 anonymous enum

Enumerator

value

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

- [gdcmStaticAssert.h](#)

25.251 `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)

25.251.1 Detailed Description

[StreamImageReader](#).

Note

its role is to convert the DICOM [DataSet](#) into a [gdcmm::Image](#) representation via an ITK streaming (ie, multithreaded) interface [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space. Currently, this class is thread safe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

See Also

[Image](#)

Examples:

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

25.251.2 Constructor & Destructor Documentation

25.251.2.1 [gdcmm::StreamImageReader::StreamImageReader](#) ()

25.251.2.2 [virtual gdcmm::StreamImageReader::~~StreamImageReader](#) () `[virtual]`

25.251.3 Member Function Documentation

25.251.3.1 [bool gdcmm::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](#).

25.251.3.2 [void gdcmm::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](#).

25.251.3.3 `uint32_t gdcmm::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](#).

25.251.3.4 `std::vector<unsigned int> gdcmm::StreamImageReader::GetDimensionsValueForResolution (unsigned int)`

25.251.3.5 `File const& gdcmm::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](#).

25.251.3.6 `bool gdcmm::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 `metainageio` 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](#).

25.251.3.7 `virtual bool gdcmm::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](#).

25.251.3.8 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:

[StreamImageReaderTest.cxx](#).

25.251.3.9 void gdcm::StreamImageReader::SetStream (std::istream & *inStream*)

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

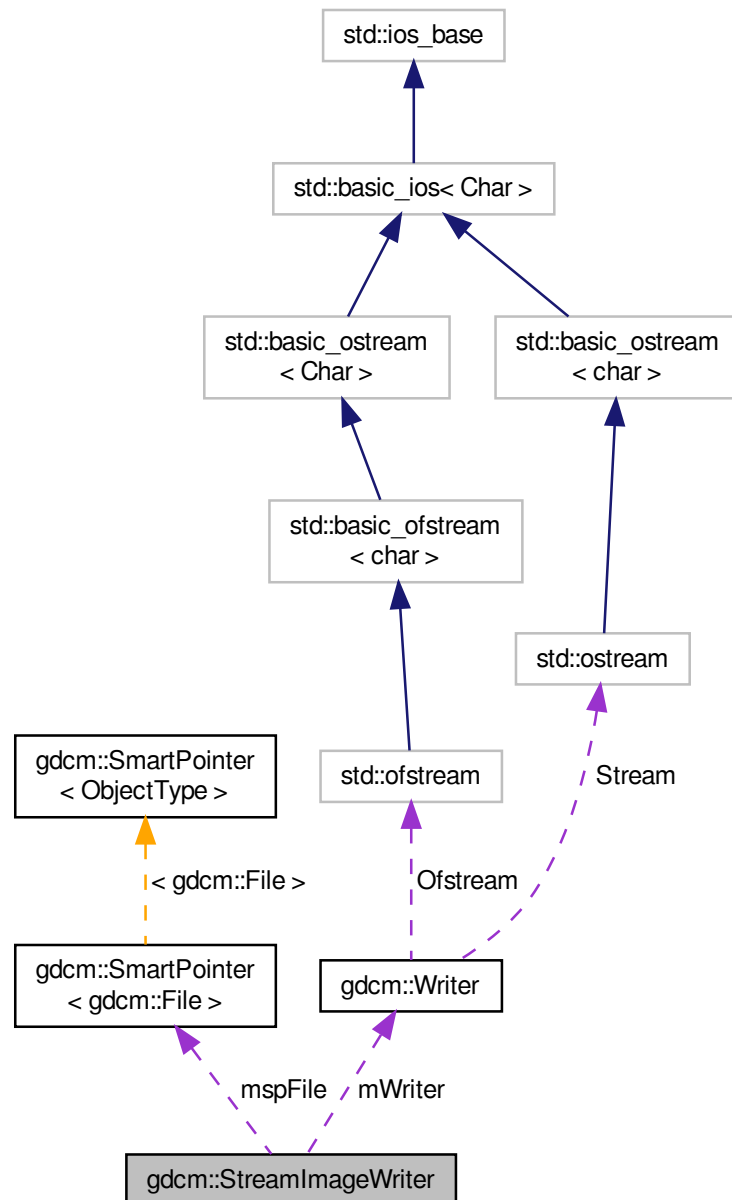
- [gdcmStreamImageReader.h](#)

25.252 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](#)

25.252.1 Detailed Description

[StreamImageReader](#).

Note

its role is to convert the DICOM [DataSet](#) into a [gdcm::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](#).

25.252.2 Constructor & Destructor Documentation

25.252.2.1 `gdcm::StreamImageWriter::StreamImageWriter ()`

25.252.2.2 `virtual gdcm::StreamImageWriter::~~StreamImageWriter ()` [virtual]

25.252.3 Member Function Documentation

25.252.3.1 `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](#).

25.252.3.2 `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](#).

25.252.3.3 `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](#).

25.252.3.4 `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](#).

25.252.3.5 `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.

25.252.3.6 void gdcm::StreamImageWriter::SetStream (std::ostream & *inStream*)

Examples:

[Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

25.252.3.7 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](#).

25.252.3.8 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](#).

25.252.3.9 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

25.252.3.10 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.

25.252.4 Member Data Documentation

25.252.4.1 `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.

25.252.4.2 `int gdcM::StreamImageWriter::mElementOffsets1` `[protected]`

25.252.4.3 `SmartPointer<File> gdcM::StreamImageWriter::mspFile` `[protected]`

25.252.4.4 `Writer gdcM::StreamImageWriter::mWriter` `[protected]`

25.252.4.5 `uint16_t gdcM::StreamImageWriter::mXMax` `[protected]`

25.252.4.6 `uint16_t gdcM::StreamImageWriter::mXMin` `[protected]`

25.252.4.7 `uint16_t gdcM::StreamImageWriter::mYMax` `[protected]`

25.252.4.8 `uint16_t gdcM::StreamImageWriter::mYMin` `[protected]`

25.252.4.9 `uint16_t gdcM::StreamImageWriter::mZMax` `[protected]`

25.252.4.10 `uint16_t gdcM::StreamImageWriter::mZMin` `[protected]`

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

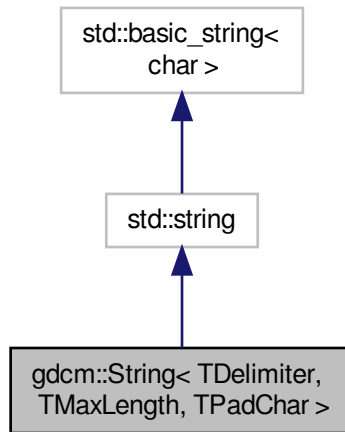
- [gdcMStreamImageWriter.h](#)

25.253 `gdcM::String< TDelimiter, TMaxLength, TPadChar >` Class Template Reference

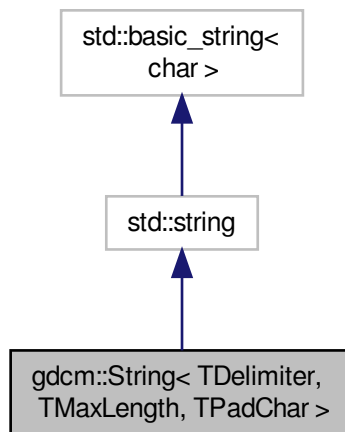
[String.](#)

```
#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
- [gdcmm::String](#)< TDelimiter, TMaxLength, TPadChar > [Truncate](#) () const

Static Public Member Functions

- static std::string [Trim](#) (const char *input)

25.253.1 Detailed Description

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>class gdcmm::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)

25.253.2 Member Typedef Documentation

25.253.2.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_iterator`

25.253.2.2 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_reference`

25.253.2.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_reverse_iterator`

25.253.2.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::difference_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::difference_type`

25.253.2.5 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator`

25.253.2.6 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::pointer gdcm::String< TDelimiter, TMaxLength, TPadChar >::pointer`

25.253.2.7 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::reference`

25.253.2.8 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::reverse_iterator`

25.253.2.9 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::size_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::size_type`

25.253.2.10 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type`

25.253.3 Constructor & Destructor Documentation

25.253.3.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String () [inline]`

[String](#) constructors.

25.253.3.2 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * s) [inline]`

25.253.3.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String (const value_type * s, size_type n) [inline]`

25.253.3.4 `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]`

25.253.4 Member Function Documentation

25.253.4.1 `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()`.

25.253.4.2 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '> gdcmm::String< TDelimiter, TMaxLength, TPadChar >::operator const char * () const [inline]`

WARNING: Trailing \0 might be lost in this operation:

25.253.4.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '> std::string gdcmm::String< TDelimiter, TMaxLength, TPadChar >::Trim () const [inline]`

Trim function is required to return a std::string object, otherwise we could not create a [gdcmm::String](#) object with an odd number of bytes...

25.253.4.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '> static std::string gdcmm::String< TDelimiter, TMaxLength, TPadChar >::Trim (const char * input) [inline], [static]`

25.253.4.5 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '> gdcmm::String< TDelimiter, TMaxLength, TPadChar > gdcmm::String< TDelimiter, TMaxLength, TPadChar >::Truncate () const [inline]`

References `gdcmm::String< TDelimiter, TMaxLength, TPadChar >::IsValid()`.

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

- [gdcmmString.h](#)

25.254 gdcmm::StringFilter Class Reference

[StringFilter](#) [StringFilter](#) is the class that make gdcmm2.x looks more like gdcmm1 and transform the binary blob contained in a [DataElement](#) into a string, typically this is a nice feature to have for wrapped language.

```
#include <gdcmmStringFilter.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)
DEPRECATED: NEVER USE IT.
- std::string [FromString](#) (const [Tag](#) &t, const char *value, size_t 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 [Tag](#) &t) const
Convert to string the ByteValue contained in a DataElement.
- std::pair< std::string, std::string > [ToStringPair](#) (const [Tag](#) &t) const
- 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

25.254.1 Detailed Description

[StringFilter](#) [StringFilter](#) is the class that make gdcmm2.x looks more like gdcmm1 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](#), and [SimplePrintPatientName.cs](#).

25.254.2 Constructor & Destructor Documentation

25.254.2.1 gdcmm::StringFilter::StringFilter ()

25.254.2.2 gdcmm::StringFilter::~~StringFilter ()

25.254.3 Member Function Documentation

25.254.3.1 bool gdcmm::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

25.254.3.2 bool gdcmm::StringFilter::ExecuteQuery (std::string const & *query*, [DataSet](#) const & *ds*, std::string & *value*) const
[protected]

25.254.3.3 std::string gdcmm::StringFilter::FromString (const [Tag](#) & *t*, const char * *value*, VL const & *vl*)

DEPRECATED: NEVER USE IT.

25.254.3.4 std::string gdcmm::StringFilter::FromString (const [Tag](#) & *t*, const char * *value*, size_t *len*)

25.254.3.5 File& gdcmm::StringFilter::GetFile () [inline]

25.254.3.6 const File& gdcmm::StringFilter::GetFile () const [inline]

25.254.3.7 void gdcmm::StringFilter::SetDicts (const [Dicts](#) & *dicts*)

Allow user to pass in there own dicts.

25.254.3.8 void gdcmm::StringFilter::SetFile (const [File](#) & *f*) [inline]

Set/Get [File](#).

Examples:

[ReadAndPrintAttributes.cxx](#).

25.254.3.9 `std::string gdcM::StringFilter::ToString (const Tag & t) const`

Convert to string the [ByteValue](#) contained in a [DataElement](#).

Examples:

[ReadAndPrintAttributes.cxx](#).

25.254.3.10 `std::pair<std::string, std::string> gdcM::StringFilter::ToStringPair (const Tag & t) 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](#) pair.second : the value encoded into a string (US,UL...) are properly converted

Examples:

[ReadAndPrintAttributes.cxx](#).

25.254.3.11 `std::pair<std::string, std::string> gdcM::StringFilter::ToStringPair (const Tag & t, DataSet const & ds) const`
[protected]

25.254.3.12 `void gdcM::StringFilter::UseDictAlways (bool)` [inline]

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

- [gdcMStringFilter.h](#)

25.255 gdcM::Study Class Reference

[Study](#).

```
#include <gdcMStudy.h>
```

Public Member Functions

- [Study](#) ()

25.255.1 Detailed Description

[Study](#).

25.255.2 Constructor & Destructor Documentation

25.255.2.1 gdcm::Study::Study () [inline]

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

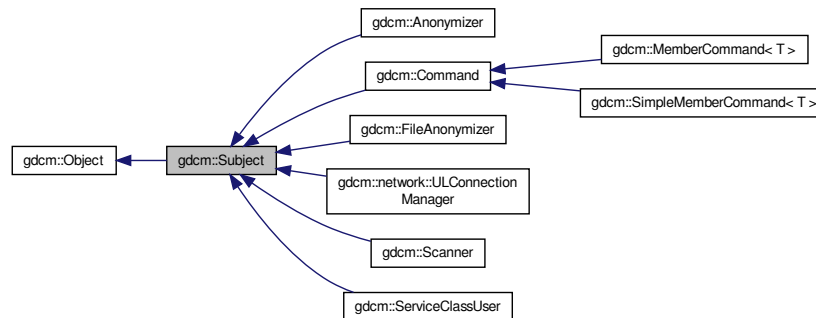
- [gdcmStudy.h](#)

25.256 gdcm::Subject Class Reference

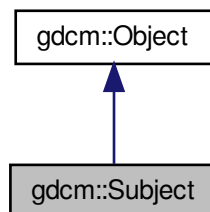
[Subject.](#)

```
#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

25.256.1 Detailed Description

[Subject](#).

See Also

[Command Event](#)

25.256.2 Constructor & Destructor Documentation

25.256.2.1 `gdcm::Subject::Subject ()`

25.256.2.2 `gdcm::Subject::~~Subject ()`

25.256.3 Member Function Documentation

25.256.3.1 `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

25.256.3.2 `unsigned long gdcm::Subject::AddObserver (const Event & event, Command *) const`

25.256.3.3 `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 assign this to a [Command::Pointer](#). Since [Command](#) inherits from [LightObject](#), at this point in the code, only a pointer or a reference to the [Command](#) can be used.

25.256.3.4 `bool gdcm::Subject::HasObserver (const Event & event) const`

Return true if an observer is registered for this event.

25.256.3.5 void gdcM::Subject::InvokeEvent (const Event &)

Call Execute on all the Commands observing this event id.

25.256.3.6 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.

25.256.3.7 void gdcM::Subject::RemoveAllObservers ()

Remove all observers .

25.256.3.8 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:

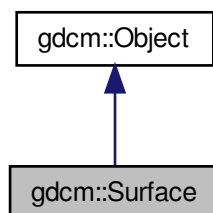
- [gdcMSubject.h](#)

25.257 gdcM::Surface Class Reference

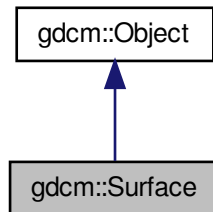
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

```
#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

25.257.1 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

25.257.2 Member Enumeration Documentation

25.257.2.1 enum `gdcm::Surface::STATES`

Enumerator

NO
YES
UNKNOWN
STATES_END

25.257.2.2 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

25.257.3 Constructor & Destructor Documentation

25.257.3.1 `gdcm::Surface::Surface ()`

25.257.3.2 `virtual gdcm::Surface::~~Surface ()` [`virtual`]

25.257.4 Member Function Documentation

25.257.4.1 **SegmentHelper::BasicCodedEntry** const& gdcm::Surface::GetAlgorithmFamily () const

25.257.4.2 **SegmentHelper::BasicCodedEntry**& gdcm::Surface::GetAlgorithmFamily ()

25.257.4.3 const char* gdcm::Surface::GetAlgorithmName () const

25.257.4.4 const char* gdcm::Surface::GetAlgorithmVersion () const

25.257.4.5 const float* gdcm::Surface::GetAxisOfRotation () const

Note

Pointer is null if undefined

25.257.4.6 const float* gdcm::Surface::GetCenterOfRotation () const

Note

Pointer is null if undefined

25.257.4.7 **STATES** gdcm::Surface::GetFiniteVolume () const

25.257.4.8 **STATES** gdcm::Surface::GetManifold () const

25.257.4.9 float gdcm::Surface::GetMaximumPointDistance () const

25.257.4.10 float gdcm::Surface::GetMeanPointDistance () const

25.257.4.11 **MeshPrimitive** const& gdcm::Surface::GetMeshPrimitive () const

25.257.4.12 **MeshPrimitive**& gdcm::Surface::GetMeshPrimitive ()

25.257.4.13 unsigned long gdcm::Surface::GetNumberOfSurfacePoints () const

25.257.4.14 unsigned long gdcm::Surface::GetNumberOfVectors () const

25.257.4.15 const **DataElement**& gdcm::Surface::GetPointCoordinatesData () const

25.257.4.16 **DataElement**& gdcm::Surface::GetPointCoordinatesData ()

25.257.4.17 const float* gdcm::Surface::GetPointPositionAccuracy () const

Note

Pointer is null if undefined

25.257.4.18 const float* gdcm::Surface::GetPointsBoundingBoxCoordinates () const

Note

Pointer is null if undefined

- 25.257.4.19 **SegmentHelper::BasicCodedEntry** const& gdcm::Surface::GetProcessingAlgorithm () const
- 25.257.4.20 **SegmentHelper::BasicCodedEntry**& gdcm::Surface::GetProcessingAlgorithm ()
- 25.257.4.21 const unsigned short* gdcm::Surface::GetRecommendedDisplayCIELabValue () const
- 25.257.4.22 unsigned short gdcm::Surface::GetRecommendedDisplayCIELabValue (const unsigned int *idx*) const
- 25.257.4.23 unsigned short gdcm::Surface::GetRecommendedDisplayGrayscaleValue () const
- 25.257.4.24 float gdcm::Surface::GetRecommendedPresentationOpacity () const
- 25.257.4.25 **VIEWType** gdcm::Surface::GetRecommendedPresentationType () const
- 25.257.4.26 static **STATES** gdcm::Surface::GetSTATES (const char * *state*) [static]
- 25.257.4.27 static const char* gdcm::Surface::GetSTATESString (**STATES** *state*) [static]
- 25.257.4.28 const char* gdcm::Surface::GetSurfaceComments () const
- 25.257.4.29 unsigned long gdcm::Surface::GetSurfaceNumber () const
- 25.257.4.30 bool gdcm::Surface::GetSurfaceProcessing () const
- 25.257.4.31 const char* gdcm::Surface::GetSurfaceProcessingDescription () const
- 25.257.4.32 float gdcm::Surface::GetSurfaceProcessingRatio () const
- 25.257.4.33 const float* gdcm::Surface::GetVectorAccuracy () const
- 25.257.4.34 const **DataElement**& gdcm::Surface::GetVectorCoordinateData () const
- 25.257.4.35 **DataElement**& gdcm::Surface::GetVectorCoordinateData ()
- 25.257.4.36 unsigned short gdcm::Surface::GetVectorDimensionality () const
- 25.257.4.37 static **VIEWType** gdcm::Surface::GetVIEWType (const char * *type*) [static]
- 25.257.4.38 static const char* gdcm::Surface::GetVIEWTypeString (**VIEWType** *type*) [static]
- 25.257.4.39 void gdcm::Surface::SetAlgorithmFamily (**SegmentHelper::BasicCodedEntry** const & *BSE*)
- 25.257.4.40 void gdcm::Surface::SetAlgorithmName (const char * *str*)
- 25.257.4.41 void gdcm::Surface::SetAlgorithmVersion (const char * *str*)
- 25.257.4.42 void gdcm::Surface::SetAxisOfRotation (const float * *axis*)

- 25.257.4.43 void gdcm::Surface::SetCenterOfRotation (const float * *center*)
- 25.257.4.44 void gdcm::Surface::SetFiniteVolume (STATES *state*)
- 25.257.4.45 void gdcm::Surface::SetManifold (STATES *state*)
- 25.257.4.46 void gdcm::Surface::SetMaximumPointDistance (float *maximum*)
- 25.257.4.47 void gdcm::Surface::SetMeanPointDistance (float *average*)
- 25.257.4.48 void gdcm::Surface::SetMeshPrimitive (MeshPrimitive & *mp*)
- 25.257.4.49 void gdcm::Surface::SetNumberOfSurfacePoints (const unsigned long *nb*)
- 25.257.4.50 void gdcm::Surface::SetNumberOfVectors (const unsigned long *nb*)
- 25.257.4.51 void gdcm::Surface::SetPointCoordinatesData (DataElement const & *de*)
- 25.257.4.52 void gdcm::Surface::SetPointPositionAccuracy (const float * *accuracies*)
- 25.257.4.53 void gdcm::Surface::SetPointsBoundingBoxCoordinates (const float * *coordinates*)
- 25.257.4.54 void gdcm::Surface::SetProcessingAlgorithm (SegmentHelper::BasicCodedEntry const & *BSE*)
- 25.257.4.55 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl[3]*)
- 25.257.4.56 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const unsigned short *vl*, const unsigned int *idx* = 0)
- 25.257.4.57 void gdcm::Surface::SetRecommendedDisplayCIELabValue (const std::vector< unsigned short > & *vl*)
- 25.257.4.58 void gdcm::Surface::SetRecommendedDisplayGrayscaleValue (const unsigned short *vl*)
- 25.257.4.59 void gdcm::Surface::SetRecommendedPresentationOpacity (const float *opacity*)
- 25.257.4.60 void gdcm::Surface::SetRecommendedPresentationType (VIEWType *type*)
- 25.257.4.61 void gdcm::Surface::SetSurfaceComments (const char * *comment*)
- 25.257.4.62 void gdcm::Surface::SetSurfaceNumber (const unsigned long *nb*)
- 25.257.4.63 void gdcm::Surface::SetSurfaceProcessing (bool *b*)
- 25.257.4.64 void gdcm::Surface::SetSurfaceProcessingDescription (const char * *description*)
- 25.257.4.65 void gdcm::Surface::SetSurfaceProcessingRatio (const float *ratio*)
- 25.257.4.66 void gdcm::Surface::SetVectorAccuracy (const float * *accuracy*)
- 25.257.4.67 void gdcm::Surface::SetVectorCoordinateData (DataElement const & *de*)

25.257.4.68 void `gdcm::Surface::SetVectorDimensionality` (const unsigned short *dim*)

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

- [gdcmSurface.h](#)

25.258 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).

25.258.1 Detailed Description

[SurfaceHelper](#) Helper class for [Surface](#) object.

25.258.2 Member Typedef Documentation

25.258.2.1 typedef std::vector< unsigned short > `gdcm::SurfaceHelper::ColorArray`

25.258.3 Member Function Documentation

25.258.3.1 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

<i>CIELab</i>	DICOM CIE-Lab array.
<i>rangeMax</i>	Max value of the RGB range.

Template Parameters

<i>T</i>	Type of CIELab components.
<i>U</i>	Type of rangeMax value.

25.258.3.2 `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

<i>CIELab</i>	DICOM CIE-Lab array.
<i>rangeMax</i>	Max value of the RGB range.

Template Parameters

<i>U</i>	Type of rangeMax value.
----------	-------------------------

25.258.3.3 `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

<i>RGB</i>	RGB array.
<i>rangeMax</i>	Max value of the RGB range.

Template Parameters

<i>T</i>	Type of RGB components.
----------	-------------------------

<i>U</i>	Type of rangeMax value.
----------	-------------------------

25.258.3.4 `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

<i>RGB</i>	RGB array.
<i>rangeMax</i>	Max value of the RGB range.

Template Parameters

<i>T</i>	Type of RGB components.
<i>U</i>	Type of rangeMax value.

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

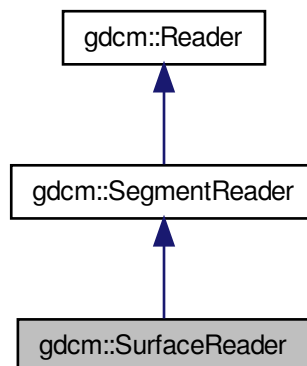
- [gdcmSurfaceHelper.h](#)

25.259 gdcm::SurfaceReader Class Reference

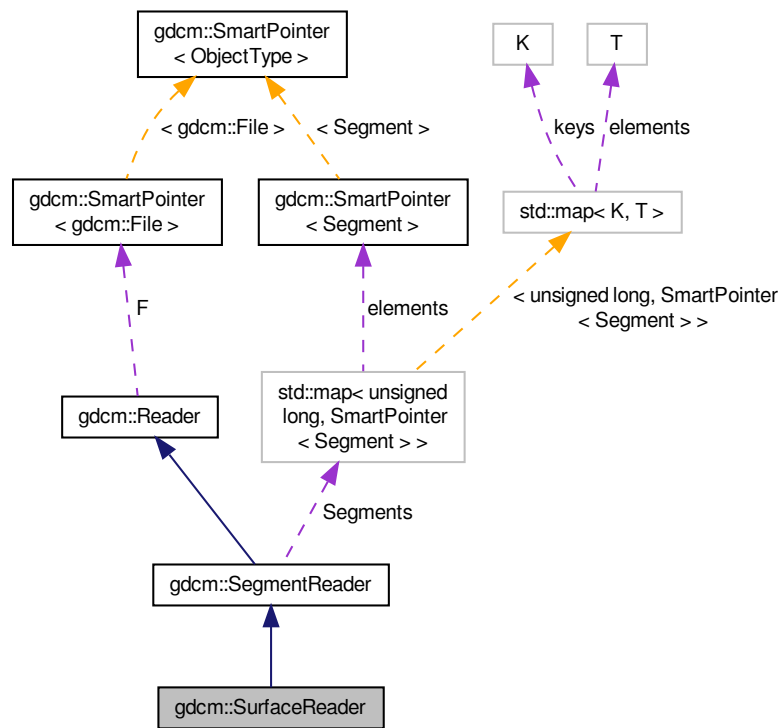
This class defines a SURFACE IE reader. It reads surface mesh module attributes.

```
#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

25.259.1 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

25.259.2 Constructor & Destructor Documentation

25.259.2.1 `gdcm::SurfaceReader::SurfaceReader ()`

25.259.2.2 `virtual gdcm::SurfaceReader::~~SurfaceReader ()` [virtual]

25.259.3 Member Function Documentation

25.259.3.1 `unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ()` const

25.259.3.2 `virtual bool gdcm::SurfaceReader::Read ()` [virtual]

Read.

Reimplemented from [gdcm::SegmentReader](#).

25.259.3.3 `bool gdcm::SurfaceReader::ReadPointMacro (SmartPointer< Surface > surface, const DataSet & surfaceDS)`
[protected]

25.259.3.4 `bool gdcm::SurfaceReader::ReadSurface (const Item & surfaceltem, const unsigned long idx)` [protected]

25.259.3.5 `bool gdcm::SurfaceReader::ReadSurfaces ()` [protected]

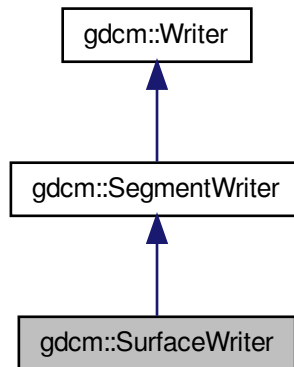
The documentation for this class was generated from the following file:

- [gdcmSurfaceReader.h](#)

25.260 gdcm::SurfaceWriter Class Reference

This class defines a SURFACE IE writer. It writes surface mesh module attributes.

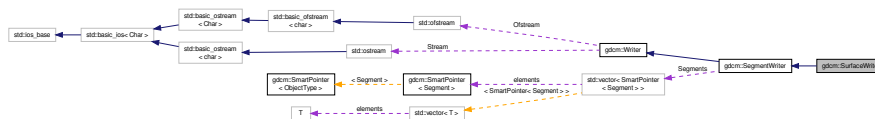
```
#include <gdcmSurfaceWriter.h>
```



```

graph LR
    A[std::basic_ostream<Char>] --> B[std::basic_ostream<char>]
    A --> C[std::basic_ios<Char>]
    C --> D[std::ios_base]

```



- virtual \approx SurfaceWrite

- Write.*

- bool **PrepareWrite** ()

- &ts)

unassigned long Number of Candidates

Additional Inherited Members

25.260.1 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

25.260.2 Constructor & Destructor Documentation

25.260.2.1 `gdcm::SurfaceWriter::SurfaceWriter ()`

25.260.2.2 `virtual gdcm::SurfaceWriter::~~SurfaceWriter () [virtual]`

25.260.3 Member Function Documentation

25.260.3.1 `void gdcm::SurfaceWriter::ComputeNumberOfSurfaces () [protected]`

25.260.3.2 `unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ()`

25.260.3.3 `bool gdcm::SurfaceWriter::PrepareWrite () [protected]`

25.260.3.4 `bool gdcm::SurfaceWriter::PrepareWritePointMacro (SmartPointer< Surface > surface, DataSet & surfaceDS, const TransferSyntax & ts) [protected]`

25.260.3.5 `void gdcm::SurfaceWriter::SetNumberOfSurfaces (const unsigned long nb)`

25.260.3.6 `bool gdcm::SurfaceWriter::Write () [virtual]`

Write.

Reimplemented from [gdcm::SegmentWriter](#).

25.260.4 Member Data Documentation

25.260.4.1 `unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]`

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

- [gdcmSurfaceWriter.h](#)

25.261 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)

25.261.1 Detailed Description

[SwapCode](#) representation.

Examples:

[TestByteSwap.cxx](#).

25.261.2 Member Enumeration Documentation

25.261.2.1 enum gdcm::SwapCode::SwapCodeType

Enumerator

Unknown

LittleEndian

BigEndian

BadLittleEndian

BadBigEndian

25.261.3 Constructor & Destructor Documentation

25.261.3.1 `gdcm::SwapCode::SwapCode (SwapCodeType sc = Unknown)` `[inline]`

25.261.4 Member Function Documentation

25.261.4.1 `static int gdcm::SwapCode::GetIndex (SwapCode const & sc)` `[static], [protected]`

25.261.4.2 `static const char* gdcm::SwapCode::GetSwapCodeString (SwapCode const & sc)` `[static]`

Referenced by `gdcm::operator<<()`.

25.261.4.3 `gdcm::SwapCode::operator SwapCode::SwapCodeType () const` `[inline]`

25.261.5 Friends And Related Function Documentation

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

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

- [gdcmSwapCode.h](#)

25.262 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)`

25.262.1 Member Function Documentation

25.262.1.1 `template<typename T > static T gdcm::SwapperDoOp::Swap (T val)` `[static]`

Referenced by `gdcm::Item::Read()`.

25.262.1.2 `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](#)

25.263 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)

25.263.1 Detailed Description

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

25.263.2 Member Function Documentation

25.263.2.1 template<typename T > static T gdcm::SwapperNoOp::Swap (T val) [inline], [static]

Referenced by gdcm::EncodingImplementation< VR::VRBINARY >::Write().

25.263.2.2 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](#)

25.264 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 * [GetLastError](#) ()
- 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 * [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)

25.264.1 Detailed Description

Class to do system operation.

OS independent functionalities

25.264.2 Member Function Documentation

25.264.2.1 static bool [gdcm::System::DeleteDirectory](#) (const char * *source*) [static]

remove a directory named source

25.264.2.2 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

25.264.2.3 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](#).

25.264.2.4 static bool gdcm::System::FilesDirectory (const char * *name*) [static]

Check whether the file specified is a directory:

Examples:

[gdcmorthoplanes.cxx](#), and [threadgdcm.cxx](#).

25.264.2.5 static bool gdcm::System::FilesSymlink (const char * *name*) [static]

Check whether name is a symlink.

25.264.2.6 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](#), and [EncapsulateFileInRawData.cxx](#).

25.264.2.7 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

25.264.2.8 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]

25.264.2.9 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

25.264.2.10 `static const char* gdcmm::System::GetCurrentModuleFileName () [static]`

Return the directory the current module is located: NOT THREAD SAFE

25.264.2.11 `static const char* gdcmm::System::GetCurrentProcessFileName () [static]`

Return the directory the current process (executable) is located: NOT THREAD SAFE

25.264.2.12 `static const char* gdcmm::System::GetCurrentResourcesDirectory () [static]`

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

25.264.2.13 `static const char* gdcmm::System::GetCurrentWorkingDirectory () [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

25.264.2.14 `static bool gdcmm::System::GetHostName (char hostname[255]) [static]`

Retrieve the hostname, only the first 255 byte are copied. This may come handy to specify the Station Name

25.264.2.15 `static const char* gdcmm::System::GetLastError () [static]`

Return the last error.

25.264.2.16 `static const char* gdcmm::System::GetLocaleCharSet () [static]`

return locale charmap

25.264.2.17 `static bool gdcmm::System::GetPermissions (const char * file, unsigned short & mode) [static],
[protected]`

NOT THREAD SAFE.

25.264.2.18 `static const char* gdcmm::System::GetTimezoneOffsetFromUTC () [static]`

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe

25.264.2.19 `static bool gdcmm::System::MakeDirectory (const char * path) [static]`

Create a directory name path.

25.264.2.20 `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)

25.264.2.21 `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](#)

25.264.2.22 `static bool gdcm::System::RemoveFile (const char * source) [static]`

remove a file named source

25.264.2.23 `static bool gdcm::System::SetPermissions (const char * file, unsigned short mode) [static],
[protected]`

25.264.2.24 `static int gdcm::System::StrCaseCmp (const char * s1, const char * s2) [static]`

consistent func for C99 spec of strcasecmp/strncasecmp

25.264.2.25 `static int gdcm::System::StrNCaseCmp (const char * s1, const char * s2, size_t n) [static]`

Precondition

`n != 0`

25.264.2.26 `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](#)

25.265 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)

25.265.1 Detailed Description

[Table](#).

25.265.2 Member Typedef Documentation

25.265.2.1 typedef std::map<[Tag](#), [TableEntry](#)> [gdcm::Table::MapTableEntry](#)

25.265.3 Constructor & Destructor Documentation

25.265.3.1 [gdcm::Table::Table](#) () `[inline]`

25.265.3.2 [gdcm::Table::~~Table](#) () `[inline]`

25.265.4 Member Function Documentation

25.265.4.1 const [TableEntry](#)& [gdcm::Table::GetTableEntry](#) (const [Tag](#) & *tag*) const `[inline]`

25.265.4.2 void [gdcm::Table::InsertEntry](#) ([Tag](#) const & *tag*, [TableEntry](#) const & *te*) `[inline]`

25.265.5 Friends And Related Function Documentation

25.265.5.1 std::ostream& [operator<<](#) (std::ostream &_os, const [Table](#) &_val) `[friend]`

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

- [gdcmTable.h](#)

25.266 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](#) ()

25.266.1 Detailed Description

[TableEntry](#).

25.266.2 Constructor & Destructor Documentation

25.266.2.1 `gdcm::TableEntry::TableEntry (const char * attribute = 0, Type const & type = Type (), const char * des = 0)`
`[inline]`

25.266.2.2 `gdcm::TableEntry::~~TableEntry ()` `[inline]`

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

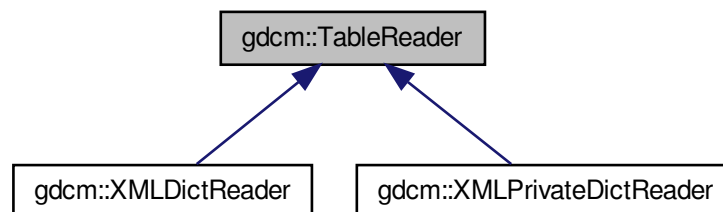
- [gdcmTableEntry.h](#)

25.267 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)

25.267.1 Detailed Description

Class for representing a [TableReader](#).

Note

This class is an empty shell meant to be derived

25.267.2 Constructor & Destructor Documentation

25.267.2.1 `gdcm::TableReader::TableReader (Defs & defs) [inline]`

25.267.2.2 `virtual gdcm::TableReader::~~TableReader () [inline],[virtual]`

25.267.3 Member Function Documentation

25.267.3.1 `virtual void gdcm::TableReader::CharacterDataHandler (const char * data, int length) [virtual]`

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

25.267.3.2 `virtual void gdcm::TableReader::EndElement (const char * name) [virtual]`

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

25.267.3.3 `const Defs& gdcm::TableReader::GetDefs () const [inline]`

25.267.3.4 `const char* gdcm::TableReader::GetFilename () [inline]`

25.267.3.5 `void gdcm::TableReader::HandleIOD (const char ** atts)`

25.267.3.6 `void gdcm::TableReader::HandleIODEntry (const char ** atts)`

25.267.3.7 `void gdcm::TableReader::HandleMacro (const char ** atts)`

25.267.3.8 `void gdcm::TableReader::HandleMacroEntry (const char ** atts)`

25.267.3.9 `void gdcm::TableReader::HandleMacroEntryDescription (const char ** atts)`

25.267.3.10 `void gdcm::TableReader::HandleModule (const char ** atts)`

- 25.267.3.11 void gdcm::TableReader::HandleModuleEntry (const char ** *atts*)
- 25.267.3.12 void gdcm::TableReader::HandleModuleEntryDescription (const char ** *atts*)
- 25.267.3.13 void gdcm::TableReader::HandleModuleInclude (const char ** *atts*)
- 25.267.3.14 int gdcm::TableReader::Read ()
- 25.267.3.15 void gdcm::TableReader::SetFilename (const char * *filename*) [inline]
- 25.267.3.16 virtual void gdcm::TableReader::StartElement (const char * *name*, const char ** *atts*) [virtual]

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

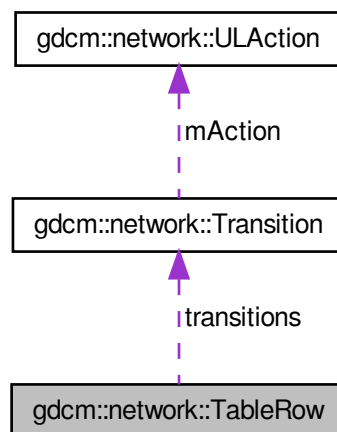
The documentation for this class was generated from the following file:

- [gdcmTableReader.h](#)

25.268 gdcm::network::TableRow Class Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::network::TableRow:



Public Member Functions

- [TableRow](#) ()
- [~TableRow](#) ()

Public Attributes

- [Transition](#) * [transitions](#) [[cMaxStateID](#)]

25.268.1 Constructor & Destructor Documentation

25.268.1.1 `gdcm::network::TableRow::TableRow ()` [[inline](#)]

References `gdcm::network::cMaxStateID`, and `transitions`.

25.268.1.2 `gdcm::network::TableRow::~~TableRow ()` [[inline](#)]

References `gdcm::network::cMaxStateID`, and `transitions`.

25.268.2 Member Data Documentation

25.268.2.1 `Transition*` `gdcm::network::TableRow::transitions`[[cMaxStateID](#)]

Referenced by `TableRow()`, and `~TableRow()`.

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

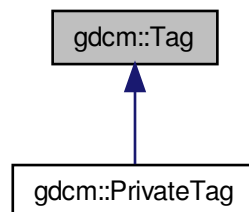
- [gdcmULTransitionTable.h](#)

25.269 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 ctor 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 [PrintAsPipeSeparatedString](#) () const
- template<typename TSwap >
 - std::istream & [Read](#) (std::istream &is)
 - [Read](#) a tag from binary representation.
- bool [ReadFromCommaSeparatedString](#) (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`)

25.269.1 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](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateJPIPDataSet.cxx](#), [DumpToSQLite3.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [PublicDict.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReformatFile.cs](#), [rle2img.cxx](#), [SimplePrintPatientName.cs](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), [StandardizeFiles.cs](#), [StreamImageReaderTest.cxx](#), [TraverseModules.cxx](#), and [VolumeSorter.cxx](#).

25.269.2 Constructor & Destructor Documentation

25.269.2.1 `gdcm::Tag::Tag (uint16_t group, uint16_t element)` `[inline]`

Constructor with 2*`uint16_t`.

25.269.2.2 `gdcm::Tag::Tag (uint32_t tag = 0)` `[inline]`

Constructor with 1*`uint32_t` Prefer the ctor that takes two `uint16_t`.

25.269.2.3 `gdcm::Tag::Tag (const Tag &_val)` `[inline]`

References tag.

25.269.3 Member Function Documentation

25.269.3.1 `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::SequenceOfFragments::ReadValue()`, and `SetPrivateCreator()`.

25.269.3.2 `uint32_t gdcm::Tag::GetElementTag () const [inline]`

Returns the full tag value of the given [Tag](#).

25.269.3.3 `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()`, and `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

25.269.3.4 `uint32_t gdcm::Tag::GetLength () const [inline]`

return the length of tag (read: size on disk)

25.269.3.5 `Tag gdcm::Tag::GetPrivateCreator () const [inline]`

Return the Private Creator Data [Element](#) tag of a private data element.

References `SetElement()`.

25.269.3.6 `bool gdcm::Tag::IsGroupLength () const [inline]`

return whether the tag correspond to a group length tag:

25.269.3.7 `bool gdcm::Tag::IsGroupXX (const Tag & t) const [inline]`

e.g 6002,3000 belong to groupXX: 6000,3000

References `GetElement()`, `GetGroup()`, and `IsPrivate()`.

25.269.3.8 `bool gdcm::Tag::IsIllegal () const [inline]`

return if the tag is considered to be an illegal tag

25.269.3.9 `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()`.

25.269.3.10 `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](#).

25.269.3.11 `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...

25.269.3.12 `bool gdcM::Tag::operator!= (const Tag &_val) const [inline]`

References tag.

25.269.3.13 `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.

25.269.3.14 `bool gdcM::Tag::operator<= (const Tag &t2) const [inline]`

25.269.3.15 `Tag& gdcM::Tag::operator= (const Tag &_val) [inline]`

References tag.

25.269.3.16 `bool gdcM::Tag::operator== (const Tag &_val) const [inline]`

References tag.

25.269.3.17 `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)

25.269.3.18 `uint16_t& gdcm::Tag::operator[] (const unsigned int &_id) [inline]`

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

25.269.3.19 `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](#)

25.269.3.20 `template<typename TSwap > std::istream& gdcm::Tag::Read (std::istream &is) [inline]`

Read a tag from binary representation.

25.269.3.21 `bool gdcm::Tag::ReadFromCommaSeparatedString (const char * str)`

Read from a comma separated string. This is a highly user oriented function, the string should be formatted as: 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

25.269.3.22 `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](#)

25.269.3.23 `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>>()`.

25.269.3.24 `void gdcm::Tag::SetElementTag (uint16_t group, uint16_t element) [inline]`

Sets the 'Group number' & '[Element](#) number' of the given [Tag](#).

25.269.3.25 `void gdcm::Tag::SetElementTag (uint32_t tag) [inline]`

Sets the full tag value of the given [Tag](#).

25.269.3.26 `void gdcm::Tag::SetGroup (uint16_t group) [inline]`

Sets the 'Group number' of the given [Tag](#).

Referenced by `gdcm::operator>>()`.

25.269.3.27 `void gdcm::Tag::SetPrivateCreator (Tag const & t) [inline]`

Set private creator:

Examples:

[DuplicatePCDE.cxx](#).

References `GetElement()`, and `IsPrivate()`.

25.269.3.28 `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()`.

25.269.4 Friends And Related Function Documentation

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

25.269.4.2 `std::istream& operator>> (std::istream & _is, Tag & _val) [friend]`

25.269.5 Member Data Documentation

25.269.5.1 `char gdcm::Tag::bytes[4]`

25.269.5.2 `uint32_t gdcm::Tag::tag`

Referenced by `operator!=()`, `operator<()`, `operator=()`, `operator==()`, and `Tag()`.

25.269.5.3 `uint16_t gdcm::Tag::tags[2]`

Referenced by `operator<()`.

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

- [gdcmTag.h](#)

25.270 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.

25.270.1 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

25.270.2 Constructor & Destructor Documentation

25.270.2.1 [gdcm::TagPath::TagPath](#) ()

25.270.2.2 [gdcm::TagPath::~~TagPath](#) ()

25.270.3 Member Function Documentation

25.270.3.1 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

25.270.3.2 bool [gdcm::TagPath::ConstructFromTagList](#) ([Tag](#) const * *l*, unsigned int *n*)

Construct from a list of tags.

25.270.3.3 static bool [gdcm::TagPath::IsValid](#) (const char * *path*) [static]

Return if path is valid or not.

25.270.3.4 void `gdcm::TagPath::Print` (std::ostream &) const

25.270.3.5 bool `gdcm::TagPath::Push` (Tag const & t)

25.270.3.6 bool `gdcm::TagPath::Push` (unsigned int *itemnum*)

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

- [gdcmTagPath.h](#)

25.271 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 [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.

25.271.1 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

25.271.2 Member Typedef Documentation

25.271.2.1 `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

25.271.2.2 `typedef const char* const(* gdcm::Testing::MediaStorageDataFilesType)[2]`

return the table that map the media storage (as string) of a filename (gdcmData)

25.271.3 Constructor & Destructor Documentation

25.271.3.1 `gdcm::Testing::Testing ()` `[inline]`

25.271.3.2 `gdcm::Testing::~~Testing ()` `[inline]`

25.271.4 Member Function Documentation

25.271.4.1 `static bool gdcm::Testing::ComputeFileMD5 (const char * filename, char digest_str[33])` `[static]`

25.271.4.2 `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.

25.271.4.3 `static const char* gdcM::Testing::GetDataExtraRoot ()` `[static]`

Return the GDCM DATA EXTRA ROOT.

Examples:

[DiscriminateVolume.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

25.271.4.4 `static const char* gdcM::Testing::GetDataRoot ()` `[static]`

Return the GDCM DATA ROOT.

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), and [Magnify-File.cxx](#).

25.271.4.5 `static const char* gdcM::Testing::GetFileName (unsigned int file)` `[static]`

25.271.4.6 `static const char* const* gdcM::Testing::GetFileNames ()` `[static]`

return the table of fullpath to gdcMData DICOM files:

Examples:

[TestReader.cxx](#).

25.271.4.7 `static int gdcM::Testing::GetLossyFlagFromFile (const char * filepath)` `[static]`

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

25.271.4.8 `static const char* const* gdcM::Testing::GetMD5DataImage (unsigned int file)` `[static]`

25.271.4.9 `static MD5DataImagesType gdcM::Testing::GetMD5DataImages ()` `[static]`

25.271.4.10 `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.

25.271.4.11 `static const char* gdcmm::Testing::GetMD5FromFile (const char * filepath) [static]`

25.271.4.12 `static const char* const* gdcmm::Testing::GetMediaStorageDataFile (unsigned int file) [static]`

25.271.4.13 `static MediaStorageDataFileType gdcmm::Testing::GetMediaStorageDataFiles () [static]`

25.271.4.14 `static const char* gdcmm::Testing::GetMediaStorageFromFile (const char * filepath) [static]`

Examples:

[TestReader.cxx](#).

25.271.4.15 `static unsigned int gdcmm::Testing::GetNumberOfFileNames () [static]`

25.271.4.16 `static unsigned int gdcmm::Testing::GetNumberOfMD5DataImages () [static]`

25.271.4.17 `static unsigned int gdcmm::Testing::GetNumberOfMediaStorageDataFiles () [static]`

25.271.4.18 `static const char* gdcmm::Testing::GetPixelSpacingDataRoot () [static]`

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

25.271.4.19 `static std::streamoff gdcmm::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

25.271.4.20 `static const char* gdcmm::Testing::GetSourceDirectory () [static]`

25.271.4.21 `static std::streamoff gdcmm::Testing::GetStreamOffsetFromFile (const char * filepath) [static]`

Return the offset of the very first pixel cell in the PixelData -1 if not found

25.271.4.22 `static const char* gdcmm::Testing::GetTempDirectory (const char * subdir = 0) [static]`

NOT THREAD SAFE Returns the temp directory as used in testing needing to output data:

25.271.4.23 `static const wchar_t* gdcmm::Testing::GetTempDirectoryW (const wchar_t * subdir = 0) [static]`

NOT THREAD SAFE.

25.271.4.24 `static const char* gdcmm::Testing::GetTempFilename (const char * filename, const char * subdir = 0) [static]`

NOT THREAD SAFE.

25.271.4.25 `static const wchar_t* gdcmm::Testing::GetTempFilenameW (const wchar_t * filename, const wchar_t * subdir = 0) [static]`

NOT THREAD SAFE.

25.271.4.26 `void gdcmm::Testing::Print (std::ostream & os = std::cout)`

Print.

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

- [gdcmmTesting.h](#)

25.272 gdcmm::Trace Class Reference

[Trace](#).

```
#include <gdcmmTrace.h>
```

Public Member Functions

- [Trace](#) ()
- [~Trace](#) ()

Static Public Member Functions

- static void [DebugOff](#) ()
- static void [DebugOn](#) ()
- static void [ErrorOff](#) ()
- static void [ErrorOn](#) ()
- static bool [GetDebugFlag](#) ()
- static std::ostream & [GetDebugStream](#) ()
- static bool [GetErrorFlag](#) ()
- static std::ostream & [GetErrorStream](#) ()
- static std::ostream & [GetStream](#) ()
- static bool [GetWarningFlag](#) ()
- static std::ostream & [GetWarningStream](#) ()
- static void [SetDebug](#) (bool debug)
Turn debug messages on (default: false)
- static void [SetDebugStream](#) (std::ostream &os)
Explicitly set the stream which receive Debug messages:
- static void [SetError](#) (bool debug)
Turn error messages on (default: true)
- static void [SetErrorStream](#) (std::ostream &os)
Explicitly set the stream which receive Error messages:
- static void [SetStream](#) (std::ostream &os)
- static void [SetStreamToFile](#) (const char *filename)
- static void [SetWarning](#) (bool debug)
Turn warning messages on (default: true)
- static void [SetWarningStream](#) (std::ostream &os)
Explicitly set the stream which receive Warning messages:
- static void [WarningOff](#) ()
- static void [WarningOn](#) ()

25.272.1 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.

25.272.2 Constructor & Destructor Documentation

25.272.2.1 `gdcm::Trace::Trace ()`

25.272.2.2 `gdcm::Trace::~~Trace ()`

25.272.3 Member Function Documentation

25.272.3.1 `static void gdcm::Trace::DebugOff ()` `[static]`

Examples:

[TestReader.cxx](#).

25.272.3.2 `static void gdcm::Trace::DebugOn ()` `[static]`

Examples:

[Fake_Image_Using_Stream_Image_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

25.272.3.3 `static void gdcm::Trace::ErrorOff ()` `[static]`

25.272.3.4 `static void gdcm::Trace::ErrorOn ()` `[static]`

25.272.3.5 `static bool gdcm::Trace::GetDebugFlag ()` `[static]`

25.272.3.6 `static std::ostream& gdcm::Trace::GetDebugStream ()` `[static]`

25.272.3.7 `static bool gdcm::Trace::GetErrorFlag ()` `[static]`

25.272.3.8 `static std::ostream& gdcm::Trace::GetErrorStream ()` `[static]`

25.272.3.9 `static std::ostream& gdcm::Trace::GetStream ()` `[static]`

25.272.3.10 `static bool gdcm::Trace::GetWarningFlag () [static]`

25.272.3.11 `static std::ostream& gdcm::Trace::GetWarningStream () [static]`

25.272.3.12 `static void gdcm::Trace::SetDebug (bool debug) [static]`

Turn debug messages on (default: false)

Examples:

[DumpToSQLITE3.cxx](#).

25.272.3.13 `static void gdcm::Trace::SetDebugStream (std::ostream & os) [static]`

Explicitely set the stream which receive Debug messages:

25.272.3.14 `static void gdcm::Trace::SetError (bool debug) [static]`

Turn error messages on (default: true)

25.272.3.15 `static void gdcm::Trace::SetErrorStream (std::ostream & os) [static]`

Explicitely set the stream which receive Error messages:

Examples:

[CStoreQtProgress.cxx](#).

25.272.3.16 `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:

25.272.3.17 `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)

25.272.3.18 `static void gdcm::Trace::SetWarning (bool debug) [static]`

Turn warning messages on (default: true)

Examples:

[DumpToSQLITE3.cxx](#).

25.272.3.19 `static void gdcm::Trace::SetWarningStream (std::ostream & os) [static]`

Explicitely set the stream which receive Warning messages:

25.272.3.20 static void gdcm::Trace::WarningOff () [static]

Examples:

[TestReader.cxx](#).

25.272.3.21 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](#)

25.273 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](#) (TType 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 TType](#) () const

Static Public Member Functions

- static const char * [GetTSString](#) (TType ts)
- static [TType](#) [GetTSType](#) (const char *str)

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [TransferSyntax](#) &ts)

25.273.1 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](#).

25.273.2 Member Enumeration Documentation

25.273.2.1 enum gdcm::TransferSyntax::NegociatedType

Enumerator

Unknown

Explicit

Implicit

25.273.2.2 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

25.273.3 Constructor & Destructor Documentation

25.273.3.1 gdcm::TransferSyntax::TransferSyntax (TSType type = ImplicitVRLittleEndian) [inline]

25.273.4 Member Function Documentation

25.273.4.1 bool gdcm::TransferSyntax::CanStoreLossy () const

return if TransFer Syntax Allow storing of Lossy Pixel Data

25.273.4.2 **NegotiatedType** gdcm::TransferSyntax::GetNegociatedType () const

25.273.4.3 const char* gdcm::TransferSyntax::GetString () const [inline]

References GetTSString().

25.273.4.4 **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.

25.273.4.5 static const char* gdcm::TransferSyntax::GetTSString (TSType ts) [static]

Examples:

[LargeVRDSExplicit.cxx](#).

Referenced by GetString(), and gdcm::operator<<().

25.273.4.6 static TSType gdcm::TransferSyntax::GetTSType (const char * str) [static]

25.273.4.7 bool gdcm::TransferSyntax::IsEncapsulated () const

Examples:

[ExtractIconFromFile.cxx](#).

25.273.4.8 bool gdcm::TransferSyntax::IsEncoded () const

25.273.4.9 bool gdcm::TransferSyntax::IsExplicit () const

25.273.4.10 bool gdcm::TransferSyntax::IsImplicit () const

25.273.4.11 bool gdcm::TransferSyntax::IsLossless () const

Return if the transfer syntax algorithm is a lossless algorithm

25.273.4.12 bool gdcm::TransferSyntax::IsLossy () const

Return if the transfer syntax algorithm is a lossy algorithm

25.273.4.13 bool gdcm::TransferSyntax::IsValid () const [inline]

25.273.4.14 gdcm::TransferSyntax::operator TSType () const [inline]

25.273.5 Friends And Related Function Documentation

25.273.5.1 `std::ostream& operator<< (std::ostream & os, const TransferSyntax & ts)` `[friend]`

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

- [gdcmTransferSyntax.h](#)

25.274 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

25.274.1 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](#)

25.274.2 Constructor & Destructor Documentation

25.274.2.1 `gdcm::network::TransferSyntaxSub::TransferSyntaxSub ()`

25.274.3 Member Function Documentation

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

25.274.3.2 `bool gdcm::network::TransferSyntaxSub::operator== (const TransferSyntaxSub & ts) const` `[inline]`

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

25.274.3.4 `std::istream& gdcm::network::TransferSyntaxSub::Read (std::istream & is)`

25.274.3.5 `void gdcm::network::TransferSyntaxSub::SetName (const char * name)`

25.274.3.6 `void gdcm::network::TransferSyntaxSub::SetNameFromUID (UIDs::TSName tsname)`

25.274.3.7 `size_t gdcmm::network::TransferSyntaxSub::Size () const`

25.274.3.8 `const std::ostream& gdcmm::network::TransferSyntaxSub::Write (std::ostream & os) const`

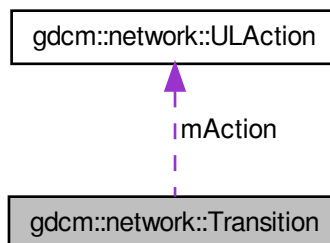
The documentation for this class was generated from the following file:

- [gdcmmTransferSyntaxSub.h](#)

25.275 gdcmm::network::Transition Struct Reference

```
#include <gdcmmULTransitionTable.h>
```

Collaboration diagram for gdcmm::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](#)

25.275.1 Constructor & Destructor Documentation

25.275.1.1 `gdcm::network::Transition::Transition () [inline]`

References `gdcm::network::eStaDoesNotExist`, `mAction`, and `mEnd`.

Referenced by `MakeNew()`.

25.275.1.2 `gdcm::network::Transition::~~Transition () [inline]`

References `mAction`.

25.275.1.3 `gdcm::network::Transition::Transition (int inEndState, ULAction * inAction) [inline]`

References `mAction`, and `mEnd`.

25.275.2 Member Function Documentation

25.275.2.1 `static Transition* gdcm::network::Transition::MakeNew (int inEndState, ULAction * inAction) [inline], [static]`

References `Transition()`.

25.275.3 Member Data Documentation

25.275.3.1 `ULAction* gdcm::network::Transition::mAction`

Referenced by `Transition()`, and `~Transition()`.

25.275.3.2 `int gdcm::network::Transition::mEnd`

Referenced by `Transition()`.

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

- [gdcmULTransitionTable.h](#)

25.276 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)

25.276.1 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](#).

25.276.2 Member Enumeration Documentation

25.276.2.1 enum [gdcm::Type::TypeType](#)

Enumerator

T1

T1C

T2

T2C

T3

UNKNOWN

25.276.3 Constructor & Destructor Documentation

25.276.3.1 `gdcm::Type::Type (TypeType type = UNKNOWN)` `[inline]`

25.276.4 Member Function Documentation

25.276.4.1 `static const char* gdcm::Type::GetTypeString (TypeType type)` `[static]`

Referenced by `gdcm::operator<<()`.

25.276.4.2 `static TypeType gdcm::Type::GetTypeType (const char * type)` `[static]`

Referenced by `gdcm::ModuleEntry::ModuleEntry()`.

25.276.4.3 `gdcm::Type::operator TypeType () const` `[inline]`

25.276.5 Friends And Related Function Documentation

25.276.5.1 `std::ostream& operator<< (std::ostream & os, const Type & vr)` `[friend]`

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

- [gdcmType.h](#)

25.277 gdcm::UI Struct Reference

```
#include <gdcmVR.h>
```

Public Attributes

- char [Internal](#) [64+1]

Friends

- std::ostream & [operator<<](#) (std::ostream &_os, const [UI](#) &_val)

25.277.1 Friends And Related Function Documentation

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

25.277.2 Member Data Documentation

25.277.2.1 `char gdcm::UI::Internal[64+1]`

Referenced by `gdcm::operator<<()`.

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

- [gdcmlVR.h](#)

25.278 gdcml::UIDGenerator Class Reference

Class for generating unique UID.

```
#include <gdcmlUIDGenerator.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)

25.278.1 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 UID Generator will simply forget the value and create a new UID.

Examples:

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GetSubSequenceData.cxx](#), [StreamImageReaderTest.cxx](#), and [uid_unique.cxx](#).

25.278.2 Constructor & Destructor Documentation

25.278.2.1 gdcml::UIDGenerator::UIDGenerator () [inline]

By default the root of a UID is a GDCM Root...

25.278.3 Member Function Documentation

25.278.3.1 `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](#), [StreamImageReader-Test.cxx](#), and [uid_unique.cxx](#).

25.278.3.2 `static bool gdcm::UIDGenerator::GenerateUUID (unsigned char * uuid_data) [static], [protected]`

25.278.3.3 `static const char* gdcm::UIDGenerator::GetGDCMUID () [static]`

Return the default (GDCM) root UID:

25.278.3.4 `static const char* gdcm::UIDGenerator::GetRoot () [static]`

25.278.3.5 `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`)

25.278.3.6 `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 responsibility for guaranteeing uniqueness of [UIDs](#) when the root is longer than 26 bytes.

Examples:

[uid_unique.cxx](#).

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

- [gdcmUIDGenerator.h](#)

25.279 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,
SPM2PDFFrameofReference = 40,
SPM2EPIFrameofReference = 41,
SPM2FILT1FrameofReference = 42,
SPM2PETFrameofReference = 43,
SPM2TRANSMFrameofReference = 44,
SPM2SPECTFrameofReference = 45,
SPM2GRAYFrameofReference = 46,
SPM2WHITEFrameofReference = 47,
SPM2CSFFFrameofReference = 48,
SPM2BRAINMASKFrameofReference = 49,
SPM2AVG305T1FrameofReference = 50,
SPM2AVG152T1FrameofReference = 51,
SPM2AVG152T2FrameofReference = 52,
SPM2AVG152PDFFrameofReference = 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)

25.279.1 Detailed Description

all known uids

Examples:

[GenerateStandardSOPClasses.cxx](#).

25.279.2 Member Typedef Documentation

25.279.2.1 `typedef const char* const(* gdcmm::UIDs::TransferSyntaxStringsType)[2]`

25.279.3 Member Enumeration Documentation

25.279.3.1 `enum gdcmm::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
JPIPRreferenced
JPIPRreferencedDeflate
MPEG2MainProfileMainLevel
RLELossless
RFC2557MIMEencapsulation
XMLEncoding
MediaStorageDirectoryStorage
TalairachBrainAtlasFrameofReference
SPM2T1FrameofReference
SPM2T2FrameofReference
SPM2PDFFrameofReference
SPM2EPIFrameofReference
SPM2FIL T1FrameofReference
SPM2PETFrameofReference
SPM2TRANSMFrameofReference
SPM2SPECTFrameofReference
SPM2GRAYFrameofReference
SPM2WHITEFrameofReference
SPM2CSFFFrameofReference
SPM2BRAINMASKFrameofReference
SPM2AVG305T1FrameofReference
SPM2AVG152T1FrameofReference
SPM2AVG152T2FrameofReference
SPM2AVG152PDFFrameofReference

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
MRIImageStorage
EnhancedMRIImageStorage
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
EnhancedXRImageStorage
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
EncapsulatedCDASStorage
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

25.279.3.2 enum gdcmm::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

```

25.279.4 Member Function Documentation

25.279.4.1 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<<()`.

25.279.4.2 `static unsigned int gdcml::UIDs::GetNumberOfTransferSyntaxStrings () [static]`

25.279.4.3 `const char* gdcml::UIDs::GetString () const`

When object is Initialize function return the uid return NULL when not initialized

Examples:

[GenerateStandardSOPClasses.cxx](#).

Referenced by `gdcml::operator<<()`.

25.279.4.4 `static const char* const* gdcml::UIDs::GetTransferSyntaxString (unsigned int ts) [static]`

25.279.4.5 `static TransferSyntaxStringsType gdcml::UIDs::GetTransferSyntaxStrings () [static]`

25.279.4.6 `static const char* gdcml::UIDs::GetUIDName (unsigned int ts) [static]`

25.279.4.7 `static const char* gdcml::UIDs::GetUIDString (unsigned int ts) [static]`

25.279.4.8 `gdcml::UIDs::operator TSType () const [inline]`

25.279.4.9 `bool gdcml::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:

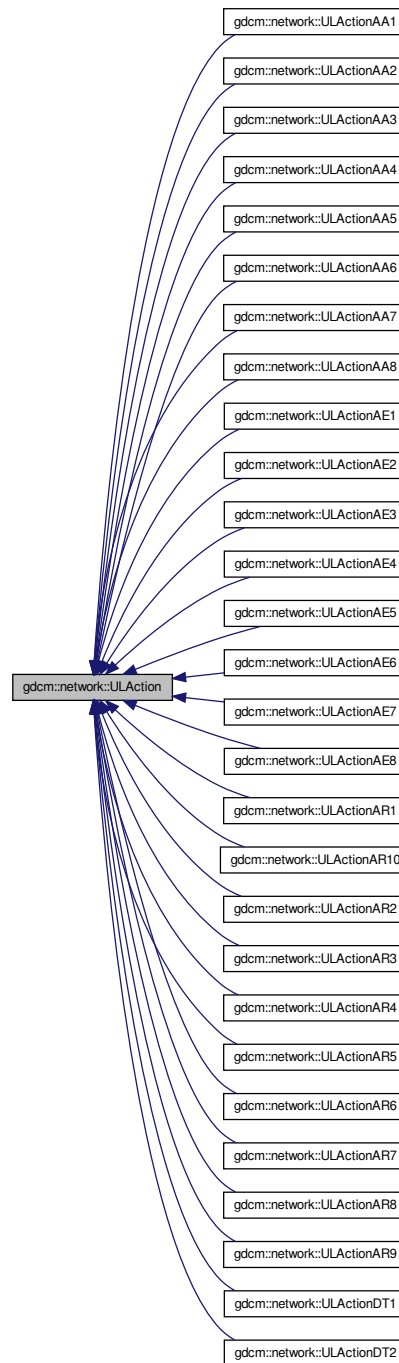
- [gdcmlUIDs.h](#)

25.280 gdcml::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 <gdcmlULAction.h>
```

Inheritance diagram for gdcn::network::ULAction:



Public Member Functions

- [ULAction](#) ()
- virtual [~ULAction](#) ()

- virtual [EStateID PerformAction](#) ([Subject *s](#), [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)=0

25.280.1 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 gdcmm-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.

25.280.2 Constructor & Destructor Documentation

25.280.2.1 `gdcmm::network::ULAction::ULAction () [inline]`

25.280.2.2 `virtual gdcmm::network::ULAction::~~ULAction () [inline], [virtual]`

25.280.3 Member Function Documentation

25.280.3.1 `virtual EStateID gdcmm::network::ULAction::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [pure virtual]`

Implemented in [gdcmm::network::ULActionAR10](#), [gdcmm::network::ULActionAR9](#), [gdcmm::network::ULActionAE8](#), [gdcmm::network::ULActionAA8](#), [gdcmm::network::ULActionAR8](#), [gdcmm::network::ULActionAE7](#), [gdcmm::network::ULActionAA7](#), [gdcmm::network::ULActionAR7](#), [gdcmm::network::ULActionAE6](#), [gdcmm::network::ULActionAA6](#), [gdcmm::network::ULActionAR6](#), [gdcmm::network::ULActionAA5](#), [gdcmm::network::ULActionAE5](#), [gdcmm::network::ULActionAR5](#), [gdcmm::network::ULActionAA4](#), [gdcmm::network::ULActionAE4](#), [gdcmm::network::ULActionAR4](#), [gdcmm::network::ULActionAA3](#), [gdcmm::network::ULActionAE3](#), [gdcmm::network::ULActionAR3](#), [gdcmm::network::ULActionAA2](#), [gdcmm::network::ULActionAE2](#), [gdcmm::network::ULActionAR2](#), [gdcmm::network::ULActionDT2](#), [gdcmm::network::ULActionAA1](#), [gdcmm::network::ULActionAE1](#), [gdcmm::network::ULActionAR1](#), and [gdcmm::network::ULActionDT1](#).

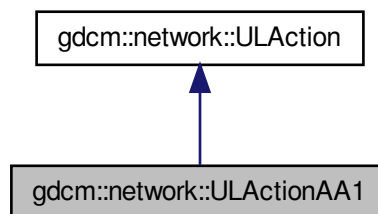
The documentation for this class was generated from the following file:

- [gdcmmULAction.h](#)

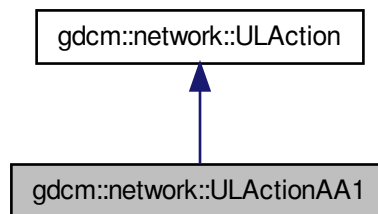
25.281 gdcmm::network::ULActionAA1 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA1:



Collaboration diagram for gdcmm::network::ULActionAA1:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.281.1 Member Function Documentation

25.281.1.1 **EStateID** gdcmm::network::ULActionAA1::PerformAction (**Subject** * s, **ULEvent** & *inEvent*, **ULConnection** & *inConnection*, bool & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements [gdcmm::network::ULAction](#).

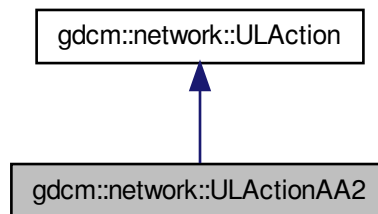
The documentation for this class was generated from the following file:

- [gdcmlActionAA.h](#)

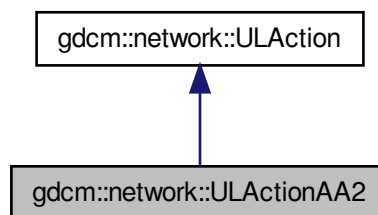
25.282 gdcmlnetwork::ULActionAA2 Class Reference

```
#include <gdcmlActionAA.h>
```

Inheritance diagram for gdcmlnetwork::ULActionAA2:



Collaboration diagram for gdcmlnetwork::ULActionAA2:



Public Member Functions

- [EStateID](#) [PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.282.1 Member Function Documentation

25.282.1.1 **EStateID** gdcmm::network::ULActionAA2::PerformAction (**Subject** * *s*, **UEvent** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements [gdcmm::network::ULAction](#).

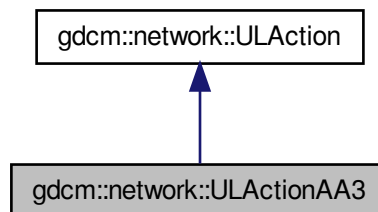
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

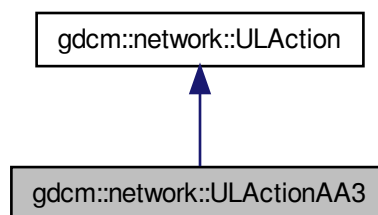
25.283 gdcmm::network::ULActionAA3 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA3:



Collaboration diagram for gdcmm::network::ULActionAA3:



Public Member Functions

- **EStateID** [PerformAction](#) ([Subject](#) **s*, [UEvent](#) &*inEvent*, [ULConnection](#) &*inConnection*, **bool** &*outWaitingForEvent*, [EEventID](#) &*outRaisedEvent*)

25.283.1 Member Function Documentation

25.283.1.1 `EStateID gdcm::network::ULActionAA3::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

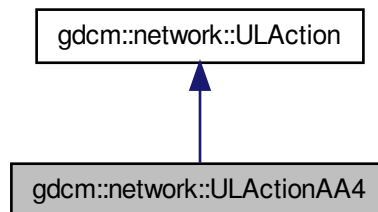
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

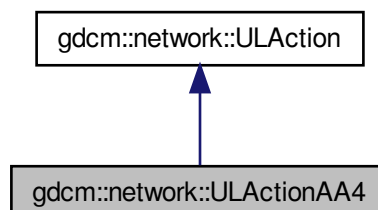
25.284 gdcm::network::ULActionAA4 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for `gdcm::network::ULActionAA4`:



Collaboration diagram for `gdcm::network::ULActionAA4`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.284.1 Member Function Documentation

25.284.1.1 [EStateID](#) `gdcm::network::ULActionAA4::PerformAction (Subject * s, ULError & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` `[virtual]`

Implements [gdcm::network::ULAction](#).

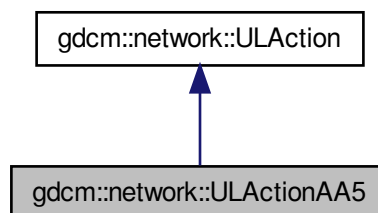
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

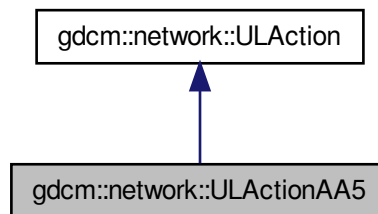
25.285 gdcm::network::ULActionAA5 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.285.1 Member Function Documentation

25.285.1.1 **EStateID** `gdcm::network::ULActionAA5::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcm::network::ULAction](#).

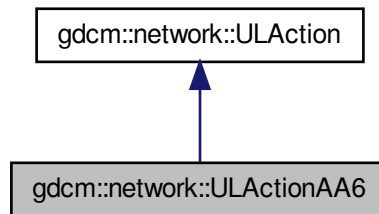
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

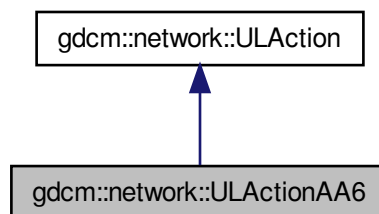
25.286 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA6:



Collaboration diagram for gdcm::network::ULActionAA6:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.286.1 Member Function Documentation

25.286.1.1 [EStateID](#) [gdcm::network::ULActionAA6::PerformAction](#) ([Subject](#) * s, [ULError](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [[virtual](#)]

Implements [gdcm::network::ULAction](#).

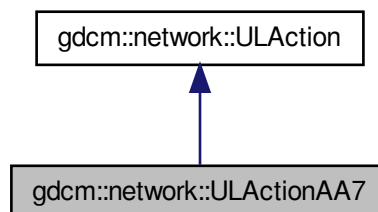
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

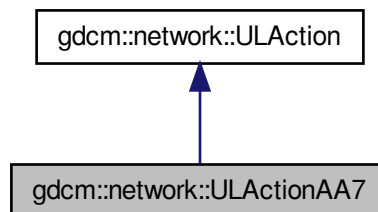
25.287 gdcmm::network::ULActionAA7 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA7:



Collaboration diagram for gdcmm::network::ULActionAA7:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.287.1 Member Function Documentation

25.287.1.1 [EStateID gdcmm::network::ULActionAA7::PerformAction](#) ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) `[virtual]`

Implements [gdcmm::network::ULAction](#).

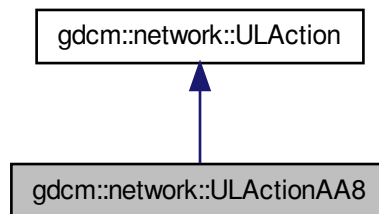
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

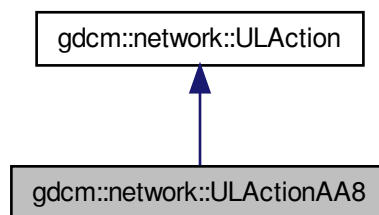
25.288 gdcmm::network::ULActionAA8 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA8:



Collaboration diagram for gdcmm::network::ULActionAA8:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.288.1 Member Function Documentation

25.288.1.1 **EStateID** `gdcm::network::ULActionAA8::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcm::network::ULAction](#).

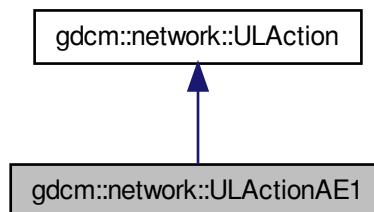
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

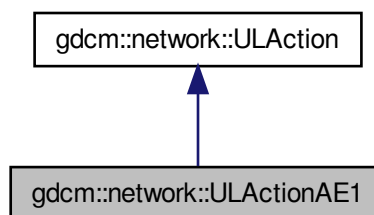
25.289 `gdcm::network::ULActionAE1` Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.289.1 Member Function Documentation

25.289.1.1 **EStateID** gdcmm::network::ULActionAE1::PerformAction (**Subject** * *s*, **ULEvent** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements [gdcmm::network::ULAction](#).

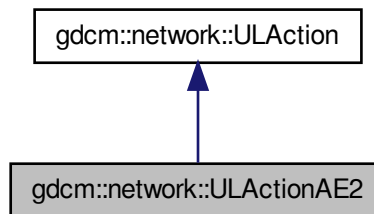
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

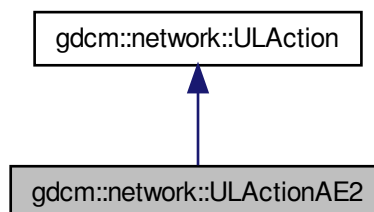
25.290 gdcmm::network::ULActionAE2 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE2:



Collaboration diagram for gdcmm::network::ULActionAE2:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.290.1 Member Function Documentation

25.290.1.1 [EStateID](#) `gdc::network::ULActionAE2::PerformAction (Subject * s, ULError & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdc::network::ULAction](#).

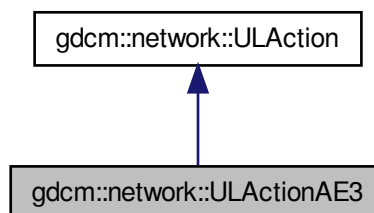
The documentation for this class was generated from the following file:

- [gdcULActionAE.h](#)

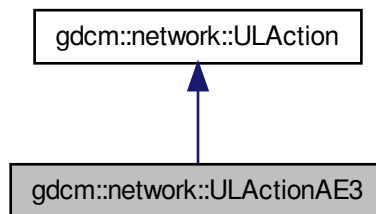
25.291 [gdc::network::ULActionAE3](#) Class Reference

```
#include <gdcULActionAE.h>
```

Inheritance diagram for `gdc::network::ULActionAE3`:



Collaboration diagram for gdcm::network::ULActionAE3:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.291.1 Member Function Documentation

25.291.1.1 `EStateID gdcm::network::ULActionAE3::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

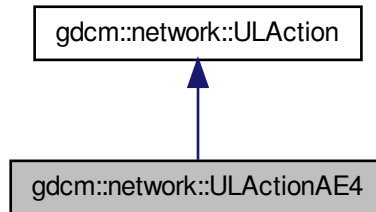
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

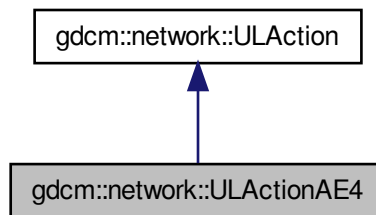
25.292 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for `gdcm::network::ULActionAE4`:



Collaboration diagram for `gdcm::network::ULActionAE4`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.292.1 Member Function Documentation

25.292.1.1 [EStateID](#) `gdcm::network::ULActionAE4::PerformAction` ([Subject](#) *s, [ULError](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) [[virtual](#)]

Implements [gdcm::network::ULAction](#).

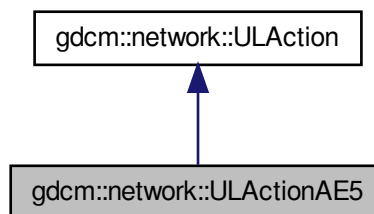
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

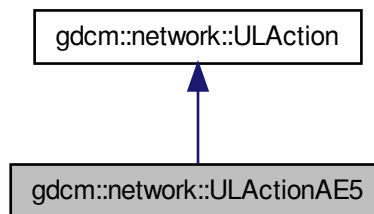
25.293 gdcmm::network::ULActionAE5 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE5:



Collaboration diagram for gdcmm::network::ULActionAE5:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.293.1 Member Function Documentation

25.293.1.1 [EStateID gdcmm::network::ULActionAE5::PerformAction](#) ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) `[virtual]`

Implements [gdcmm::network::ULAction](#).

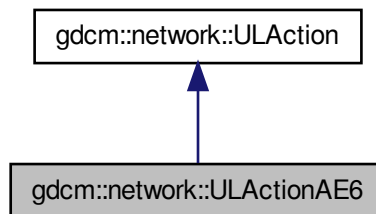
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

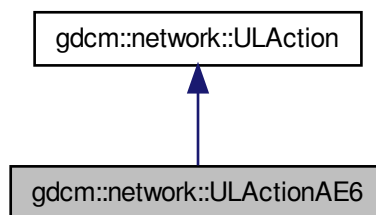
25.294 gdcm::network::ULActionAE6 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.294.1 Member Function Documentation

25.294.1.1 **EStateID** gdcm::network::ULActionAE6::PerformAction (**Subject** * *s*, **ULError** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements [gdcm::network::ULAction](#).

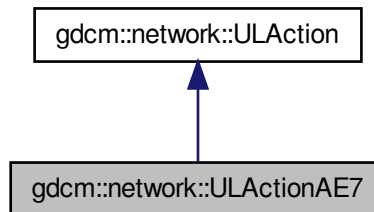
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

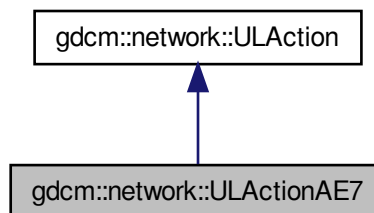
25.295 gdcm::network::ULActionAE7 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE7:



Collaboration diagram for gdcm::network::ULActionAE7:



Public Member Functions

- **EStateID** [PerformAction](#) (**Subject** **s*, **ULError** &*inEvent*, **ULConnection** &*inConnection*, **bool** &*outWaitingForEvent*, **EEventID** &*outRaisedEvent*)

25.295.1 Member Function Documentation

25.295.1.1 `EStateID gdcmm::network::ULActionAE7::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcmm::network::ULAction](#).

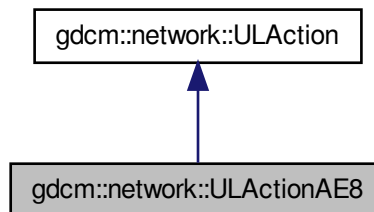
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

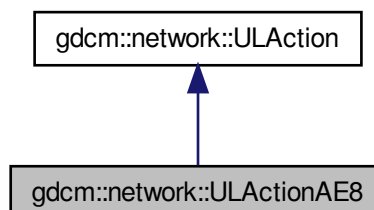
25.296 gdcmm::network::ULActionAE8 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for `gdcmm::network::ULActionAE8`:



Collaboration diagram for `gdcmm::network::ULActionAE8`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.296.1 Member Function Documentation

25.296.1.1 [EStateID](#) `gdcm::network::ULActionAE8::PerformAction (Subject * s, ULError & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcm::network::ULAction](#).

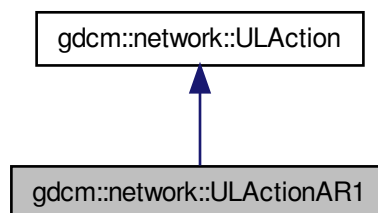
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

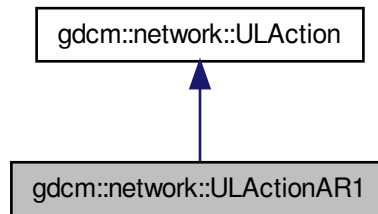
25.297 gdcm::network::ULActionAR1 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.297.1 Member Function Documentation

25.297.1.1 **EStateID** `gdcm::network::ULActionAR1::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcm::network::ULAction](#).

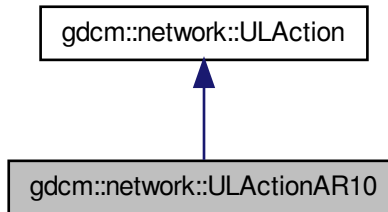
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

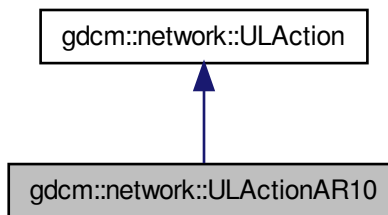
25.298 gdcm::network::ULActionAR10 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.298.1 Member Function Documentation

25.298.1.1 **EStateID** gdcm::network::ULActionAR10::PerformAction (**Subject** * s, **ULEvent** & *inEvent*, **ULConnection** & *inConnection*, bool & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements [gdcm::network::ULAction](#).

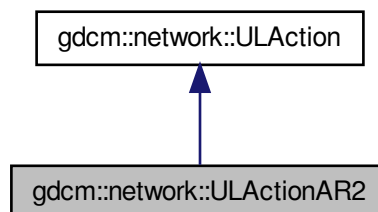
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

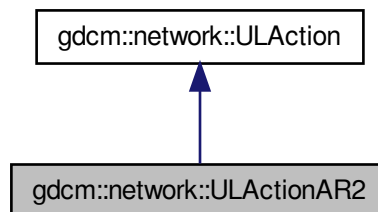
25.299 gdcm::network::ULActionAR2 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.299.1 Member Function Documentation

25.299.1.1 `EStateID gdcm::network::ULActionAR2::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

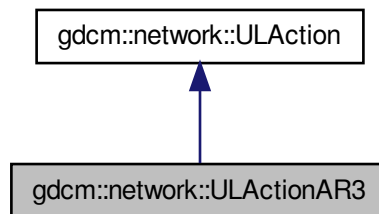
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

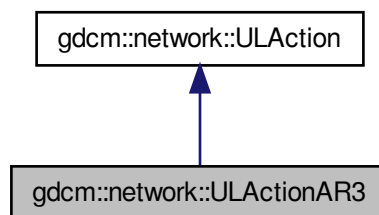
25.300 gdcmm::network::ULActionAR3 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR3:



Collaboration diagram for gdcmm::network::ULActionAR3:



Public Member Functions

- [EStateID](#) [PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.300.1 Member Function Documentation

25.300.1.1 **EStateID** `gdcmm::network::ULActionAR3::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcmm::network::ULAction](#).

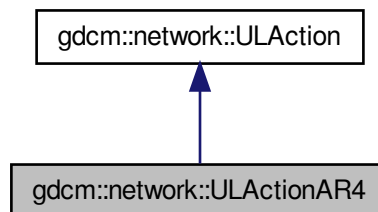
The documentation for this class was generated from the following file:

- [gdcmmULActionAR.h](#)

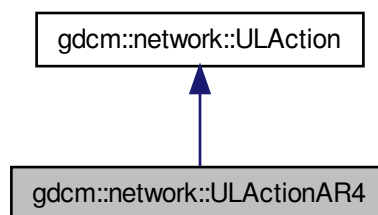
25.301 `gdcmm::network::ULActionAR4` Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for `gdcmm::network::ULActionAR4`:



Collaboration diagram for `gdcmm::network::ULActionAR4`:



Public Member Functions

- **EStateID** `PerformAction` (`Subject` *s, `ULEvent` &inEvent, `ULConnection` &inConnection, bool &outWaitingForEvent, `EEventID` &outRaisedEvent)

25.301.1 Member Function Documentation

25.301.1.1 **EStateID** gdcm::network::ULActionAR4::PerformAction (**Subject** * *s*, **ULError** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements [gdcm::network::ULAction](#).

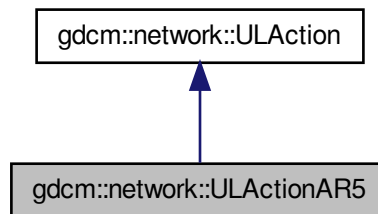
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

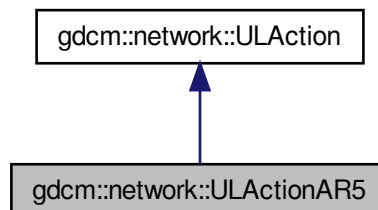
25.302 gdcm::network::ULActionAR5 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR5:



Collaboration diagram for gdcm::network::ULActionAR5:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.302.1 Member Function Documentation

25.302.1.1 [EStateID](#) `gdcm::network::ULActionAR5::PerformAction (Subject * s, ULError & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcm::network::ULAction](#).

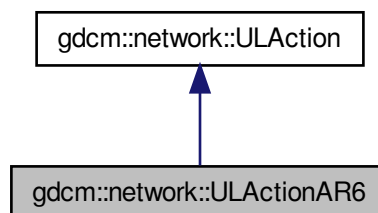
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

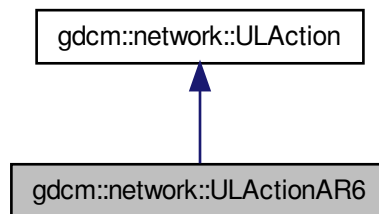
25.303 gdcm::network::ULActionAR6 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.303.1 Member Function Documentation

25.303.1.1 **EStateID** `gdcm::network::ULActionAR6::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent)` [virtual]

Implements [gdcm::network::ULAction](#).

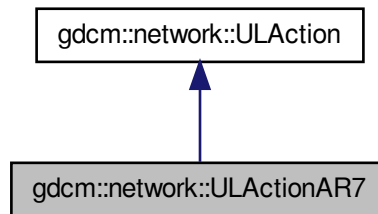
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

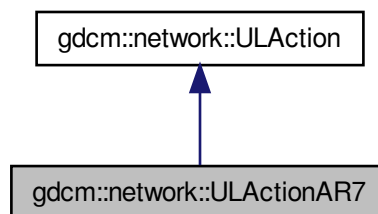
25.304 gdcm::network::ULActionAR7 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for `gdcm::network::ULActionAR7`:



Collaboration diagram for `gdcm::network::ULActionAR7`:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.304.1 Member Function Documentation

25.304.1.1 `EStateID gdcm::network::ULActionAR7::PerformAction (Subject * s, ULError & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

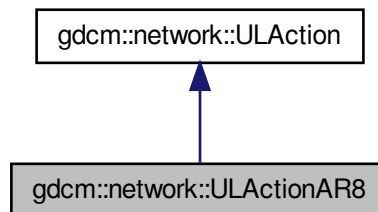
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

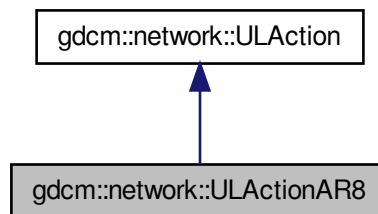
25.305 gdcm::network::ULActionAR8 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.305.1 Member Function Documentation

25.305.1.1 [EStateID](#) `gdcm::network::ULActionAR8::PerformAction` ([Subject](#) * s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent*) `[virtual]`

Implements [gdcm::network::ULAction](#).

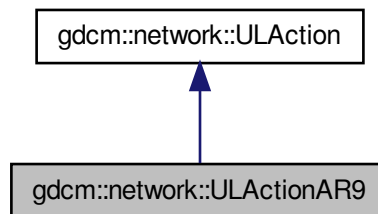
The documentation for this class was generated from the following file:

- [gdcmlActionAR.h](#)

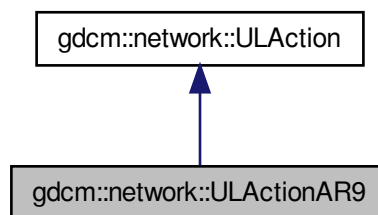
25.306 gdcmlnetwork::ULActionAR9 Class Reference

```
#include <gdcmlActionAR.h>
```

Inheritance diagram for gdcmlnetwork::ULActionAR9:



Collaboration diagram for gdcmlnetwork::ULActionAR9:



Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.306.1 Member Function Documentation

25.306.1.1 **EStateID** gdcm::network::ULActionAR9::PerformAction (**Subject** * *s*, **UEvent** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent*) [virtual]

Implements [gdcm::network::ULAction](#).

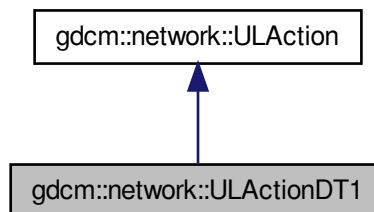
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

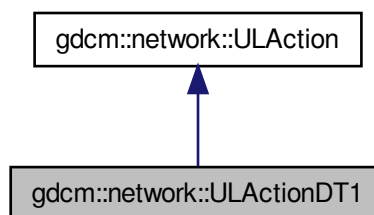
25.307 gdcm::network::ULActionDT1 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for gdcm::network::ULActionDT1:



Collaboration diagram for gdcm::network::ULActionDT1:



Public Member Functions

- **EStateID** PerformAction (**Subject** **s*, **UEvent** &*inEvent*, **ULConnection** &*inConnection*, **bool** &*outWaitingForEvent*, **EEventID** &*outRaisedEvent*)

25.307.1 Member Function Documentation

25.307.1.1 `EStateID gdcm::network::ULActionDT1::PerformAction (Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent) [virtual]`

Implements [gdcm::network::ULAction](#).

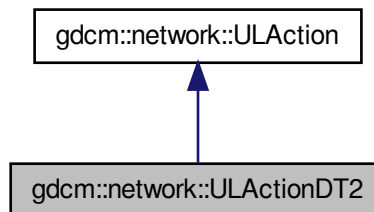
The documentation for this class was generated from the following file:

- [gdcmULActionDT.h](#)

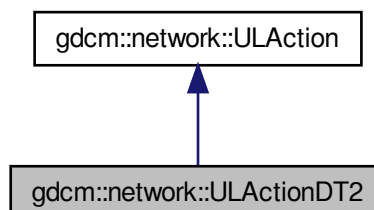
25.308 gdcm::network::ULActionDT2 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

25.308.1 Member Function Documentation

25.308.1.1 [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:

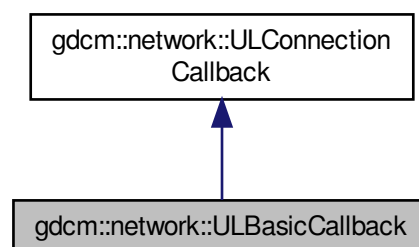
- [gdcmULActionDT.h](#)

25.309 gdcm::network::ULBasicCallback Class Reference

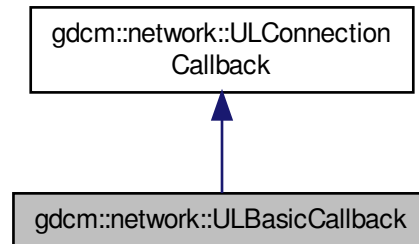
[ULBasicCallback](#) This is the most basic of callbacks for how the [ULConnectionManager](#) handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to 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

25.309.1 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](#).

25.309.2 Constructor & Destructor Documentation

25.309.2.1 `gdcm::network::ULBasicCallback::ULBasicCallback ()` `[inline]`

25.309.2.2 `virtual gdcm::network::ULBasicCallback::~~ULBasicCallback ()` `[inline], [virtual]`

25.309.3 Member Function Documentation

25.309.3.1 `std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetDataSets () const`

25.309.3.2 `std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetResponses () const`

25.309.3.3 `virtual void gdcm::network::ULBasicCallback::HandleDataSet (const DataSet & inDataSet)` `[virtual]`

Implements [gdcm::network::ULConnectionCallback](#).

25.309.3.4 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](#)

25.310 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](#) ()

25.310.1 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.

25.310.2 Constructor & Destructor Documentation

25.310.2.1 **gdcm::network::ULConnection::ULConnection (const **ULConnectionInfo** & *inUserInfo*)**

25.310.2.2 **virtual gdcm::network::ULConnection::~~ULConnection () [virtual]**

25.310.3 Member Function Documentation

25.310.3.1 **void gdcm::network::ULConnection::AddAcceptedPresentationContext (const **PresentationContextAC** & *inPC*)**

25.310.3.2 **PresentationContextRQ gdcm::network::ULConnection::FindContext (const **DataElement** & *de*) const**

25.310.3.3 **std::vector<**PresentationContextAC**> const& gdcm::network::ULConnection::GetAcceptedPresentationContexts () const**

25.310.3.4 **std::vector<**PresentationContextAC**>& gdcm::network::ULConnection::GetAcceptedPresentationContexts ()**

25.310.3.5 **const **ULConnectionInfo**& gdcm::network::ULConnection::GetConnectionInfo () const**

25.310.3.6 **uint32_t gdcm::network::ULConnection::GetMaxPDUSize () const**

25.310.3.7 **const **PresentationContextAC*** gdcm::network::ULConnection::GetPresentationContextACByID (uint8_t *id*) const**

25.310.3.8 **uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext (**PresentationContextRQ** const & *pc*) const**

return 0 upon error

25.310.3.9 **const **PresentationContextRQ*** gdcm::network::ULConnection::GetPresentationContextRQByID (uint8_t *id*) const**

25.310.3.10 **std::vector<**PresentationContextRQ**> const& gdcm::network::ULConnection::GetPresentationContexts () const**

25.310.3.11 **std::iostream* gdcm::network::ULConnection::GetProtocol ()**

25.310.3.12 **EStateID gdcm::network::ULConnection::GetState () const**

25.310.3.13 **ARTIMTimer&** gdcm::network::ULConnection::GetTimer ()

25.310.3.14 **bool** gdcm::network::ULConnection::InitializeConnection ()

used to establish scu connections

25.310.3.15 **bool** gdcm::network::ULConnection::InitializeIncomingConnection ()

used to establish scp connections

25.310.3.16 **void** gdcm::network::ULConnection::SetMaxPDUSize (**uint32_t** *inSize*)

25.310.3.17 **void** gdcm::network::ULConnection::SetPresentationContexts (**const** std::vector< **PresentationContextRQ** > & *inContexts*)

25.310.3.18 **void** gdcm::network::ULConnection::SetPresentationContexts (**const** std::vector< **PresentationContext** > & *inContexts*)

25.310.3.19 **void** gdcm::network::ULConnection::SetState (**const** **EStateID** & *inState*)

25.310.3.20 **void** gdcm::network::ULConnection::StopProtocol ()

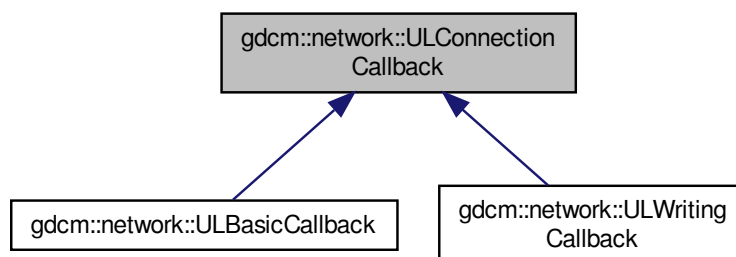
The documentation for this class was generated from the following file:

- [gdcmULConnection.h](#)

25.311 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](#) ()

Protected Member Functions

- void [DataSetHandled](#) ()

25.311.1 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 [mHandledDataSet](#) to true. otherwise, the cmove event loop handler will not know data was received, and proceed to end the loop prematurely.

25.311.2 Constructor & Destructor Documentation

25.311.2.1 `gdcmm::network::ULConnectionCallback::ULConnectionCallback ()` `[inline]`

25.311.2.2 `virtual gdcmm::network::ULConnectionCallback::~~ULConnectionCallback ()` `[inline]`, `[virtual]`

25.311.3 Member Function Documentation

25.311.3.1 `void gdcmm::network::ULConnectionCallback::DataSetHandled ()` `[inline]`, `[protected]`

25.311.3.2 `bool gdcmm::network::ULConnectionCallback::DataSetHandles ()` const `[inline]`

25.311.3.3 `virtual void gdcmm::network::ULConnectionCallback::HandleDataSet (const DataSet & inDataSet)` `[pure virtual]`

Implemented in [gdcmm::network::ULBasicCallback](#), and [gdcmm::network::ULWritingCallback](#).

25.311.3.4 `virtual void gdcmm::network::ULConnectionCallback::HandleResponse (const DataSet & inDataSet)` `[pure virtual]`

Implemented in [gdcmm::network::ULBasicCallback](#), and [gdcmm::network::ULWritingCallback](#).

25.311.3.5 `void gdcmm::network::ULConnectionCallback::ResetHandledDataSet ()` `[inline]`

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

- [gdcmmULConnectionCallback.h](#)

25.312 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](#) ([UserInfo](#) const &inUserInfo, const char *inCalledAETitle, const char *inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void [SetMaxPDULength](#) (unsigned long inMaxPDULength)

25.312.1 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.

25.312.2 Constructor & Destructor Documentation

25.312.2.1 `gdcm::network::ULConnectionInfo::ULConnectionInfo ()`

25.312.3 Member Function Documentation

25.312.3.1 `const char* gdcm::network::ULConnectionInfo::GetCalledAETitle () const`

25.312.3.2 `std::string gdcm::network::ULConnectionInfo::GetCalledComputerName () const`

25.312.3.3 `unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress () const`

25.312.3.4 `int gdcm::network::ULConnectionInfo::GetCalledIPPort () const`

25.312.3.5 `const char* gdcm::network::ULConnectionInfo::GetCallingAETitle () const`

25.312.3.6 `unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength () const`

25.312.3.7 `bool gdcm::network::ULConnectionInfo::Initialize (UserInfo const & inUserInfo, const char * inCalledAETitle, const char * inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)`

25.312.3.8 void `gdcm::network::ULConnectionInfo::SetMaxPDULength` (unsigned long *inMaxPDULength*)

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

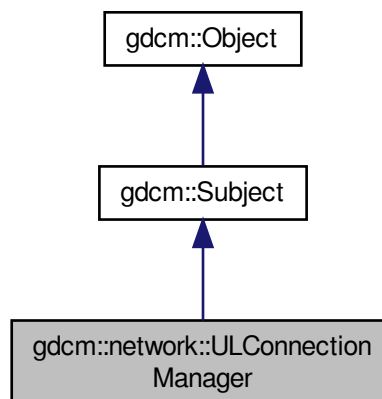
- [gdcmULConnectionInfo.h](#)

25.313 `gdcm::network::ULConnectionManager` Class Reference

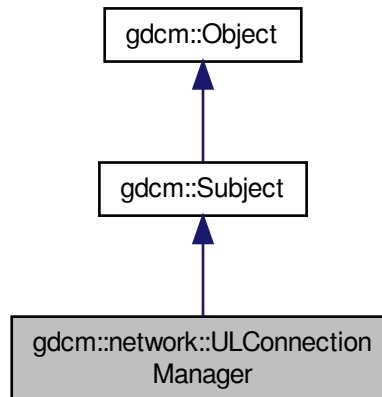
[ULConnectionManager](#) The [ULConnectionManager](#) performs actions on the [ULConnection](#) given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

```
#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

25.313.1 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.

25.313.2 Constructor & Destructor Documentation

25.313.2.1 `gdcmm::network::ULConnectionManager::ULConnectionManager ()`

25.313.2.2 `gdcmm::network::ULConnectionManager::~~ULConnectionManager ()`

25.313.3 Member Function Documentation

25.313.3.1 `bool gdcmm::network::ULConnectionManager::BreakConnection (const double & inTimeout)`

25.313.3.2 `void gdcmm::network::ULConnectionManager::BreakConnectionNow ()`

25.313.3.3 `bool gdcmm::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.

25.313.3.4 `bool gdcmm::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

25.313.3.5 `std::vector<PresentationDataValue> gdcmm::network::ULConnectionManager::SendEcho ()`

25.313.3.6 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendFind (const BaseRootQuery * inRootQuery)`

25.313.3.7 `void gdcmm::network::ULConnectionManager::SendFind (const BaseRootQuery * inRootQuery, ULConnectionCallback * inCallback)`

25.313.3.8 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendMove (const BaseRootQuery * inRootQuery)`

25.313.3.9 `bool gdcmm::network::ULConnectionManager::SendMove (const BaseRootQuery * inRootQuery, ULConnectionCallback * inCallback)`

return false upon error

25.313.3.10 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendStore (const File & file)`

25.313.3.11 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](#)

25.314 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)

25.314.1 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!

25.314.2 Constructor & Destructor Documentation

25.314.2.1 gdcm::network::ULEvent::ULEvent (const [EEventID](#) & inEventID, std::vector< [BasePDU](#) * > const & inBasePDU)
[inline]

25.314.2.2 gdcm::network::ULEvent::ULEvent (const [EEventID](#) & inEventID, [BasePDU](#) * inBasePDU) [inline]

25.314.2.3 gdcm::network::ULEvent::~~ULEvent () [inline]

25.314.3 Member Function Documentation

25.314.3.1 [EEventID](#) gdcm::network::ULEvent::GetEvent () const [inline]

25.314.3.2 std::vector<[BasePDU](#)*> const& gdcm::network::ULEvent::GetPDUs () const [inline]

25.314.3.3 void gdcm::network::ULEvent::SetEvent (const [EEventID](#) & inEvent) [inline]

25.314.3.4 void gdcmm::network::ULEvent::SetPDU (std::vector< BasePDU * > const & inPDU) [inline]

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

- [gdcmmULEvent.h](#)

25.315 gdcmm::network::ULTransitionTable Class Reference

[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.

```
#include <gdcmmULTransitionTable.h>
```

Public Member Functions

- [ULTransitionTable](#) ()
- void [HandleEvent](#) (Subject *s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [E-EventID](#) &outRaisedEvent) const
- void [PrintTable](#) () const

25.315.1 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.

25.315.2 Constructor & Destructor Documentation

25.315.2.1 gdcmm::network::ULTransitionTable::ULTransitionTable ()

25.315.3 Member Function Documentation

25.315.3.1 void gdcmm::network::ULTransitionTable::HandleEvent (Subject * s, [ULEvent](#) & inEvent, [ULConnection](#) & inConnection, bool & outWaitingForEvent, [EEventID](#) & outRaisedEvent) const

25.315.3.2 void gdcmm::network::ULTransitionTable::PrintTable () const

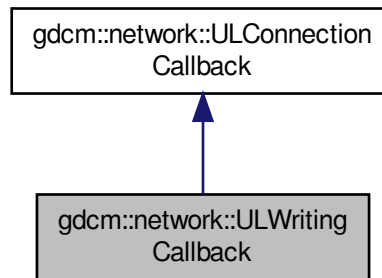
The documentation for this class was generated from the following file:

- [gdcmmULTransitionTable.h](#)

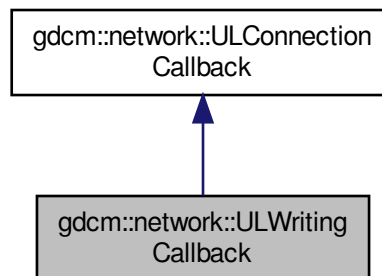
25.316 gdcmm::network::ULWritingCallback Class Reference

```
#include <gdcmmULWritingCallback.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

25.316.1 Constructor & Destructor Documentation

25.316.1.1 `gdcm::network::ULWritingCallback::ULWritingCallback ()` `[inline]`

25.316.1.2 `virtual gdcm::network::ULWritingCallback::~~ULWritingCallback ()` `[inline],[virtual]`

25.316.2 Member Function Documentation

25.316.2.1 `virtual void gdcm::network::ULWritingCallback::HandleDataSet (const DataSet & inDataSet)` `[virtual]`

Implements [gdcm::network::ULConnectionCallback](#).

25.316.2.2 `virtual void gdcm::network::ULWritingCallback::HandleResponse (const DataSet & inDataSet)` `[virtual]`

Implements [gdcm::network::ULConnectionCallback](#).

25.316.2.3 `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:

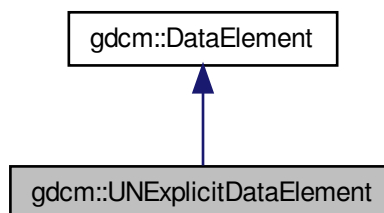
- [gdcmULWritingCallback.h](#)

25.317 gdcm::UNExplicitDataElement Class Reference

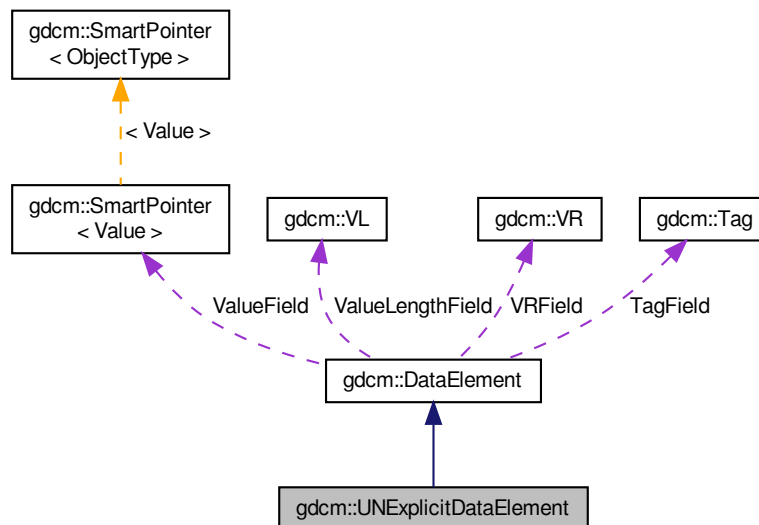
Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

```
#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)
- template<typename TSwap >
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

Additional Inherited Members

25.317.1 Detailed Description

Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

Note

bla

25.317.2 Member Function Documentation

25.317.2.1 VL gdcm::UNExplicitDataElement::GetLength () const

25.317.2.2 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::Read (std::istream & is)`

25.317.2.3 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadPreValue (std::istream & is)`

25.317.2.4 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadValue (std::istream & is)`

25.317.2.5 `template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadWithLength (std::istream & is, VL & length)`

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

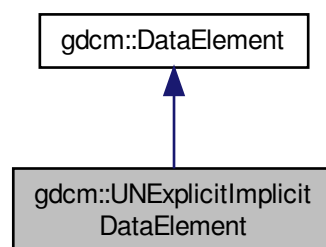
- [gdcmUNExplicitDataElement.h](#)

25.318 gdcm::UNExplicitImplicitDataElement Class Reference

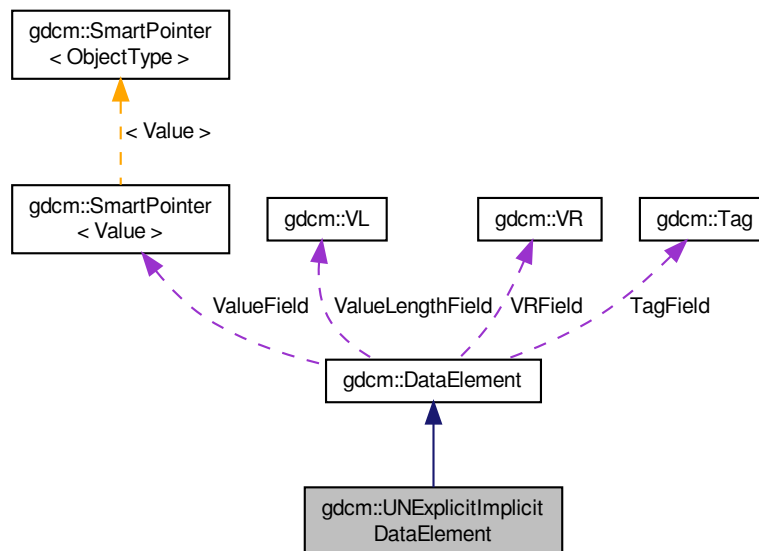
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#) This class gather two known bugs:

```
#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

25.318.1 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/TheralyGDCM120Bug.dcm

25.318.2 Member Function Documentation

25.318.2.1 VL gdcm::UNExplicitImplicitDataElement::GetLength () const

25.318.2.2 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::Read (std::istream & *is*)

25.318.2.3 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue (std::istream & *is*)

25.318.2.4 template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadValue (std::istream & *is*)

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

- [gdcmUNExplicitImplicitDataElement.h](#)

25.319 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)

25.319.1 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](#)

25.319.2 Member Function Documentation

25.319.2.1 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

25.319.2.2 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](#)

25.320 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)

25.320.1 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.

25.320.2 Member Enumeration Documentation

25.320.2.1 enum gdcm::Usage::UsageType

Enumerator

Mandatory

Conditional

UserOption

Invalid

25.320.3 Constructor & Destructor Documentation

25.320.3.1 gdcm::Usage::Usage (UsageType type = Invalid) [inline]

25.320.4 Member Function Documentation

25.320.4.1 static const char* gdcm::Usage::GetString (UsageType type) [static]

Referenced by gdcm::operator<<().

25.320.4.2 static UsageType gdcm::Usage::GetUsageType (const char * type) [static]

25.320.4.3 gdcm::Usage::operator UsageType () const [inline]

25.320.5 Friends And Related Function Documentation

25.320.5.1 std::ostream& operator<< (std::ostream & os, const Usage & vr) [friend]

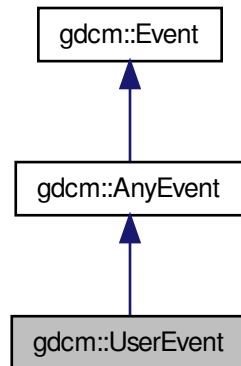
The documentation for this class was generated from the following file:

- [gdcmUsage.h](#)

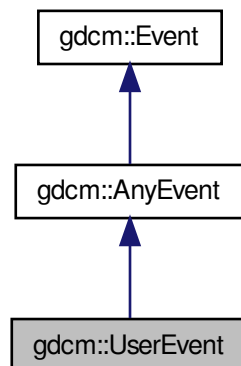
25.321 gdcm::UserEvent Class Reference

```
#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](#)

25.322 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

25.322.1 Detailed Description

[UserInformation Table 9-16](#) USER INFORMATION ITEM FIELDS.

TODO what is the goal of :

[Table 9-20](#) USER INFORMATION ITEM FIELDS

25.322.2 Constructor & Destructor Documentation

25.322.2.1 `gdcm::network::UserInformation::UserInformation ()`

25.322.2.2 `gdcm::network::UserInformation::~~UserInformation ()`

25.322.3 Member Function Documentation

25.322.3.1 `void gdcm::network::UserInformation::AddRoleSelectionSub (RoleSelectionSub const & r)`

25.322.3.2 `void gdcm::network::UserInformation::AddSOPClassExtendedNegociationSub (SOPClassExtendedNegociationSub const & s)`

25.322.3.3 `const MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub () const` `[inline]`

25.322.3.4 `MaximumLengthSub& gdcm::network::UserInformation::GetMaximumLengthSub ()` `[inline]`

25.322.3.5 `UserInformation& gdcm::network::UserInformation::operator= (const UserInformation &)`

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

25.322.3.7 `std::istream& gdcm::network::UserInformation::Read (std::istream & is)`

25.322.3.8 `size_t gdcm::network::UserInformation::Size () const`

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

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

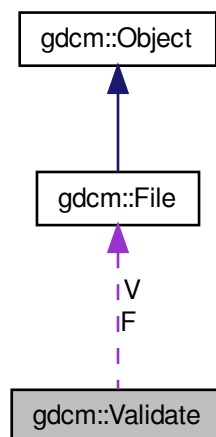
- [gdcmUserInformation.h](#)

25.323 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](#)

25.323.1 Detailed Description

[Validate](#) class.

25.323.2 Constructor & Destructor Documentation

25.323.2.1 `gdcm::Validate::Validate ()`

25.323.2.2 `gdcm::Validate::~~Validate ()`

25.323.3 Member Function Documentation

25.323.3.1 `const File& gdcm::Validate::GetValidatedFile ()` `[inline]`

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

25.323.3.3 `void gdcm::Validate::Validation ()`

25.323.4 Member Data Documentation

25.323.4.1 `const File* gdcm::Validate::F` `[protected]`

25.323.4.2 `File gdcm::Validate::V` `[protected]`

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

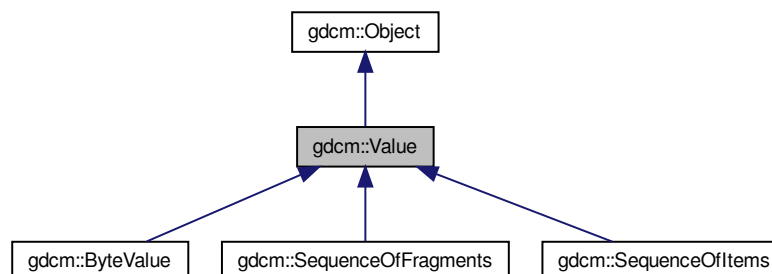
- [gdcmValidate.h](#)

25.324 gdcm::Value Class Reference

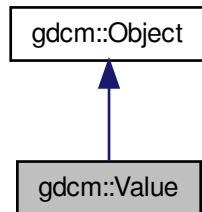
Class to represent the value of a Data [Element](#).

```
#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

Additional Inherited Members

25.324.1 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.

25.324.2 Constructor & Destructor Documentation

25.324.2.1 `gdcm::Value::Value ()` `[inline]`

25.324.2.2 `gdcm::Value::~~Value ()` `[inline]`

25.324.3 Member Function Documentation

25.324.3.1 `virtual void gdcm::Value::Clear ()` `[pure virtual]`

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfItems](#), and [gdcm::SequenceOfFragments](#).

25.324.3.2 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\(\)](#).

25.324.3.3 virtual bool gdcM::Value::operator== (const Value & val) const [pure virtual]

Implemented in [gdcM::SequenceOfFragments](#), [gdcM::SequenceOfItems](#), and [gdcM::ByteValue](#).

25.324.3.4 virtual void gdcM::Value::SetLength (VL /) [pure virtual]

Implemented in [gdcM::ByteValue](#), [gdcM::SequenceOfItems](#), and [gdcM::SequenceOfFragments](#).

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

- [gdcMValue.h](#)

25.325 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)
- static const std::ostream & [Write](#) (std::ostream &os, const [Value](#) &v)

25.325.1 Detailed Description

```
template<typename TDE, typename TSwap, typename TType = uint8_t> class gdcM::ValueIO< TDE, TSwap, TType >
```

Class to dispatch template calls.

25.325.2 Member Function Documentation

25.325.2.1 template<typename TDE , typename TSwap , typename TType = uint8_t> static std::istream& gdcM::ValueIO< TDE, TSwap, TType >::Read (std::istream & is, Value & v) [static]

25.325.2.2 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](#)

25.326 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)

25.326.1 Detailed Description

major/minor and build version

25.326.2 Constructor & Destructor Documentation

25.326.2.1 `gdcm::Version::Version ()` [inline]

25.326.2.2 `gdcm::Version::~~Version ()` [inline]

25.326.3 Member Function Documentation

25.326.3.1 `static int gdcm::Version::GetBuildVersion ()` [static]

25.326.3.2 `static int gdcm::Version::GetMajorVersion ()` [static]

25.326.3.3 `static int gdcm::Version::GetMinorVersion ()` [static]

25.326.3.4 `static const char* gdcm::Version::GetVersion ()` [static]

25.326.3.5 `void gdcm::Version::Print (std::ostream & os = std::cout) const`

Referenced by `gdcm::operator<<()`.

25.326.4 Friends And Related Function Documentation

25.326.4.1 `std::ostream& operator<< (std::ostream &_os, const Version &v)` [*friend*]

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

- [gdcVersion.h](#)

25.327 gdc::VL Class Reference

Value Length.

```
#include <gdcVL.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)

25.327.1 Detailed Description

Value Length.

Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

Examples:

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [NewSequence.cs](#), and [rle2img.cxx](#).

25.327.2 Member Typedef Documentation

25.327.2.1 `typedef uint32_t gdcm::VL::Type`

25.327.3 Constructor & Destructor Documentation

25.327.3.1 `gdcm::VL::VL (uint32_t v/ = 0) [inline]`

25.327.4 Member Function Documentation

25.327.4.1 `VL gdcm::VL::GetLength () const [inline]`

Referenced by `gdcm::FileMetaInformation::GetFullLength()`, `gdcm::Fragment::GetLength()`, and `gdcm::Item::Write()`.

25.327.4.2 `static uint16_t gdcm::VL::GetVL16Max () [inline], [static]`

25.327.4.3 `static uint32_t gdcm::VL::GetVL32Max () [inline], [static]`

25.327.4.4 `bool gdcm::VL::IsOdd () const [inline]`

Return whether or not the [VL](#) is odd or not.

Referenced by `gdcm::ByteValue::SetLength()`.

25.327.4.5 `bool gdcm::VL::IsUndefined () const [inline]`

Referenced by `gdcm::ByteValue::SetLength()`.

25.327.4.6 `gdcm::VL::operator uint32_t () const [inline]`

25.327.4.7 `VL& gdcm::VL::operator++ () [inline]`

25.327.4.8 `VL gdcm::VL::operator++ (int) [inline]`

25.327.4.9 `VL& gdcm::VL::operator+= (VL const & v/) [inline]`

`+=` operator

25.327.4.10 `template<typename TSwap > std::istream& gdcml::VL::Read (std::istream & is) [inline]`

25.327.4.11 `template<typename TSwap > std::istream& gdcml::VL::Read16 (std::istream & is) [inline]`

25.327.4.12 `void gdcml::VL::SetToUndefined () [inline]`

25.327.4.13 `template<typename TSwap > const std::ostream& gdcml::VL::Write (std::ostream & os) const [inline]`

Referenced by `gdcml::Fragment::Write()`, `gdcml::SequenceOfItems::Write()`, `gdcml::Item::Write()`, and `gdcml::SequenceOfFragments::Write()`.

25.327.4.14 `template<typename TSwap > const std::ostream& gdcml::VL::Write16 (std::ostream & os) const [inline]`

25.327.5 Friends And Related Function Documentation

25.327.5.1 `std::ostream& operator<< (std::ostream & os, const VL & vl) [friend]`

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

- [gdcmlVL.h](#)

25.328 gdcml::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 <gdcmlVM.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)

25.328.1 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

25.328.2 Member Enumeration Documentation

25.328.2.1 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

25.328.3 Constructor & Destructor Documentation

25.328.3.1 `gdcm::VM::VM (VMType type = VM0) [inline]`

25.328.4 Member Function Documentation

25.328.4.1 `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.

25.328.4.2 `static unsigned int gdcm::VM::GetIndex (VMType vm) [static], [protected]`

25.328.4.3 `unsigned int gdcm::VM::GetLength () const`

25.328.4.4 `static unsigned int gdcm::VM::GetNumberOfElementsFromArray (const char * array, unsigned int length) [static]`

25.328.4.5 `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<<()`.

25.328.4.6 `static VMType gdcM::VM::GetVMType (const char * vm) [static]`

25.328.4.7 `static VMType gdcM::VM::GetVMTypeFromLength (unsigned int length, unsigned int size) [static]`

25.328.4.8 `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

25.328.4.9 `gdcM::VM::operator VMType () const [inline]`

25.328.5 Friends And Related Function Documentation

25.328.5.1 `std::ostream& operator<< (std::ostream & os, const VM & vm) [friend]`

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

- [gdcMVM.h](#)

25.329 gdcM::VMToLength< T > Struct Template Reference

```
#include <gdcMVM.h>
```

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

- [gdcMVM.h](#)

25.330 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,
 - `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 | 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 | OF | OW | SL | SQ | SS | UL | UN | US,
 - `VR_VM1` = AS | LT | ST | UT | SQ | OF | 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)`

25.330.1 Detailed Description

VR class This is adapted from DICOM standard The biggest difference is the INVALID **VR** and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict.

Note

VALUE REPRESENTATION (**VR**) Specifies the data type and format of the Value(s) contained in the **Value** Field of a Data **Element**. VALUE REPRESENTATION FIELD: The field where the **Value** Representation of a Data **Element** is stored in the encoding of a Data **Element** structure with explicit **VR**.

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [NewSequence.cs](#).

25.330.2 Member Enumeration Documentation

25.330.2.1 `enum gdcm::VR::VRType`

Enumerator

INVALID
AE
AS
AT
CS
DA

DS
DT
FD
FL
IS
LO
LT
OB
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

25.330.3 Constructor & Destructor Documentation

25.330.3.1 `gdcmm::VR::VR (VRType vr = INVALID) [inline]`

25.330.4 Member Function Documentation

25.330.4.1 `static bool gdcmm::VR::CanDisplay (VRType vr) [static]`

25.330.4.2 `bool gdcmm::VR::Compatible (VR const & vr) const`

25.330.4.3 `int gdcm::VR::GetLength () const [inline]`

25.330.4.4 `static uint32_t gdcm::VR::GetLength (VRType vr) [inline], [static]`

25.330.4.5 `unsigned int gdcm::VR::GetSize () const [inline]`

References AE, US_SS, and VRTypeTemplateCase.

25.330.4.6 `unsigned int gdcm::VR::GetSizeof () const`

25.330.4.7 `static const char* gdcm::VR::GetVRString (VRType vr) [static]`

Referenced by `gdcm::operator<<()`.

25.330.4.8 `static const char* gdcm::VR::GetVRStringFromFile (VRType vr) [static]`

25.330.4.9 `static VRType gdcm::VR::GetVRType (const char * vr) [static]`

25.330.4.10 `static VRType gdcm::VR::GetVRTypeFromFile (const char * vr) [static]`

25.330.4.11 `static bool gdcm::VR::IsASCII (VRType vr) [static]`

25.330.4.12 `static bool gdcm::VR::IsASCII2 (VRType vr) [static]`

25.330.4.13 `static bool gdcm::VR::IsBinary (VRType vr) [static]`

25.330.4.14 `static bool gdcm::VR::IsBinary2 (VRType vr) [static]`

25.330.4.15 `bool gdcm::VR::IsDual () const`

25.330.4.16 `static bool gdcm::VR::IsSwap (const char * vr) [static]`

25.330.4.17 `static bool gdcm::VR::IsValid (const char * vr) [static]`

25.330.4.18 `static bool gdcm::VR::IsValid (const char * vr1, VRType vr2) [static]`

25.330.4.19 `bool gdcm::VR::IsVRFile () const`

Referenced by `gdcm::DataElement::SetVR()`.

25.330.4.20 `gdcm::VR::operator VRType () const [inline]`

25.330.4.21 `std::istream& gdcm::VR::Read (std::istream & is) [inline]`

References `gdcmDebugMacro`, `INVALID`, and `VR_END`.

25.330.4.22 `const std::ostream& gdcm::VR::Write (std::ostream & os) const [inline]`

References `gdcmAssertAlwaysMacro`, and `INVALID`.

25.330.5 Friends And Related Function Documentation

25.330.5.1 `std::ostream& operator<< (std::ostream & os, const VR & vr)` `[friend]`

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

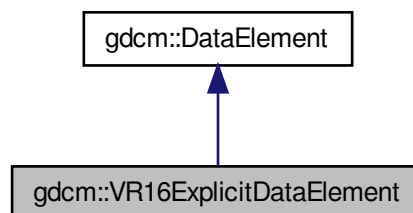
- [gdcmVR.h](#)

25.331 gdcm::VR16ExplicitDataElement Class Reference

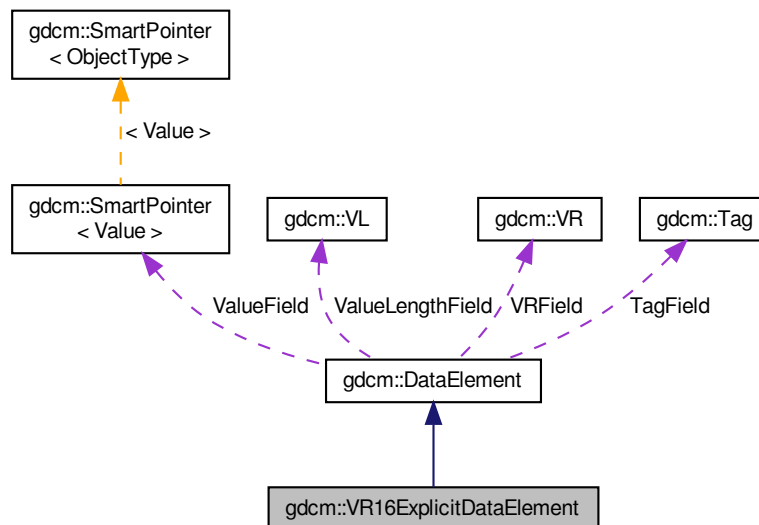
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#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)
- template<typename TSwap >
std::istream & [ReadWithLength](#) (std::istream &is, [VL](#) &length)

Additional Inherited Members

25.331.1 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

25.331.2 Member Function Documentation

25.331.2.1 VL `gdcm::VR16ExplicitDataElement::GetLength` () const

25.331.2.2 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::Read (std::istream & is)`

25.331.2.3 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadPreValue (std::istream & is)`

25.331.2.4 `template<typename TSwap > std::istream& gdcm::VR16ExplicitDataElement::ReadValue (std::istream & is)`

25.331.2.5 `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](#)

25.332 `gdcm::VRToEncoding< T >` Struct Template Reference

```
#include <gdcmVR.h>
```

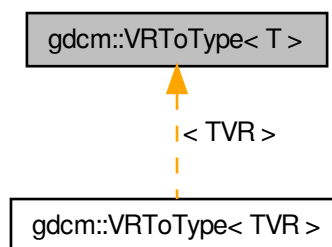
The documentation for this struct was generated from the following file:

- [gdcmVR.h](#)

25.333 `gdcm::VRToType< T >` Struct Template Reference

```
#include <gdcmVR.h>
```

Inheritance diagram for `gdcm::VRToType< T >`:



25.333.1 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](#)

25.334 `gdcm::VRVLSize< T >` Class Template Reference

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

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

- [gdcmAttribute.h](#)

25.335 `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)

25.335.1 Member Function Documentation

25.335.1.1 static uint16_t `gdcm::VRVLSize< 0 >::Read (std::istream &_is)` [inline], [static]

25.335.1.2 static void `gdcm::VRVLSize< 0 >::Write (std::ostream &os)` [inline], [static]

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

- [gdcmAttribute.h](#)

25.336 `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)

25.336.1 Member Function Documentation

25.336.1.1 static uint32_t `gdcm::VRVLSize< 1 >::Read (std::istream &_is)` [inline], [static]

25.336.1.2 static void `gdcm::VRVLSize< 1 >::Write (std::ostream &os)` [inline], [static]

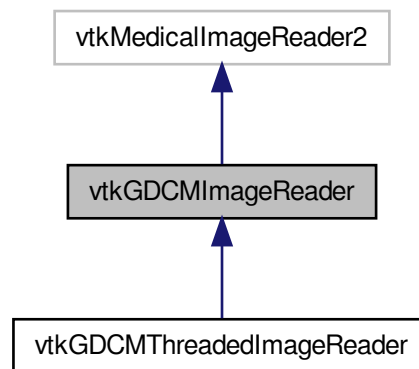
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

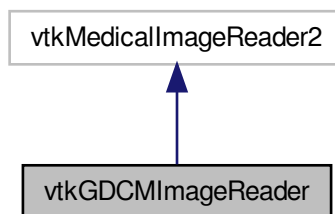
25.337 vtkGDCMImageReader Class Reference

```
#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)
- `vtkBooleanMacro` (ApplyYBRToRGB, int)
- `vtkGetMacro` (LoadOverlays, int)
- `vtkGetMacro` (LoadIconImage, int)
- `vtkGetMacro` (LossyFlag, int)
- `vtkGetMacro` (NumberOfOverlays, int)
- `vtkGetMacro` (NumberOfIconImages, int)
- `vtkGetMacro` (ApplyLookupTable, int)
- `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](#)

25.337.1 Detailed Description

Examples:

[AWTMedical3.java](#), [Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmmorthoplanes.cxx](#), [gdcmmreslice.cxx](#), [gdcmmtexture.cxx](#), [gdcmmvolume.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](#).

25.337.2 Constructor & Destructor Documentation

25.337.2.1 `vtkGDCMImageReader::vtkGDCMImageReader ()` [protected]

25.337.2.2 `vtkGDCMImageReader::~~vtkGDCMImageReader ()` [protected]

25.337.3 Member Function Documentation

25.337.3.1 `virtual int vtkGDCMImageReader::CanReadFile (const char * fname)` [virtual]

25.337.3.2 `void vtkGDCMImageReader::ExecuteData (vtkDataObject * out)` [protected]

25.337.3.3 `void vtkGDCMImageReader::ExecuteInformation ()` [protected]

- 25.337.3.4 `void vtkGDCMImageReader::FillMedicalImageInformation (const gdcm::ImageReader & reader)`
[protected]
- 25.337.3.5 `virtual const char* vtkGDCMImageReader::GetDescriptiveName ()` [inline],[virtual]
- 25.337.3.6 `virtual const char* vtkGDCMImageReader::GetFileExtensions ()` [inline],[virtual]
- 25.337.3.7 `vtkImageData* vtkGDCMImageReader::GetIconImage ()`
- 25.337.3.8 `vtkImageData* vtkGDCMImageReader::GetOverlay (int i)`
- 25.337.3.9 `int vtkGDCMImageReader::LoadSingleFile (const char * filename, char * pointer, unsigned long & outlen)`
[protected]
- 25.337.3.10 `static vtkGDCMImageReader* vtkGDCMImageReader::New ()` [static]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingle-BitTo8Bits.cxx](#), [gdcmorthoplanes.cxx](#), [gdcmreslice.cxx](#), [gdcmtexture.cxx](#), [gdcmvolume.cxx](#), [MagnifyFile.cxx](#), [offscreenimage.cxx](#), and [reslicesphere.cxx](#).

- 25.337.3.11 `virtual void vtkGDCMImageReader::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

Reimplemented in [vtkGDCMThreadedImageReader](#).

- 25.337.3.12 `int vtkGDCMImageReader::RequestDataCompat ()` [protected]
- 25.337.3.13 `int vtkGDCMImageReader::RequestInformationCompat ()` [protected]
- 25.337.3.14 `virtual void vtkGDCMImageReader::SetCurve (vtkPolyData * pd)` [virtual]
- 25.337.3.15 `virtual void vtkGDCMImageReader::SetFileNames (vtkStringArray *)` [virtual]

Examples:

[gdcmorthoplanes.cxx](#).

- 25.337.3.16 `void vtkGDCMImageReader::SetFilePattern (const char *)` [inline],[protected]
- 25.337.3.17 `void vtkGDCMImageReader::SetFilePrefix (const char *)` [inline],[protected]
- 25.337.3.18 `virtual void vtkGDCMImageReader::SetMedicalImageProperties (vtkMedicalImageProperties * pd)` [virtual]
- 25.337.3.19 `vtkGDCMImageReader::vtkBooleanMacro (LoadOverlays , int)`
- 25.337.3.20 `vtkGDCMImageReader::vtkBooleanMacro (LoadIconImage , int)`
- 25.337.3.21 `vtkGDCMImageReader::vtkBooleanMacro (LossyFlag , int)`

- 25.337.3.22 `vtkGDCMImageReader::vtkBooleanMacro (ApplyLookupTable , int)`
- 25.337.3.23 `vtkGDCMImageReader::vtkBooleanMacro (ApplyYBRToRGB , int)`
- 25.337.3.24 `vtkGDCMImageReader::vtkGetMacro (LoadOverlays , int)`
- 25.337.3.25 `vtkGDCMImageReader::vtkGetMacro (LoadIconImage , int)`
- 25.337.3.26 `vtkGDCMImageReader::vtkGetMacro (LossyFlag , int)`
- 25.337.3.27 `vtkGDCMImageReader::vtkGetMacro (NumberOfOverlays , int)`
- 25.337.3.28 `vtkGDCMImageReader::vtkGetMacro (NumberOfIconImages , int)`
- 25.337.3.29 `vtkGDCMImageReader::vtkGetMacro (ApplyLookupTable , int)`
- 25.337.3.30 `vtkGDCMImageReader::vtkGetMacro (ImageFormat , int)`
- 25.337.3.31 `vtkGDCMImageReader::vtkGetMacro (PlanarConfiguration , int)`
- 25.337.3.32 `vtkGDCMImageReader::vtkGetMacro (Shift , double)`
- 25.337.3.33 `vtkGDCMImageReader::vtkGetMacro (Scale , double)`
- 25.337.3.34 `vtkGDCMImageReader::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)`
- 25.337.3.35 `vtkGDCMImageReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)`
- 25.337.3.36 `vtkGDCMImageReader::vtkGetObjectMacro (FileNames , vtkStringArray)`
- 25.337.3.37 `vtkGDCMImageReader::vtkGetObjectMacro (Curve , vtkPolyData)`
- 25.337.3.38 `vtkGDCMImageReader::vtkGetStringMacro (FilePrefix) [protected]`
- 25.337.3.39 `vtkGDCMImageReader::vtkGetStringMacro (FilePattern) [protected]`
- 25.337.3.40 `vtkGDCMImageReader::vtkGetVector3Macro (ImagePositionPatient , double)`
- 25.337.3.41 `vtkGDCMImageReader::vtkGetVector6Macro (ImageOrientationPatient , double)`
- 25.337.3.42 `vtkGDCMImageReader::vtkSetMacro (LoadOverlays , int)`
- 25.337.3.43 `vtkGDCMImageReader::vtkSetMacro (LoadIconImage , int)`
- 25.337.3.44 `vtkGDCMImageReader::vtkSetMacro (LossyFlag , int)`
- 25.337.3.45 `vtkGDCMImageReader::vtkSetMacro (ApplyLookupTable , int)`
- 25.337.3.46 `vtkGDCMImageReader::vtkSetVector6Macro (ImageOrientationPatient , double) [protected]`
- 25.337.3.47 `vtkGDCMImageReader::vtkTypeRevisionMacro (vtkGDCMImageReader , vtkMedicalImageReader2)`

25.337.4 Member Data Documentation

- 25.337.4.1 `int vtkGDCMImageReader::ApplyInverseVideo` [protected]
- 25.337.4.2 `int vtkGDCMImageReader::ApplyLookupTable` [protected]
- 25.337.4.3 `int vtkGDCMImageReader::ApplyPlanarConfiguration` [protected]
- 25.337.4.4 `int vtkGDCMImageReader::ApplyShiftScale` [protected]
- 25.337.4.5 `int vtkGDCMImageReader::ApplyYBRToRGB` [protected]
- 25.337.4.6 `vtkPolyData* vtkGDCMImageReader::Curve` [protected]
- 25.337.4.7 `vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines` [protected]
- 25.337.4.8 `vtkStringArray* vtkGDCMImageReader::FileNames` [protected]
- 25.337.4.9 `int vtkGDCMImageReader::ForceRescale` [protected]
- 25.337.4.10 `int vtkGDCMImageReader::IconDataScalarType` [protected]
- 25.337.4.11 `int vtkGDCMImageReader::IconImageDataExtent[6]` [protected]
- 25.337.4.12 `int vtkGDCMImageReader::IconNumberOfScalarComponents` [protected]
- 25.337.4.13 `int vtkGDCMImageReader::ImageFormat` [protected]
- 25.337.4.14 `double vtkGDCMImageReader::ImageOrientationPatient[6]` [protected]
- 25.337.4.15 `double vtkGDCMImageReader::ImagePositionPatient[3]` [protected]
- 25.337.4.16 `int vtkGDCMImageReader::LoadIconImage` [protected]
- 25.337.4.17 `int vtkGDCMImageReader::LoadOverlays` [protected]
- 25.337.4.18 `int vtkGDCMImageReader::LossyFlag` [protected]
- 25.337.4.19 `vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties` [protected]
- 25.337.4.20 `int vtkGDCMImageReader::NumberOfIconImages` [protected]
- 25.337.4.21 `int vtkGDCMImageReader::NumberOfOverlays` [protected]
- 25.337.4.22 `int vtkGDCMImageReader::PlanarConfiguration` [protected]
- 25.337.4.23 `double vtkGDCMImageReader::Scale` [protected]
- 25.337.4.24 `double vtkGDCMImageReader::Shift` [protected]

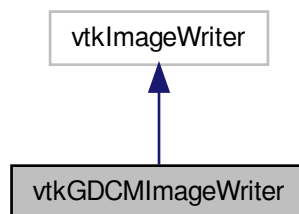
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader.h](#)

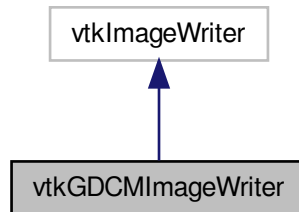
25.338 vtkGDCMImageWriter Class Reference

```
#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)

25.338.1 Detailed Description

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingle-BitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), and [RefCounting.cs](#).

25.338.2 Member Enumeration Documentation

25.338.2.1 enum vtkGDCMImageWriter::CompressionTypes

Enumerator

NO_COMPRESSION
JPEG_COMPRESSION
JPEG2000_COMPRESSION
JPEGLS_COMPRESSION
RLE_COMPRESSION

25.338.3 Constructor & Destructor Documentation

25.338.3.1 `vtkGDCMImageWriter::vtkGDCMImageWriter ()` [protected]

25.338.3.2 `vtkGDCMImageWriter::~~vtkGDCMImageWriter ()` [protected]

25.338.4 Member Function Documentation

25.338.4.1 `virtual const char* vtkGDCMImageWriter::GetDescriptiveName ()` [inline],[virtual]

25.338.4.2 `virtual const char* vtkGDCMImageWriter::GetFileExtensions ()` [inline],[virtual]

25.338.4.3 `virtual char* vtkGDCMImageWriter::GetFileName ()` [protected],[virtual]

25.338.4.4 `static vtkGDCMImageWriter* vtkGDCMImageWriter::New ()` [static]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), and [MagnifyFile.cxx](#).

25.338.4.5 `virtual void vtkGDCMImageWriter::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

25.338.4.6 `virtual void vtkGDCMImageWriter::SetDirectionCosines (vtkMatrix4x4 * matrix)` [virtual]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), and [MagnifyFile.cxx](#).

25.338.4.7 `virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient (const double dircos[6])` [virtual]

25.338.4.8 `virtual void vtkGDCMImageWriter::SetFileNames (vtkStringArray *)` [virtual]

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

25.338.4.9 virtual void vtkGDCMImageWriter::SetMedicalImageProperties (vtkMedicalImageProperties *) [virtual]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), and [MagnifyFile.cxx](#).

25.338.4.10 vtkGDCMImageWriter::vtkBooleanMacro (LossyFlag , int)

25.338.4.11 vtkGDCMImageWriter::vtkBooleanMacro (FileLowerLeft , int)

25.338.4.12 vtkGDCMImageWriter::vtkGetMacro (LossyFlag , int)

25.338.4.13 vtkGDCMImageWriter::vtkGetMacro (Shift , double)

25.338.4.14 vtkGDCMImageWriter::vtkGetMacro (Scale , double)

25.338.4.15 vtkGDCMImageWriter::vtkGetMacro (ImageFormat , int)

25.338.4.16 vtkGDCMImageWriter::vtkGetMacro (FileLowerLeft , int)

25.338.4.17 vtkGDCMImageWriter::vtkGetMacro (PlanarConfiguration , int)

25.338.4.18 vtkGDCMImageWriter::vtkGetMacro (CompressionType , int)

25.338.4.19 vtkGDCMImageWriter::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)

25.338.4.20 vtkGDCMImageWriter::vtkGetObjectMacro (FileNames , vtkStringArray)

25.338.4.21 vtkGDCMImageWriter::vtkGetObjectMacro (DirectionCosines , vtkMatrix4x4)

25.338.4.22 vtkGDCMImageWriter::vtkGetStringMacro (StudyUID)

25.338.4.23 vtkGDCMImageWriter::vtkGetStringMacro (SeriesUID)

25.338.4.24 vtkGDCMImageWriter::vtkSetMacro (LossyFlag , int)

25.338.4.25 vtkGDCMImageWriter::vtkSetMacro (Shift , double)

25.338.4.26 vtkGDCMImageWriter::vtkSetMacro (Scale , double)

25.338.4.27 vtkGDCMImageWriter::vtkSetMacro (ImageFormat , int)

25.338.4.28 vtkGDCMImageWriter::vtkSetMacro (FileLowerLeft , int)

25.338.4.29 vtkGDCMImageWriter::vtkSetMacro (PlanarConfiguration , int)

25.338.4.30 vtkGDCMImageWriter::vtkSetMacro (CompressionType , int)

25.338.4.31 vtkGDCMImageWriter::vtkSetStringMacro (StudyUID)

25.338.4.32 `vtkGDCMImageWriter::vtkSetStringMacro (SeriesUID)`

25.338.4.33 `vtkGDCMImageWriter::vtkTypeRevisionMacro (vtkGDCMImageWriter , vtkImageWriter)`

25.338.4.34 `virtual void vtkGDCMImageWriter::Write ()` [virtual]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingle-BitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), and [MagnifyFile.cxx](#).

25.338.4.35 `int vtkGDCMImageWriter::WriteGDCMData (vtkImageData * data, int timeStep)` [protected]

25.338.4.36 `void vtkGDCMImageWriter::WriteSlice (vtkImageData * data)` [protected]

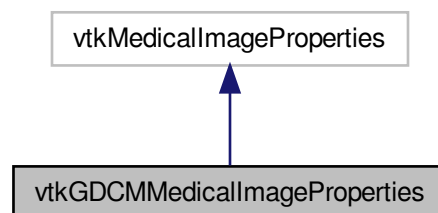
The documentation for this class was generated from the following file:

- [vtkGDCMImageWriter.h](#)

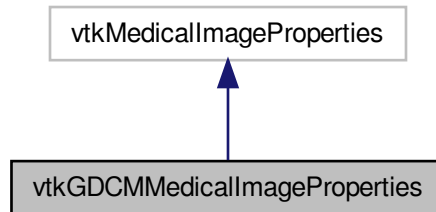
25.339 vtkGDCMMedicalImageProperties Class Reference

```
#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](#) ()
- [gdcmm::File](#) const & [GetFile](#) (unsigned int t)
- void [PushBackFile](#) ([gdcmm::File](#) const &f)

Friends

- class [vtkGDCMImageReader](#)
- class [vtkGDCMImageWriter](#)

25.339.1 Constructor & Destructor Documentation

25.339.1.1 `vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ()` [protected]

25.339.1.2 `vtkGDCMMedicalImageProperties::~~vtkGDCMMedicalImageProperties ()` [protected]

25.339.2 Member Function Documentation

- 25.339.2.1 `virtual void vtkGDCMMedicalImageProperties::Clear ()` `[virtual]`
- 25.339.2.2 `gdcmm::File const& vtkGDCMMedicalImageProperties::GetFile (unsigned int f)` `[protected]`
- 25.339.2.3 `static vtkGDCMMedicalImageProperties* vtkGDCMMedicalImageProperties::New ()` `[static]`
- 25.339.2.4 `void vtkGDCMMedicalImageProperties::PrintSelf (ostream & os, vtkIndent indent)`
- 25.339.2.5 `void vtkGDCMMedicalImageProperties::PushBackFile (gdcmm::File const & f)` `[protected]`
- 25.339.2.6 `vtkGDCMMedicalImageProperties::vtkTypeRevisionMacro (vtkGDCMMedicalImageProperties ,
vtkMedicalImageProperties)`

25.339.3 Friends And Related Function Documentation

- 25.339.3.1 `friend class vtkGDCMImageReader` `[friend]`
- 25.339.3.2 `friend class vtkGDCMImageWriter` `[friend]`

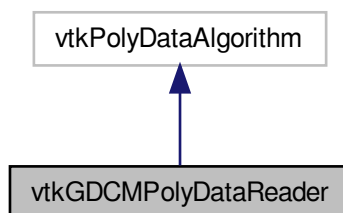
The documentation for this class was generated from the following file:

- [vtkGDCMMedicalImageProperties.h](#)

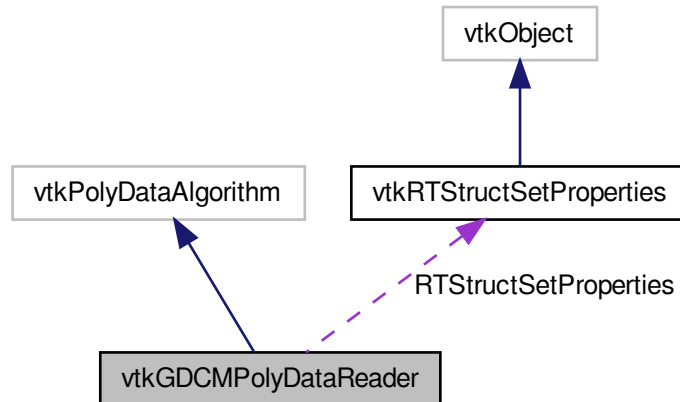
25.340 vtkGDCMPolyDataReader Class Reference

```
#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 `gdcmm::Reader` &reader)
- int `RequestData` (`vtkInformation` *, `vtkInformationVector` **, `vtkInformationVector` *)
- int `RequestData_HemodynamicWaveformStorage` (`gdcmm::Reader` const &reader, `vtkInformationVector` *outputVector)
- int `RequestData_RTStructureSetStorage` (`gdcmm::Reader` const &reader, `vtkInformationVector` *outputVector)
- int `RequestInformation` (`vtkInformation` *vtkNotUsed(request), `vtkInformationVector` **vtkNotUsed(inputVector), `vtkInformationVector` *outputVector)
- int `RequestInformation_HemodynamicWaveformStorage` (`gdcmm::Reader` const &reader)
- int `RequestInformation_RTStructureSetStorage` (`gdcmm::Reader` const &reader)

Protected Attributes

- char * [FileName](#)
- vtkMedicalImageProperties * [MedicalImageProperties](#)
- vtkRTStructSetProperties * [RTStructSetProperties](#)

25.340.1 Detailed Description

Examples:

[gdcmscene.cxx](#), [GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

25.340.2 Constructor & Destructor Documentation

25.340.2.1 `vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ()` [protected]

25.340.2.2 `vtkGDCMPolyDataReader::~~vtkGDCMPolyDataReader ()` [protected]

25.340.3 Member Function Documentation

25.340.3.1 `void vtkGDCMPolyDataReader::FillMedicalImageInformation (const gdcm::Reader & reader)` [protected]

25.340.3.2 `static vtkGDCMPolyDataReader* vtkGDCMPolyDataReader::New ()` [static]

Examples:

[gdcmscene.cxx](#), [GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

25.340.3.3 `virtual void vtkGDCMPolyDataReader::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

25.340.3.4 `int vtkGDCMPolyDataReader::RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *)` [protected]

25.340.3.5 `int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage (gdcm::Reader const & reader, vtkInformationVector * outputVector)` [protected]

25.340.3.6 `int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage (gdcm::Reader const & reader, vtkInformationVector * outputVector)` [protected]

25.340.3.7 `int vtkGDCMPolyDataReader::RequestInformation (vtkInformation * vtkNotUsed(request), vtkInformationVector **, vtkNotUsed(inputVector), vtkInformationVector * outputVector)` [protected]

25.340.3.8 `int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage (gdcm::Reader const & reader)` [protected]

25.340.3.9 `int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage (gdcm::Reader const & reader)` [protected]

25.340.3.10 `vtkGDCMPolyDataReader::vtkGetObjectMacro (MedicalImageProperties , vtkMedicalImageProperties)`

25.340.3.11 `vtkGDCMPolyDataReader::vtkGetObjectMacro (RTStructSetProperties , vtkRTStructSetProperties)`

25.340.3.12 `vtkGDCMPolyDataReader::vtkGetStringMacro (FileName)`

25.340.3.13 `vtkGDCMPolyDataReader::vtkSetStringMacro (FileName)`

25.340.3.14 `vtkGDCMPolyDataReader::vtkTypeRevisionMacro (vtkGDCMPolyDataReader , vtkPolyDataAlgorithm)`

25.340.4 Member Data Documentation

25.340.4.1 `char* vtkGDCMPolyDataReader::FileName` [protected]

25.340.4.2 `vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties` [protected]

25.340.4.3 `vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties` [protected]

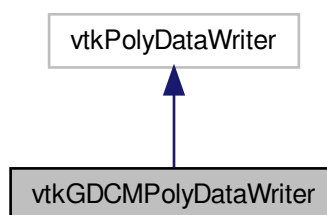
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataReader.h](#)

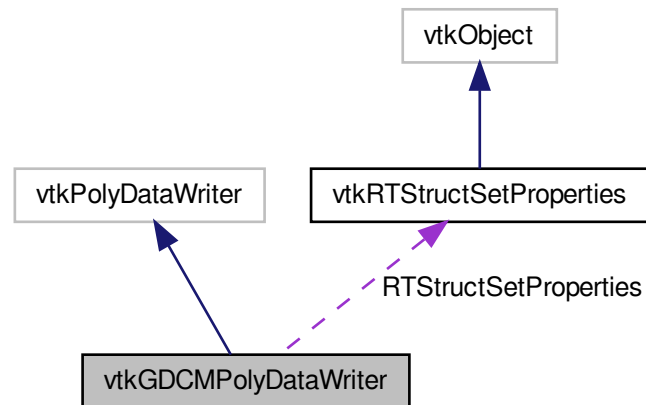
25.341 vtkGDCMPolyDataWriter Class Reference

```
#include <vtkGDCMPolyDataWriter.h>
```

Inheritance diagram for `vtkGDCMPolyDataWriter`:



Collaboration diagram for vtkGDCMPolyDataWriter:



Public Member Functions

- void [InitializeRTStructSet](#) (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray *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](#)

25.341.1 Detailed Description

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

25.341.2 Constructor & Destructor Documentation

25.341.2.1 `vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ()` [protected]

25.341.2.2 `vtkGDCMPolyDataWriter::~~vtkGDCMPolyDataWriter ()` [protected]

25.341.3 Member Function Documentation

25.341.3.1 `void vtkGDCMPolyDataWriter::InitializeRTStructSet (vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray * inROINames, vtkStringArray * inROIAlgorithmName, vtkStringArray * inROIType)`

Examples:

[GenerateRTSTRUCT.cxx](#).

25.341.3.2 `static vtkGDCMPolyDataWriter* vtkGDCMPolyDataWriter::New ()` [static]

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

25.341.3.3 `virtual void vtkGDCMPolyDataWriter::PrintSelf (ostream & os, vtkIndent indent)` [virtual]

25.341.3.4 `virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties (vtkMedicalImageProperties * pd)` [virtual]

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

25.341.3.5 `void vtkGDCMPolyDataWriter::SetNumberOfInputPorts (int n)`

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

25.341.3.6 `virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties (vtkRTStructSetProperties * pd)` [virtual]

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

25.341.3.7 `vtkGDCMPolyDataWriter::vtkTypeRevisionMacro (vtkGDCMPolyDataWriter , vtkPolyDataWriter)`

25.341.3.8 `void vtkGDCMPolyDataWriter::WriteData ()` [protected]

25.341.3.9 `void vtkGDCMPolyDataWriter::WriteRTSTRUCTData (gdcm::File & file, int num)` [protected]

25.341.3.10 `void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo (gdcm::File & file)` [protected]

25.341.4 Member Data Documentation

25.341.4.1 `vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties` [protected]

25.341.4.2 `vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties` [protected]

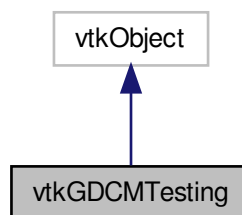
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataWriter.h](#)

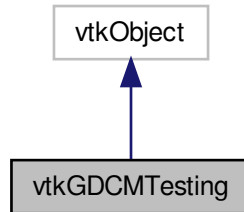
25.342 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Inheritance diagram for vtkGDCMTesting:



Collaboration diagram for vtkGDCMTesting:



Public Types

- typedef const char *const (* [MD5MetalmagesType](#))[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 * [GetMD5MetalImage](#) (unsigned int file)
- static const char * [GetMHDMD5FromFile](#) (const char *filepath)
- static unsigned int [GetNumberOfMD5MetalImages](#) ()
- static const char * [GetRAWMD5FromFile](#) (const char *filepath)
- static const char * [GetVTKDataRoot](#) ()
- static [vtkGDCMTesting](#) * [New](#) ()

Protected Member Functions

- [vtkGDCMTesting](#) ()
- [~vtkGDCMTesting](#) ()

25.342.1 Detailed Description

Examples:

[ReadSeriesIntoVTK.java](#), and [RefCounting.cs](#).

25.342.2 Member Typedef Documentation

25.342.2.1 `typedef const char* const(* vtkGDCMTesting::MD5MetalmagesType)[3]`

25.342.3 Constructor & Destructor Documentation

25.342.3.1 `vtkGDCMTesting::vtkGDCMTesting ()` [protected]

25.342.3.2 `vtkGDCMTesting::~~vtkGDCMTesting ()` [protected]

25.342.4 Member Function Documentation

25.342.4.1 `static const char* vtkGDCMTesting::GetGDCMDataRoot ()` [static]

25.342.4.2 `static const char* const* vtkGDCMTesting::GetMD5Metalmage (unsigned int file)` [static]

25.342.4.3 `static const char* vtkGDCMTesting::GetMHDMD5FromFile (const char * filepath)` [static]

25.342.4.4 `static unsigned int vtkGDCMTesting::GetNumberOfMD5Metalmages ()` [static]

25.342.4.5 `static const char* vtkGDCMTesting::GetRAWMD5FromFile (const char * filepath)` [static]

25.342.4.6 `static const char* vtkGDCMTesting::GetVTKDataRoot ()` [static]

25.342.4.7 `static vtkGDCMTesting* vtkGDCMTesting::New ()` [static]

25.342.4.8 `void vtkGDCMTesting::PrintSelf (ostream & os, vtkIndent indent)`

25.342.4.9 `vtkGDCMTesting::vtkTypeRevisionMacro (vtkGDCMTesting , vtkObject)`

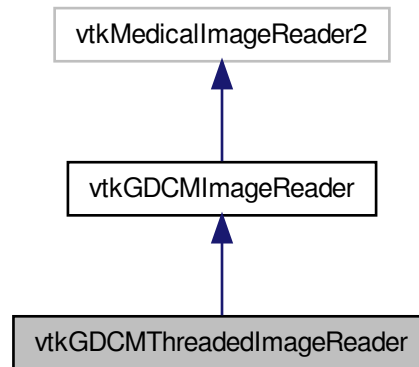
The documentation for this class was generated from the following file:

- [vtkGDCMTesting.h](#)

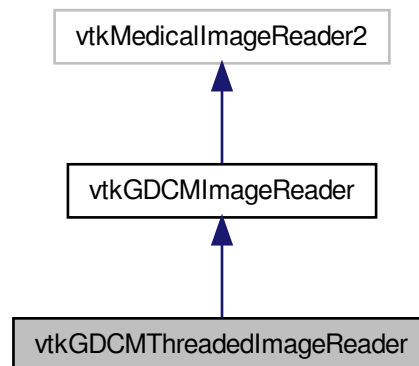
25.343 vtkGDCMThreadedImageReader Class Reference

```
#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

25.343.1 Constructor & Destructor Documentation

25.343.1.1 [vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader](#) () [protected]

25.343.1.2 [vtkGDCMThreadedImageReader::~~vtkGDCMThreadedImageReader](#) () [protected]

25.343.2 Member Function Documentation

25.343.2.1 void [vtkGDCMThreadedImageReader::ExecuteData](#) (vtkDataObject * *out*) [protected]

25.343.2.2 void [vtkGDCMThreadedImageReader::ExecuteInformation](#) () [protected]

25.343.2.3 static [vtkGDCMThreadedImageReader*](#) [vtkGDCMThreadedImageReader::New](#) () [static]

25.343.2.4 virtual void [vtkGDCMThreadedImageReader::PrintSelf](#) (ostream & *os*, vtkIndent *indent*) [virtual]

Reimplemented from [vtkGDCMImageReader](#).

25.343.2.5 void [vtkGDCMThreadedImageReader::ReadFiles](#) (unsigned int *nfiles*, const char * *filenames*[]) [protected]

25.343.2.6 void [vtkGDCMThreadedImageReader::RequestDataCompat](#) () [protected]

25.343.2.7 [vtkGDCMThreadedImageReader::vtkBooleanMacro](#) (UseShiftScale , int)

25.343.2.8 [vtkGDCMThreadedImageReader::vtkGetMacro](#) (UseShiftScale , int)

25.343.2.9 [vtkGDCMThreadedImageReader::vtkSetMacro](#) (Shift , double)

25.343.2.10 [vtkGDCMThreadedImageReader::vtkSetMacro](#) (Scale , double)

25.343.2.11 [vtkGDCMThreadedImageReader::vtkSetMacro](#) (UseShiftScale , int)

25.343.2.12 [vtkGDCMThreadedImageReader::vtkTypeRevisionMacro](#) ([vtkGDCMThreadedImageReader](#) , [vtkGDCMImageReader](#))

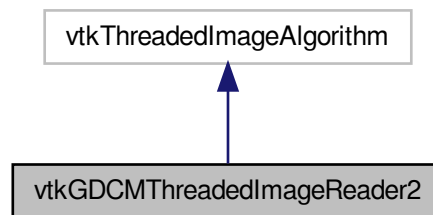
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader.h](#)

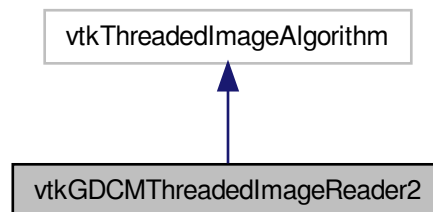
25.344 vtkGDCMThreadedImageReader2 Class Reference

```
#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)

25.344.1 Constructor & Destructor Documentation

25.344.1.1 [vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2](#) () [protected]

25.344.1.2 [vtkGDCMThreadedImageReader2::~~vtkGDCMThreadedImageReader2](#) () [protected]

25.344.2 Member Function Documentation

25.344.2.1 virtual const char* [vtkGDCMThreadedImageReader2::GetFileName](#) (int *i* = 0) [virtual]

25.344.2.2 static [vtkGDCMThreadedImageReader2](#)* [vtkGDCMThreadedImageReader2::New](#) () [static]

- 25.344.2.3 virtual void vtkGDCMThreadedImageReader2::PrintSelf (ostream & *os*, vtkIndent *indent*) [virtual]
- 25.344.2.4 int vtkGDCMThreadedImageReader2::RequestInformation (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*) [protected]
- 25.344.2.5 virtual void vtkGDCMThreadedImageReader2::SetFileName (const char * *filename*) [virtual]
- 25.344.2.6 virtual void vtkGDCMThreadedImageReader2::SetFileNames (vtkStringArray *) [virtual]
- 25.344.2.7 int vtkGDCMThreadedImageReader2::SplitExtent (int *splitExt[6]*, int *startExt[6]*, int *num*, int *total*)
- 25.344.2.8 void vtkGDCMThreadedImageReader2::ThreadedRequestData (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*, vtkImageData *** *inData*, vtkImageData ** *outData*, int *outExt[6]*, int *id*) [protected]
- 25.344.2.9 vtkGDCMThreadedImageReader2::vtkBooleanMacro (FileLowerLeft , int)
- 25.344.2.10 vtkGDCMThreadedImageReader2::vtkBooleanMacro (LoadOverlays , int)
- 25.344.2.11 vtkGDCMThreadedImageReader2::vtkBooleanMacro (UseShiftScale , int)
- 25.344.2.12 vtkGDCMThreadedImageReader2::vtkGetMacro (FileLowerLeft , int)
- 25.344.2.13 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfOverlays , int)
- 25.344.2.14 vtkGDCMThreadedImageReader2::vtkGetMacro (DataScalarType , int)
- 25.344.2.15 vtkGDCMThreadedImageReader2::vtkGetMacro (NumberOfScalarComponents , int)
- 25.344.2.16 vtkGDCMThreadedImageReader2::vtkGetMacro (LoadOverlays , int)
- 25.344.2.17 vtkGDCMThreadedImageReader2::vtkGetMacro (Shift , double)
- 25.344.2.18 vtkGDCMThreadedImageReader2::vtkGetMacro (Scale , double)
- 25.344.2.19 vtkGDCMThreadedImageReader2::vtkGetMacro (UseShiftScale , int)
- 25.344.2.20 vtkGDCMThreadedImageReader2::vtkGetObjectMacro (FileNames , vtkStringArray)
- 25.344.2.21 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataOrigin , double)
- 25.344.2.22 vtkGDCMThreadedImageReader2::vtkGetVector3Macro (DataSpacing , double)
- 25.344.2.23 vtkGDCMThreadedImageReader2::vtkGetVector6Macro (DataExtent , int)
- 25.344.2.24 vtkGDCMThreadedImageReader2::vtkSetMacro (FileLowerLeft , int)
- 25.344.2.25 vtkGDCMThreadedImageReader2::vtkSetMacro (DataScalarType , int)
- 25.344.2.26 vtkGDCMThreadedImageReader2::vtkSetMacro (NumberOfScalarComponents , int)
- 25.344.2.27 vtkGDCMThreadedImageReader2::vtkSetMacro (LoadOverlays , int)

25.344.2.28 `vtkGDCMThreadedImageReader2::vtkSetMacro (Shift , double)`

25.344.2.29 `vtkGDCMThreadedImageReader2::vtkSetMacro (Scale , double)`

25.344.2.30 `vtkGDCMThreadedImageReader2::vtkSetMacro (UseShiftScale , int)`

25.344.2.31 `vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataOrigin , double)`

25.344.2.32 `vtkGDCMThreadedImageReader2::vtkSetVector3Macro (DataSpacing , double)`

25.344.2.33 `vtkGDCMThreadedImageReader2::vtkSetVector6Macro (DataExtent , int)`

25.344.2.34 `vtkGDCMThreadedImageReader2::vtkTypeRevisionMacro (vtkGDCMThreadedImageReader2 ,
vtkThreadedImageAlgorithm)`

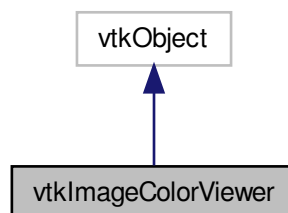
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader2.h](#)

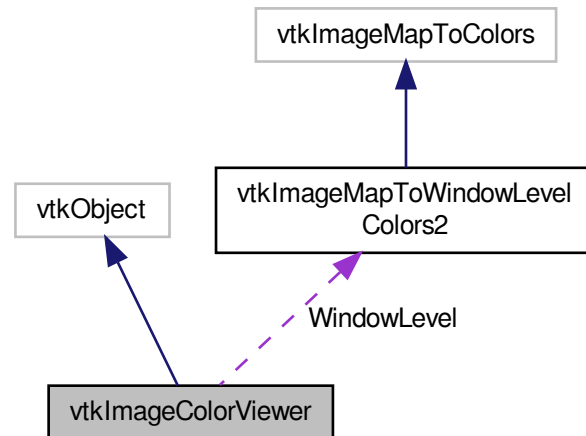
25.345 vtkImageColorViewer Class Reference

```
#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](#)

25.345.1 Detailed Description

Examples:

[gdcmrionplan.cxx](#), and [gdcmrtpplan.cxx](#).

25.345.2 Member Enumeration Documentation

25.345.2.1 anonymous enum

Enumerator

SLICE_ORIENTATION_YZ
SLICE_ORIENTATION_XZ
SLICE_ORIENTATION_XY

25.345.3 Constructor & Destructor Documentation

25.345.3.1 `vtkImageColorViewer::vtkImageColorViewer ()` [protected]

25.345.3.2 `vtkImageColorViewer::~~vtkImageColorViewer ()` [protected]

25.345.4 Member Function Documentation

25.345.4.1 `virtual void vtkImageColorViewer::AddInput (vtkImageData * input)` [virtual]

25.345.4.2 `virtual void vtkImageColorViewer::AddInputConnection (vtkAlgorithmOutput * input)` [virtual]

25.345.4.3 `virtual double vtkImageColorViewer::GetColorLevel ()` [virtual]

25.345.4.4 `virtual double vtkImageColorViewer::GetColorWindow ()` [virtual]

25.345.4.5 `virtual vtkImageData* vtkImageColorViewer::GetInput ()` [virtual]

25.345.4.6 `virtual int vtkImageColorViewer::GetOffScreenRendering ()` [virtual]

25.345.4.7 `double vtkImageColorViewer::GetOverlayVisibility ()`

- 25.345.4.8 `virtual int* vtkImageColorViewer::GetPosition () [virtual]`
- 25.345.4.9 `virtual int* vtkImageColorViewer::GetSize () [virtual]`
- 25.345.4.10 `virtual int vtkImageColorViewer::GetSliceMax () [virtual]`
- 25.345.4.11 `virtual int vtkImageColorViewer::GetSliceMin () [virtual]`
- 25.345.4.12 `virtual void vtkImageColorViewer::GetSliceRange (int range[2]) [inline],[virtual]`
- 25.345.4.13 `virtual void vtkImageColorViewer::GetSliceRange (int & min, int & max) [virtual]`
- 25.345.4.14 `virtual int* vtkImageColorViewer::GetSliceRange () [virtual]`
- 25.345.4.15 `virtual const char* vtkImageColorViewer::GetWindowName () [virtual]`
- 25.345.4.16 `virtual void vtkImageColorViewer::InstallPipeline () [protected],[virtual]`
- 25.345.4.17 `static vtkImageColorViewer* vtkImageColorViewer::New () [static]`

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

- 25.345.4.18 `void vtkImageColorViewer::PrintSelf (ostream & os, vtkIndent indent)`
- 25.345.4.19 `virtual void vtkImageColorViewer::Render (void) [virtual]`

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

- 25.345.4.20 `virtual void vtkImageColorViewer::SetColorLevel (double s) [virtual]`
- 25.345.4.21 `virtual void vtkImageColorViewer::SetColorWindow (double s) [virtual]`
- 25.345.4.22 `virtual void vtkImageColorViewer::SetDisplayId (void * a) [virtual]`
- 25.345.4.23 `virtual void vtkImageColorViewer::SetInput (vtkImageData * in) [virtual]`

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

- 25.345.4.24 `virtual void vtkImageColorViewer::SetInputConnection (vtkAlgorithmOutput * input) [virtual]`
- 25.345.4.25 `virtual void vtkImageColorViewer::SetOffScreenRendering (int) [virtual]`
- 25.345.4.26 `void vtkImageColorViewer::SetOverlayVisibility (double vis)`

25.345.4.27 `virtual void vtkImageColorViewer::SetParentId (void * a) [virtual]`

25.345.4.28 `virtual void vtkImageColorViewer::SetPosition (int a, int b) [virtual]`

25.345.4.29 `virtual void vtkImageColorViewer::SetPosition (int a[2]) [inline],[virtual]`

References `SetPosition()`.

Referenced by `SetPosition()`.

25.345.4.30 `virtual void vtkImageColorViewer::SetRenderer (vtkRenderer * arg) [virtual]`

25.345.4.31 `virtual void vtkImageColorViewer::SetRenderWindow (vtkRenderWindow * arg) [virtual]`

25.345.4.32 `virtual void vtkImageColorViewer::SetSize (int a, int b) [virtual]`

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

25.345.4.33 `virtual void vtkImageColorViewer::SetSize (int a[2]) [inline],[virtual]`

References `SetSize()`.

Referenced by `SetSize()`.

25.345.4.34 `virtual void vtkImageColorViewer::SetSlice (int s) [virtual]`

25.345.4.35 `virtual void vtkImageColorViewer::SetSliceOrientation (int orientation) [virtual]`

25.345.4.36 `virtual void vtkImageColorViewer::SetSliceOrientationToXY () [inline],[virtual]`

References `SLICE_ORIENTATION_XY`.

25.345.4.37 `virtual void vtkImageColorViewer::SetSliceOrientationToXZ () [inline],[virtual]`

References `SLICE_ORIENTATION_XZ`.

25.345.4.38 `virtual void vtkImageColorViewer::SetSliceOrientationToYZ () [inline],[virtual]`

References `SLICE_ORIENTATION_YZ`.

25.345.4.39 `virtual void vtkImageColorViewer::SetupInteractor (vtkRenderWindowInteractor *) [virtual]`

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

- 25.345.4.40 virtual void vtkImageColorViewer::SetWindowId (void * a) [virtual]
- 25.345.4.41 virtual void vtkImageColorViewer::UnInstallPipeline () [protected],[virtual]
- 25.345.4.42 virtual void vtkImageColorViewer::UpdateDisplayExtent () [virtual]
- 25.345.4.43 virtual void vtkImageColorViewer::UpdateOrientation () [protected],[virtual]
- 25.345.4.44 vtkImageColorViewer::VTK_LEGACY (int GetWholeZMin())
- 25.345.4.45 vtkImageColorViewer::VTK_LEGACY (int GetWholeZMax())
- 25.345.4.46 vtkImageColorViewer::VTK_LEGACY (int GetZSlice())
- 25.345.4.47 vtkImageColorViewer::VTK_LEGACY (void SetZSliceint)
- 25.345.4.48 vtkImageColorViewer::vtkBooleanMacro (OffScreenRendering , int)
- 25.345.4.49 vtkImageColorViewer::vtkGetMacro (SliceOrientation , int)
- 25.345.4.50 vtkImageColorViewer::vtkGetMacro (Slice , int)
- 25.345.4.51 vtkImageColorViewer::vtkGetObjectMacro (RenderWindow , vtkRenderWindow)
- 25.345.4.52 vtkImageColorViewer::vtkGetObjectMacro (Renderer , vtkRenderer)
- 25.345.4.53 vtkImageColorViewer::vtkGetObjectMacro (ImageActor , vtkImageActor)
- 25.345.4.54 vtkImageColorViewer::vtkGetObjectMacro (WindowLevel , vtkImageMapToWindowLevelColors2)
- 25.345.4.55 vtkImageColorViewer::vtkGetObjectMacro (InteractorStyle , vtkInteractorStyleImage)
- 25.345.4.56 vtkImageColorViewer::vtkTypeRevisionMacro (vtkImageColorViewer , vtkObject)

25.345.5 Member Data Documentation

- 25.345.5.1 int vtkImageColorViewer::FirstRender [protected]
- 25.345.5.2 vtkImageActor* vtkImageColorViewer::ImageActor [protected]
- 25.345.5.3 vtkRenderWindowInteractor* vtkImageColorViewer::Interactor [protected]
- 25.345.5.4 vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle [protected]
- 25.345.5.5 vtkImageActor* vtkImageColorViewer::OverlayImageActor [protected]
- 25.345.5.6 vtkRenderer* vtkImageColorViewer::Renderer [protected]
- 25.345.5.7 vtkRenderWindow* vtkImageColorViewer::RenderWindow [protected]
- 25.345.5.8 int vtkImageColorViewer::Slice [protected]

25.345.5.9 `int vtkImageColorViewer::SliceOrientation` [protected]

25.345.5.10 `vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel` [protected]

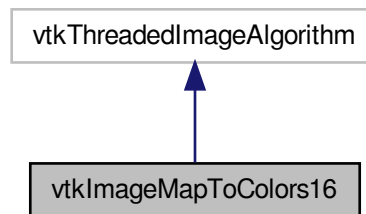
The documentation for this class was generated from the following file:

- [vtkImageColorViewer.h](#)

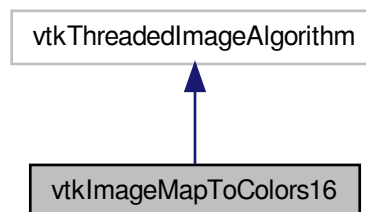
25.346 `vtkImageMapToColors16` Class Reference

```
#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](#)

25.346.1 Constructor & Destructor Documentation

25.346.1.1 [vtkImageMapToColors16::vtkImageMapToColors16](#) () [protected]

25.346.1.2 [vtkImageMapToColors16::~~vtkImageMapToColors16](#) () [protected]

25.346.2 Member Function Documentation

25.346.2.1 virtual unsigned long [vtkImageMapToColors16::GetMTime](#) () [virtual]

25.346.2.2 static [vtkImageMapToColors16](#)* [vtkImageMapToColors16::New](#) () [static]

- 25.346.2.3 void vtkImageMapToColors16::PrintSelf (ostream & *os*, vtkIndent *indent*)
- 25.346.2.4 virtual int vtkImageMapToColors16::RequestData (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*) [protected],[virtual]
- 25.346.2.5 virtual int vtkImageMapToColors16::RequestInformation (vtkInformation * , vtkInformationVector ** , vtkInformationVector *) [protected],[virtual]
- 25.346.2.6 virtual void vtkImageMapToColors16::SetLookupTable (vtkScalarsToColors *) [virtual]
- 25.346.2.7 void vtkImageMapToColors16::SetOutputFormatToLuminance () [inline]
- 25.346.2.8 void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha () [inline]
- 25.346.2.9 void vtkImageMapToColors16::SetOutputFormatToRGB () [inline]
- 25.346.2.10 void vtkImageMapToColors16::SetOutputFormatToRGBA () [inline]
- 25.346.2.11 void vtkImageMapToColors16::ThreadedRequestData (vtkInformation * *request*, vtkInformationVector ** *inputVector*, vtkInformationVector * *outputVector*, vtkImageData *** *inData*, vtkImageData ** *outData*, int *extent*[6], int *id*) [protected]
- 25.346.2.12 vtkImageMapToColors16::vtkBooleanMacro (PassAlphaToOutput , int)
- 25.346.2.13 vtkImageMapToColors16::vtkGetMacro (OutputFormat , int)
- 25.346.2.14 vtkImageMapToColors16::vtkGetMacro (ActiveComponent , int)
- 25.346.2.15 vtkImageMapToColors16::vtkGetMacro (PassAlphaToOutput , int)
- 25.346.2.16 vtkImageMapToColors16::vtkGetObjectMacro (LookupTable , vtkScalarsToColors)
- 25.346.2.17 vtkImageMapToColors16::vtkSetMacro (OutputFormat , int)
- 25.346.2.18 vtkImageMapToColors16::vtkSetMacro (ActiveComponent , int)
- 25.346.2.19 vtkImageMapToColors16::vtkSetMacro (PassAlphaToOutput , int)
- 25.346.2.20 vtkImageMapToColors16::vtkTypeRevisionMacro (vtkImageMapToColors16 , vtkThreadedImageAlgorithm)

25.346.3 Member Data Documentation

- 25.346.3.1 int vtkImageMapToColors16::ActiveComponent [protected]
- 25.346.3.2 int vtkImageMapToColors16::DataWasPassed [protected]
- 25.346.3.3 vtkScalarsToColors* vtkImageMapToColors16::LookupTable [protected]
- 25.346.3.4 int vtkImageMapToColors16::OutputFormat [protected]

25.346.3.5 int vtkImageMapToColors16::PassAlphaToOutput [protected]

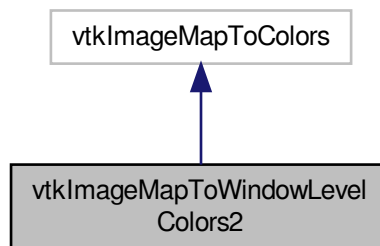
The documentation for this class was generated from the following file:

- [vtkImageMapToColors16.h](#)

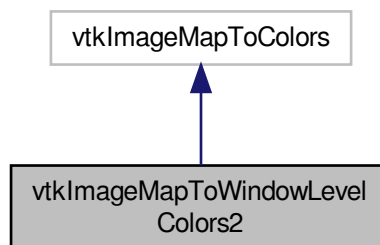
25.347 vtkImageMapToWindowLevelColors2 Class Reference

```
#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](#)

25.347.1 Constructor & Destructor Documentation

25.347.1.1 [vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2](#) () [protected]

25.347.1.2 [vtkImageMapToWindowLevelColors2::~~vtkImageMapToWindowLevelColors2](#) () [protected]

25.347.2 Member Function Documentation

25.347.2.1 static [vtkImageMapToWindowLevelColors2*](#) [vtkImageMapToWindowLevelColors2::New](#) () [static]

25.347.2.2 void [vtkImageMapToWindowLevelColors2::PrintSelf](#) ([ostream](#) & *os*, [vtkIndent](#) *indent*)

25.347.2.3 virtual int [vtkImageMapToWindowLevelColors2::RequestData](#) ([vtkInformation](#) * *request*, [vtkInformationVector](#) ** *inputVector*, [vtkInformationVector](#) * *outputVector*) [protected],[virtual]

25.347.2.4 virtual int [vtkImageMapToWindowLevelColors2::RequestInformation](#) ([vtkInformation](#) * , [vtkInformationVector](#) ** , [vtkInformationVector](#) *) [protected],[virtual]

25.347.2.5 void [vtkImageMapToWindowLevelColors2::ThreadedRequestData](#) ([vtkInformation](#) * *request*, [vtkInformationVector](#) ** *inputVector*, [vtkInformationVector](#) * *outputVector*, [vtkImageData](#) *** *inData*, [vtkImageData](#) ** *outData*, int *extent*[6], int *id*) [protected]

25.347.2.6 [vtkImageMapToWindowLevelColors2::vtkGetMacro](#) ([Window](#) , double)

25.347.2.7 [vtkImageMapToWindowLevelColors2::vtkGetMacro](#) ([Level](#) , double)

25.347.2.8 vtkImageMapToWindowLevelColors2::vtkSetMacro (Window , double)

25.347.2.9 vtkImageMapToWindowLevelColors2::vtkSetMacro (Level , double)

25.347.2.10 vtkImageMapToWindowLevelColors2::vtkTypeRevisionMacro (vtkImageMapToWindowLevelColors2 ,
vtkImageMapToColors)

25.347.3 Member Data Documentation

25.347.3.1 double vtkImageMapToWindowLevelColors2::Level [protected]

25.347.3.2 double vtkImageMapToWindowLevelColors2::Window [protected]

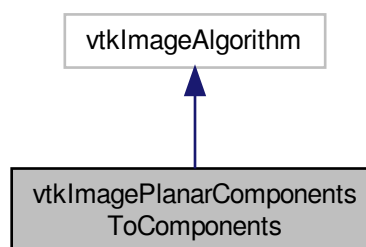
The documentation for this class was generated from the following file:

- [vtkImageMapToWindowLevelColors2.h](#)

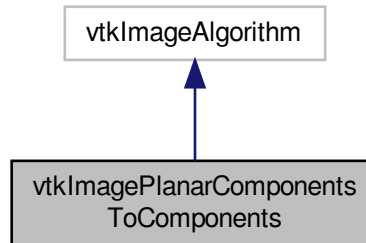
25.348 vtkImagePlanarComponentsToComponents Class Reference

```
#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 *)

25.348.1 Constructor & Destructor Documentation

25.348.1.1 `vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ()` [protected]

25.348.1.2 `vtkImagePlanarComponentsToComponents::~~vtkImagePlanarComponentsToComponents ()` [inline], [protected]

25.348.2 Member Function Documentation

25.348.2.1 `static vtkImagePlanarComponentsToComponents* vtkImagePlanarComponentsToComponents::New ()` [static]

25.348.2.2 `void vtkImagePlanarComponentsToComponents::PrintSelf (ostream & os, vtkIndent indent)`

25.348.2.3 `virtual int vtkImagePlanarComponentsToComponents::RequestData (vtkInformation *, vtkInformationVector **, vtkInformationVector *) [protected],[virtual]`

25.348.2.4 `vtkImagePlanarComponentsToComponents::vtkTypeRevisionMacro (vtkImagePlanarComponentsToComponents , vtkImageAlgorithm)`

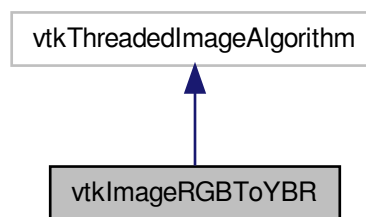
The documentation for this class was generated from the following file:

- [vtkImagePlanarComponentsToComponents.h](#)

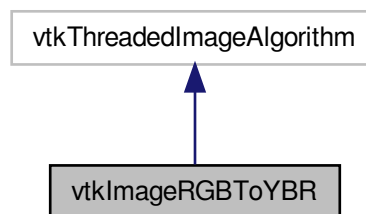
25.349 vtkImageRGBToYBR Class Reference

```
#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*)

25.349.1 Constructor & Destructor Documentation

25.349.1.1 [vtkImageRGBToYBR::vtkImageRGBToYBR](#) () [[protected](#)]

25.349.1.2 [vtkImageRGBToYBR::~~vtkImageRGBToYBR](#) () [[inline](#)], [[protected](#)]

25.349.2 Member Function Documentation

25.349.2.1 static [vtkImageRGBToYBR*](#) [vtkImageRGBToYBR::New](#) () [[static](#)]

25.349.2.2 void [vtkImageRGBToYBR::PrintSelf](#) ([ostream](#) & *os*, [vtkIndent](#) *indent*)

25.349.2.3 void [vtkImageRGBToYBR::ThreadedExecute](#) ([vtkImageData](#) * *inData*, [vtkImageData](#) * *outData*, int *ext*[6], int *id*)
[[protected](#)]

25.349.2.4 [vtkImageRGBToYBR::vtkTypeRevisionMacro](#) ([vtkImageRGBToYBR](#) , [vtkThreadedImageAlgorithm](#))

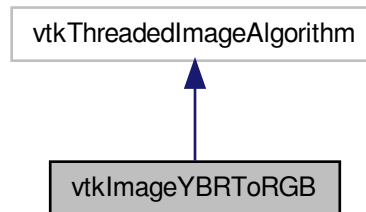
The documentation for this class was generated from the following file:

- [vtkImageRGBToYBR.h](#)

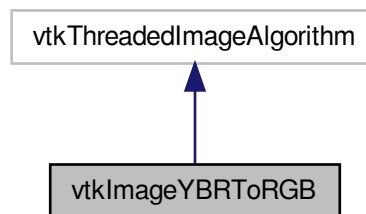
25.350 vtkImageYBRTToRGB Class Reference

```
#include <vtkImageYBRTToRGB.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)

25.350.1 Constructor & Destructor Documentation

25.350.1.1 `vtkImageYBRToRGB::vtkImageYBRToRGB ()` `[protected]`

25.350.1.2 `vtkImageYBRToRGB::~~vtkImageYBRToRGB ()` `[inline], [protected]`

25.350.2 Member Function Documentation

25.350.2.1 `static vtkImageYBRToRGB* vtkImageYBRToRGB::New ()` `[static]`

25.350.2.2 `void vtkImageYBRToRGB::PrintSelf (ostream & os, vtkIndent indent)`

25.350.2.3 `void vtkImageYBRToRGB::ThreadedExecute (vtkImageData * inData, vtkImageData * outData, int ext[6], int id)`
`[protected]`

25.350.2.4 `vtkImageYBRToRGB::vtkTypeRevisionMacro (vtkImageYBRToRGB , vtkThreadedImageAlgorithm)`

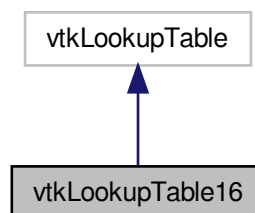
The documentation for this class was generated from the following file:

- [vtkImageYBRToRGB.h](#)

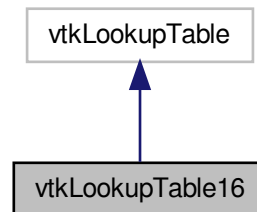
25.351 vtkLookupTable16 Class Reference

```
#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 size=256, int ext=256)
- [~vtkLookupTable16](#) ()
- void [MapScalarsThroughTable2](#) (void *input, unsigned char *output, int inputDataType, int numberOfValues, int inputIncrement, int outputFormat)

Protected Attributes

- vtkUnsignedShortArray * [Table16](#)

25.351.1 Constructor & Destructor Documentation

25.351.1.1 `vtkLookupTable16::vtkLookupTable16 (int size = 256, int ext = 256)` [protected]

25.351.1.2 `vtkLookupTable16::~~vtkLookupTable16 ()` [protected]

25.351.2 Member Function Documentation

- 25.351.2.1 void vtkLookupTable16::Build ()
- 25.351.2.2 unsigned short* vtkLookupTable16::GetPointer (const vtkIdType *id*) [inline]
- 25.351.2.3 void vtkLookupTable16::MapScalarsThroughTable2 (void * *input*, unsigned char * *output*, int *inputDataType*, int *numberOfValues*, int *inputIncrement*, int *outputFormat*) [protected]
- 25.351.2.4 static vtkLookupTable16* vtkLookupTable16::New () [static]
- 25.351.2.5 void vtkLookupTable16::PrintSelf (ostream & *os*, vtkIndent *indent*)
- 25.351.2.6 void vtkLookupTable16::SetNumberOfTableValues (vtkIdType *number*)
- 25.351.2.7 vtkLookupTable16::vtkTypeRevisionMacro (vtkLookupTable16 , vtkLookupTable)
- 25.351.2.8 unsigned char * vtkLookupTable16::WritePointer (const vtkIdType *id*, const int *number*) [inline]

References Table16.

25.351.3 Member Data Documentation

- 25.351.3.1 vtkUnsignedShortArray* vtkLookupTable16::Table16 [protected]

Referenced by WritePointer().

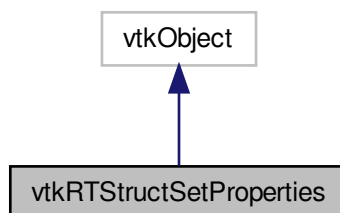
The documentation for this class was generated from the following file:

- [vtkLookupTable16.h](#)

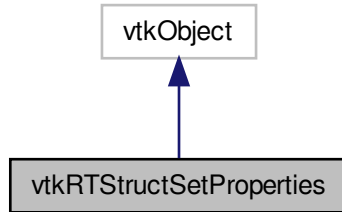
25.352 vtkRTStructSetProperties Class Reference

```
#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 *refframerefid, 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](#)

25.352.1 Detailed Description

Examples:

[GenerateRTSTRUCT.cxx](#).

25.352.2 Constructor & Destructor Documentation

25.352.2.1 [vtkRTStructSetProperties::vtkRTStructSetProperties \(\)](#) [protected]

25.352.2.2 [vtkRTStructSetProperties::~~vtkRTStructSetProperties \(\)](#) [protected]

25.352.3 Member Function Documentation

- 25.352.3.1 void vtkRTStructSetProperties::AddContourReferencedFrameOfReference (vtkIdType *pdnum*, const char * *classuid*, const char * *instanceuid*)
- 25.352.3.2 void vtkRTStructSetProperties::AddReferencedFrameOfReference (const char * *classuid*, const char * *instanceuid*)
- 25.352.3.3 void vtkRTStructSetProperties::AddStructureSetROI (int *roinumber*, const char * *refframerefid*, const char * *roiname*, const char * *ROIGenerationAlgorithm*, const char * *ROIDescription* = 0)
- 25.352.3.4 void vtkRTStructSetProperties::AddStructureSetROIObservation (int *refnumber*, int *observationnumber*, const char * *rtroiinterpretedtype*, const char * *roiinterpreter*, const char * *roiobservationlabel* = 0)
- 25.352.3.5 virtual void vtkRTStructSetProperties::Clear () [virtual]
- 25.352.3.6 virtual void vtkRTStructSetProperties::DeepCopy (vtkRTStructSetProperties * *p*) [virtual]
- 25.352.3.7 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID (vtkIdType *pdnum*, vtkIdType *id*)
- 25.352.3.8 const char* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID (vtkIdType *pdnum*, vtkIdType *id*)
- 25.352.3.9 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ()
- 25.352.3.10 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences (vtkIdType *pdnum*)
- 25.352.3.11 vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ()
- 25.352.3.12 vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ()
- 25.352.3.13 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID (vtkIdType *id*)
- 25.352.3.14 const char* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID (vtkIdType *id*)
- 25.352.3.15 int vtkRTStructSetProperties::GetStructureSetObservationNumber (vtkIdType *id*)
- 25.352.3.16 const char* vtkRTStructSetProperties::GetStructureSetROIDescription (vtkIdType *id*)
- 25.352.3.17 const char* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm (vtkIdType)
- 25.352.3.18 const char* vtkRTStructSetProperties::GetStructureSetROIName (vtkIdType)
- 25.352.3.19 int vtkRTStructSetProperties::GetStructureSetROINumber (vtkIdType *id*)
- 25.352.3.20 const char* vtkRTStructSetProperties::GetStructureSetROIObservationLabel (vtkIdType *id*)
- 25.352.3.21 const char* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID (vtkIdType)
- 25.352.3.22 const char* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType (vtkIdType *id*)

25.352.3.23 `static vtkRTStructSetProperties* vtkRTStructSetProperties::New ()` `[static]`

Examples:

[GenerateRTSTRUCT.cxx](#).

25.352.3.24 `void vtkRTStructSetProperties::PrintSelf (ostream & os, vtkIndent indent)`

25.352.3.25 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetLabel)`

25.352.3.26 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetName)`

25.352.3.27 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetDate)`

25.352.3.28 `vtkRTStructSetProperties::vtkGetStringMacro (StructureSetTime)`

25.352.3.29 `vtkRTStructSetProperties::vtkGetStringMacro (SOPInstanceUID)`

25.352.3.30 `vtkRTStructSetProperties::vtkGetStringMacro (StudyInstanceUID)`

25.352.3.31 `vtkRTStructSetProperties::vtkGetStringMacro (SeriesInstanceUID)`

25.352.3.32 `vtkRTStructSetProperties::vtkGetStringMacro (ReferenceSeriesInstanceUID)`

25.352.3.33 `vtkRTStructSetProperties::vtkGetStringMacro (ReferenceFrameOfReferenceUID)`

25.352.3.34 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetLabel)`

25.352.3.35 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetName)`

25.352.3.36 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetDate)`

25.352.3.37 `vtkRTStructSetProperties::vtkSetStringMacro (StructureSetTime)`

25.352.3.38 `vtkRTStructSetProperties::vtkSetStringMacro (SOPInstanceUID)`

25.352.3.39 `vtkRTStructSetProperties::vtkSetStringMacro (StudyInstanceUID)`

25.352.3.40 `vtkRTStructSetProperties::vtkSetStringMacro (SeriesInstanceUID)`

25.352.3.41 `vtkRTStructSetProperties::vtkSetStringMacro (ReferenceSeriesInstanceUID)`

25.352.3.42 `vtkRTStructSetProperties::vtkSetStringMacro (ReferenceFrameOfReferenceUID)`

25.352.3.43 `vtkRTStructSetProperties::vtkTypeRevisionMacro (vtkRTStructSetProperties , vtkObject)`

25.352.4 Member Data Documentation

25.352.4.1 `vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals` `[protected]`

25.352.4.2 `char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID` `[protected]`

25.352.4.3 `char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID` [protected]

25.352.4.4 `char* vtkRTStructSetProperties::SeriesInstanceUID` [protected]

25.352.4.5 `char* vtkRTStructSetProperties::SOPInstanceUID` [protected]

25.352.4.6 `char* vtkRTStructSetProperties::StructureSetDate` [protected]

25.352.4.7 `char* vtkRTStructSetProperties::StructureSetLabel` [protected]

25.352.4.8 `char* vtkRTStructSetProperties::StructureSetName` [protected]

25.352.4.9 `char* vtkRTStructSetProperties::StructureSetTime` [protected]

25.352.4.10 `char* vtkRTStructSetProperties::StudyInstanceUID` [protected]

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

- [vtkRTStructSetProperties.h](#)

25.353 gdcm::Waveform Class Reference

[Waveform](#) class.

```
#include <gdcmWaveform.h>
```

Public Member Functions

- [Waveform](#) ()

25.353.1 Detailed Description

[Waveform](#) class.

25.353.2 Constructor & Destructor Documentation

25.353.2.1 `gdcm::Waveform::Waveform ()` [inline]

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

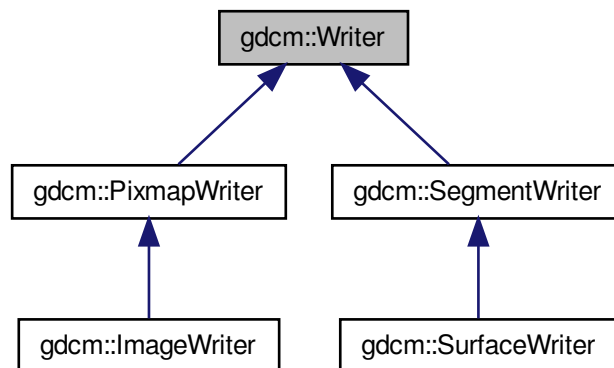
- [gdcmWaveform.h](#)

25.354 gdcm::Writer Class Reference

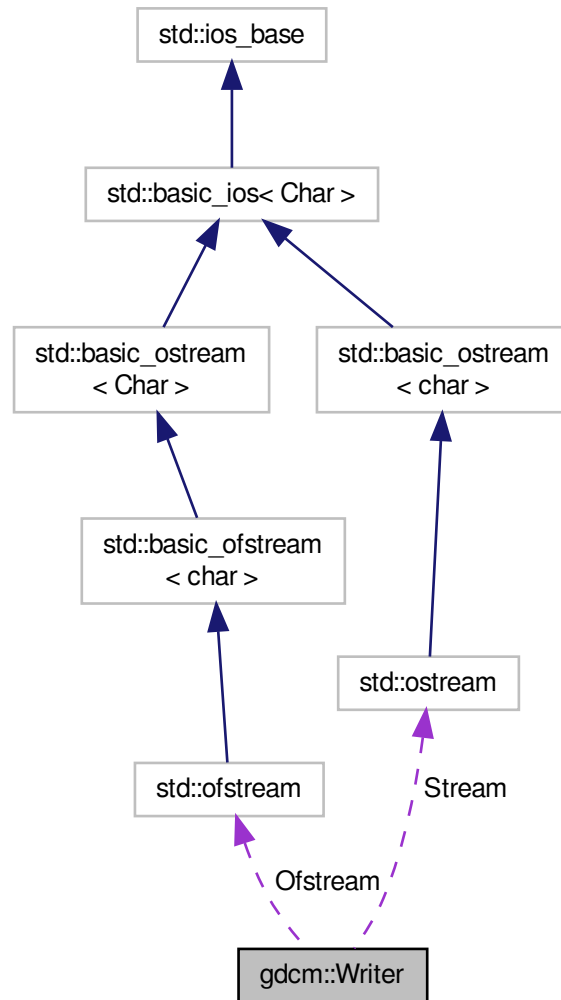
[Writer](#) ala DOM (Document [Object](#) Model) This class is a non-validating writer, it will only performs well- formedness check only.

```
#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](#)

25.354.1 Detailed Description

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

Detailed description here To avoid GDCM being yet another broken DICOM lib we try to be user level and avoid writing illegal stuff (odd length, non-zero value for [Item](#) start/end length ...) Therefore you cannot (well unless you are really smart) write DICOM with even length tag. All the checks are consider basics:

- Correct Meta Information Header (see [gdcm::FileMetaInformation](#))
- Zero value for [Item](#) Length (0xfffe, 0xe00d/0xe0dd)
- Even length for any elements
- Alphabetical order for elements (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:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting_All_Resolution.cxx](#), [Fake_Image_Using_Stream_Image_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenerateDICOMDIR.cs](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

25.354.2 Constructor & Destructor Documentation

25.354.2.1 `gdcm::Writer::Writer ()`

25.354.2.2 `virtual gdcm::Writer::~~Writer ()` `[virtual]`

25.354.3 Member Function Documentation

25.354.3.1 `void gdcm::Writer::CheckFileMetaInformationOff ()` `[inline]`

Examples:

[FixBrokenJ2K.cxx](#), and [HelloWorld.cxx](#).

25.354.3.2 `void gdcm::Writer::CheckFileMetaInformationOn ()` `[inline]`

25.354.3.3 `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](#), [iU22tomultisc.cxx](#), [pmsct_rgb1.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

25.354.3.4 `std::ostream* gdcm::Writer::GetStreamPtr () const` `[inline]`, `[protected]`

25.354.3.5 `void gdcm::Writer::SetCheckFileMetaInformation (bool b)` `[inline]`

Undocumented function, do not use (= leave default)

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [PatchFile.cxx](#).

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

Set/Get the DICOM file ([DataSet](#) + Header)

Examples:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [DuplicatePCDE.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakelImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

25.354.3.7 void `gdcm::Writer::SetFileName` (const char * *filename_native*)

Set the filename of DICOM file to write:

Examples:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [GenFakelImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

25.354.3.8 void `gdcm::Writer::SetStream` (std::ostream & *output_stream*) [inline]

Set user ostream buffer.

25.354.3.9 void `gdcm::Writer::SetWriteDataSetOnly` (bool *b*) [inline],[protected]

25.354.3.10 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:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct_rgb1.cxx](#), and [rle2img.cxx](#).

25.354.4 Friends And Related Function Documentation

25.354.4.1 friend class `StreamImageWriter` [friend]

25.354.5 Member Data Documentation

25.354.5.1 std::ofstream* `gdcm::Writer::Ofstream` [protected]

25.354.5.2 std::ostream* `gdcm::Writer::Stream` [protected]

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

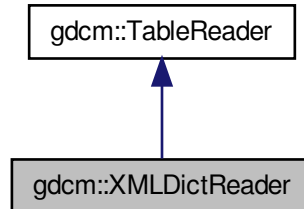
- [gdcmWriter.h](#)

25.355 gdcm::XMLDictReader Class Reference

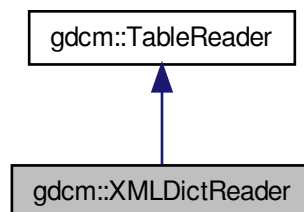
Class for representing a [XMLDictReader](#).

```
#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)

25.355.1 Detailed Description

Class for representing a [XMLDictReader](#).

Note

bla Will read the DICOMV3.xml file

25.355.2 Constructor & Destructor Documentation

25.355.2.1 `gdcm::XMLDictReader::XMLDictReader ()`

25.355.2.2 `gdcm::XMLDictReader::~~XMLDictReader ()` `[inline]`

25.355.3 Member Function Documentation

25.355.3.1 `void gdcm::XMLDictReader::CharacterDataHandler (const char * data, int length)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

25.355.3.2 `void gdcm::XMLDictReader::EndElement (const char * name)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

25.355.3.3 `const Dict& gdcm::XMLDictReader::GetDict ()` `[inline]`

25.355.3.4 `void gdcm::XMLDictReader::HandleDescription (const char ** atts)` `[protected]`

25.355.3.5 `void gdcm::XMLDictReader::HandleEntry (const char ** atts)` `[protected]`

25.355.3.6 `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:

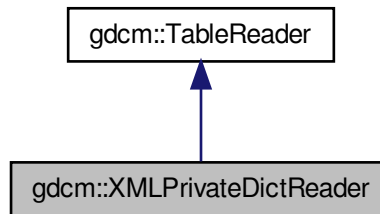
- [gdcmXMLDictReader.h](#)

25.356 gdcm::XMLPrivateDictReader Class Reference

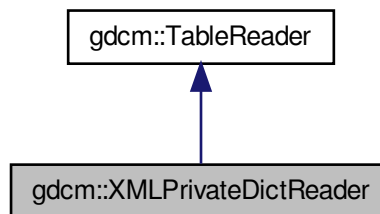
Class for representing a [XMLPrivateDictReader](#).

```
#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)

25.356.1 Detailed Description

Class for representing a [XMLPrivateDictReader](#).

Note

bla Will read the Private.xml file

25.356.2 Constructor & Destructor Documentation

25.356.2.1 `gdcm::XMLPrivateDictReader::XMLPrivateDictReader ()`

25.356.2.2 `gdcm::XMLPrivateDictReader::~~XMLPrivateDictReader ()` `[inline]`

25.356.3 Member Function Documentation

25.356.3.1 `void gdcm::XMLPrivateDictReader::CharacterDataHandler (const char * data, int length)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

25.356.3.2 `void gdcm::XMLPrivateDictReader::EndElement (const char * name)` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

25.356.3.3 `const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ()` `[inline]`

25.356.3.4 `void gdcm::XMLPrivateDictReader::HandleDescription (const char ** atts)` `[protected]`

25.356.3.5 `void gdcm::XMLPrivateDictReader::HandleEntry (const char ** atts)` `[protected]`

25.356.3.6 `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 26

File Documentation

26.1 gdcm2pnm.man File Reference

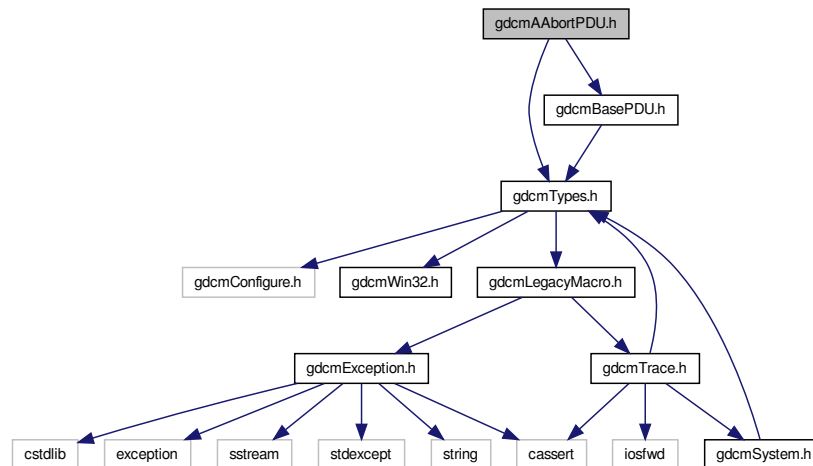
26.2 gdcm2vtk.man File Reference

26.3 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.

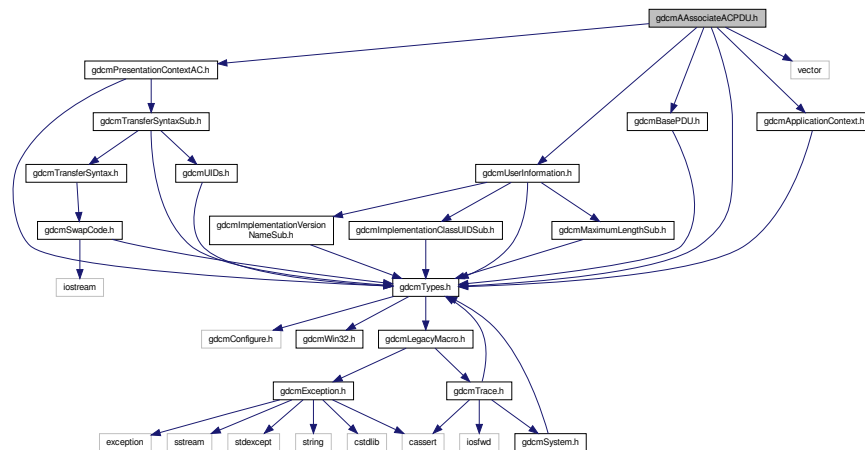
Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.4 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.

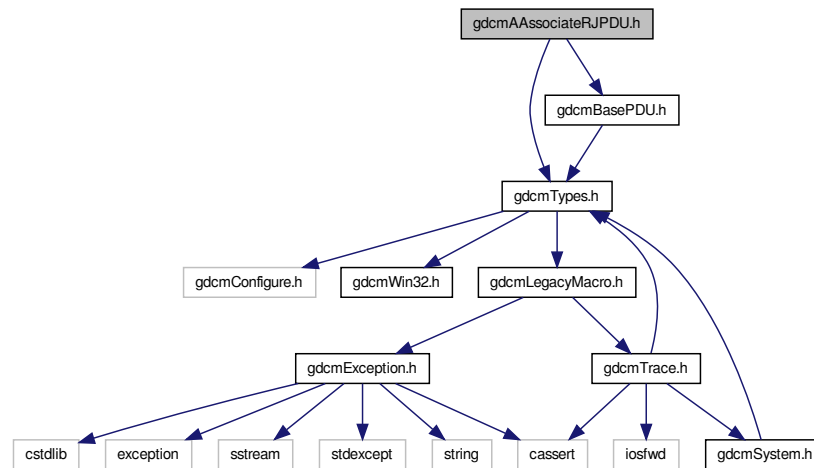
Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.5 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`

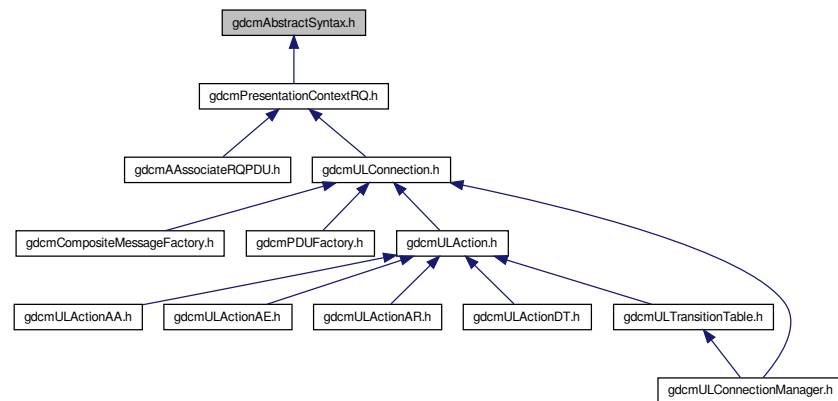
26.6 gdcmAAssociateRQPDU.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmUserInfo.h"
#include "gdcmBasePDU.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

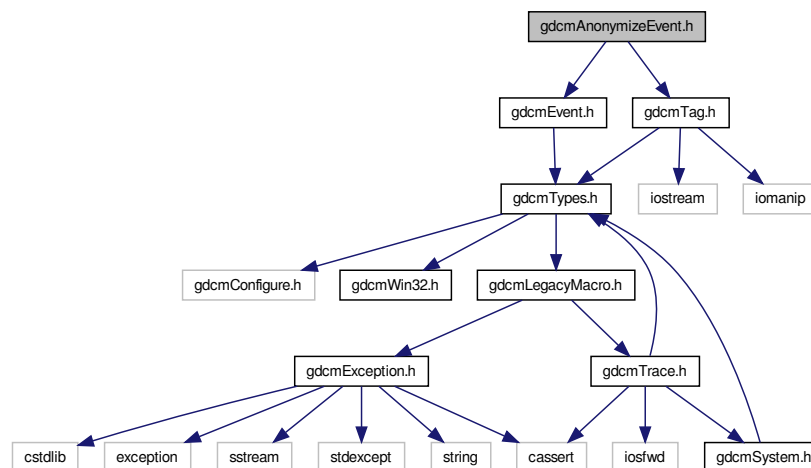
- [gdcm](#)
- [gdcm::network](#)

26.8 gdcmanon.man File Reference

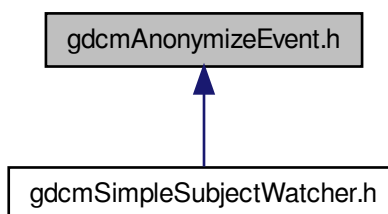
26.9 gdcmAnonymizeEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for `gdcmAnonymizeEvent.h`:



This graph shows which files directly or indirectly include this file:



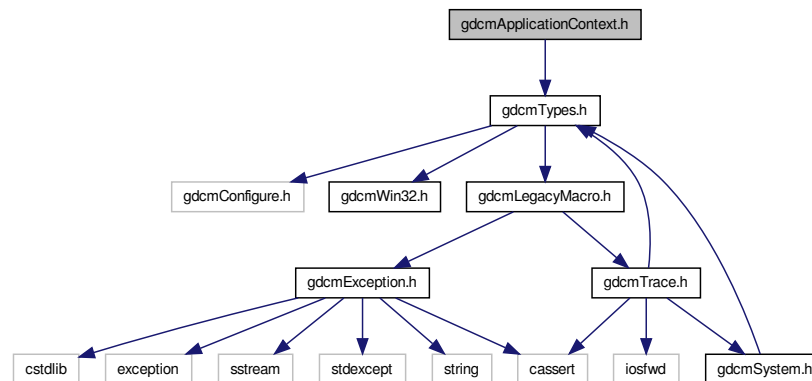
Classes

- class `gdcm::AnonymizeEvent`
AnonymizeEvent Special type of event triggered during the Anonymization process.

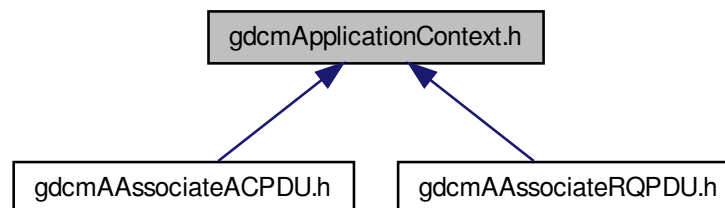
Namespaces

- `gdcm`

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

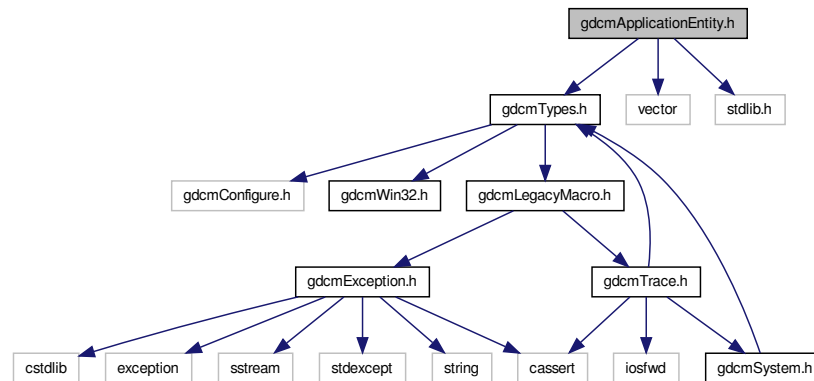
Namespaces

- `gdcm`
- `gdcm::network`

26.12 gdcmApplicationEntity.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <vector>
#include <stdlib.h>
Include dependency graph for gdcmApplicationEntity.h:
```



Classes

- class `gdcm::ApplicationEntity`

ApplicationEntity.

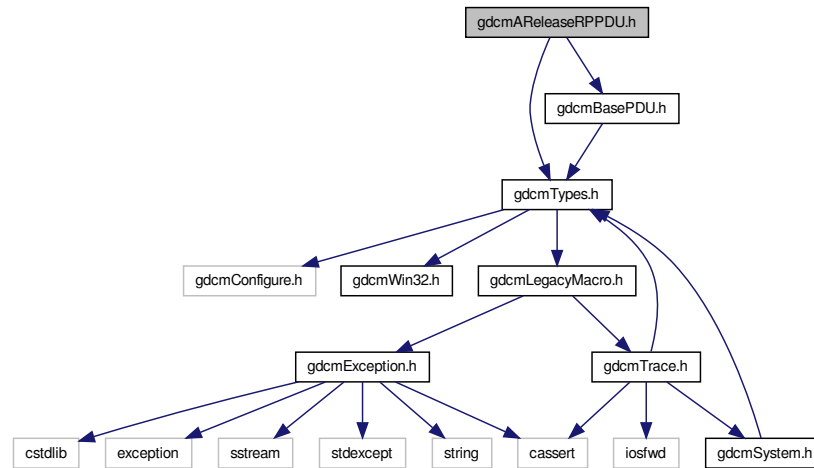
Namespaces

- `gdcm`

26.13 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.

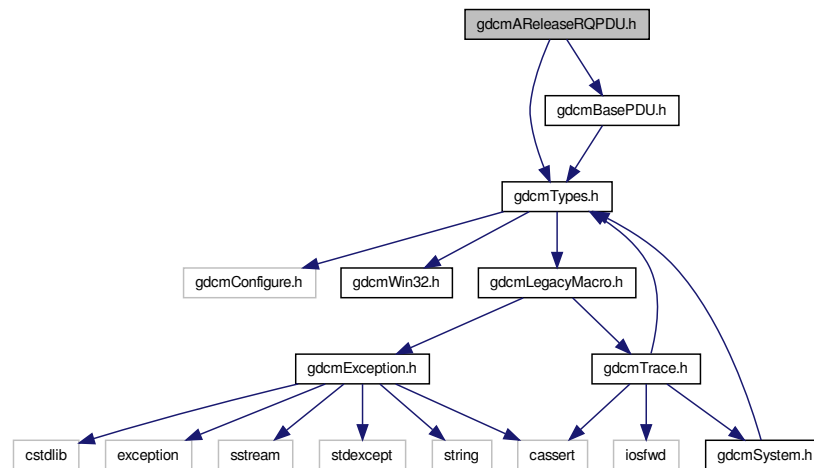
Namespaces

- `gdcm`
- `gdcm::network`

26.14 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```


Include dependency graph for gdcMAReleaseRQPDU.h:



Classes

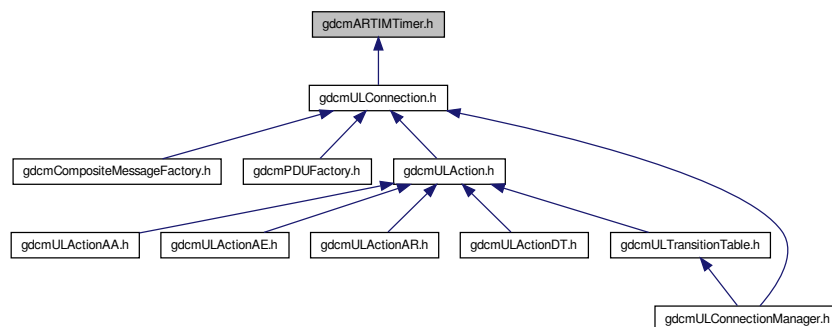
- class [gdcM::network::AReleaseRQPDU](#)
[AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.](#)

Namespaces

- [gdcM](#)
- [gdcM::network](#)

26.15 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.

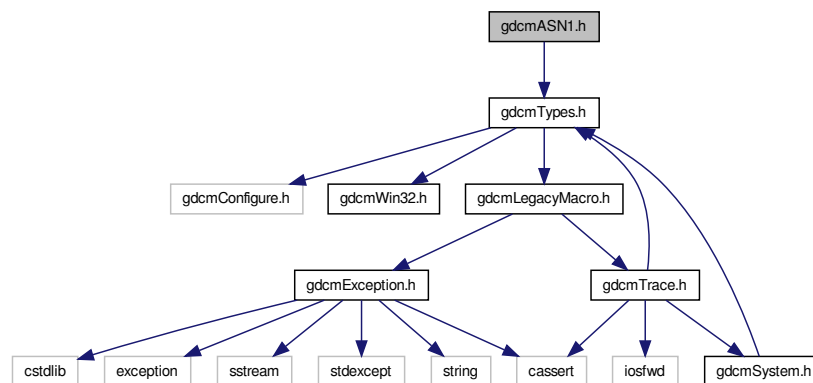
Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.16 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



Classes

- class [gdcm::ASN1](#)

Class for [ASN1](#).

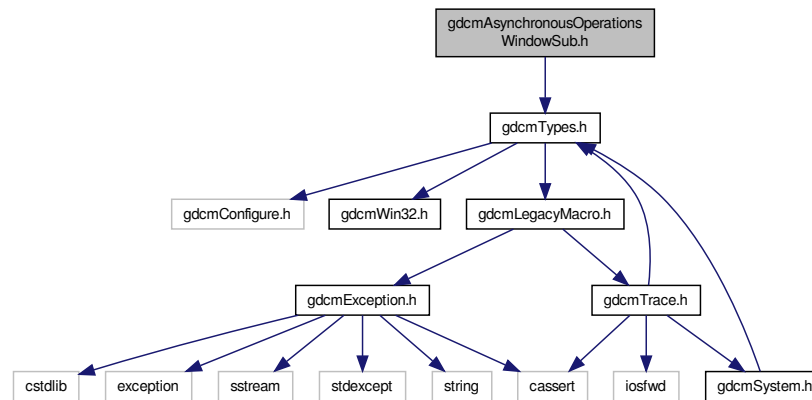
Namespaces

- [gdcm](#)

26.17 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 FIELDS (A-ASSOCIATE-RQ)

Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.18 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>
```

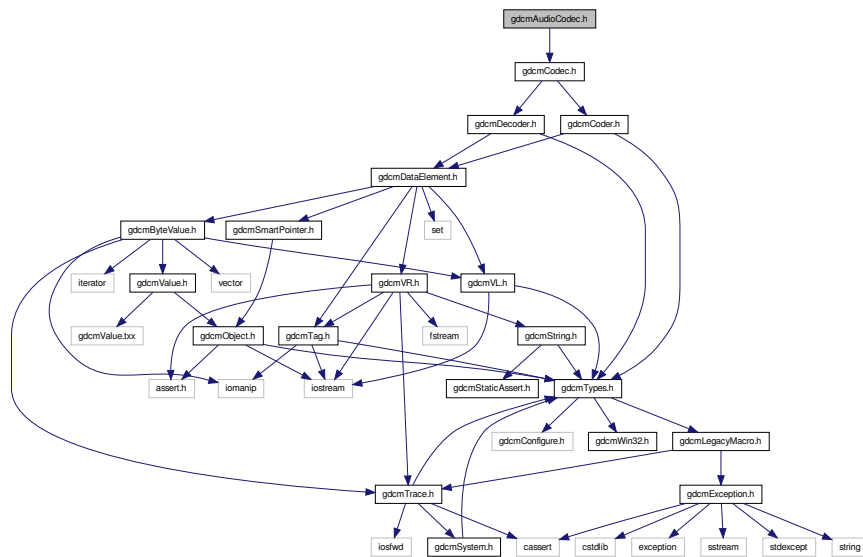

Namespaces

- [gdcm](#)

26.19 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



Classes

- class [gdcm::AudioCodec](#)
AudioCodec.

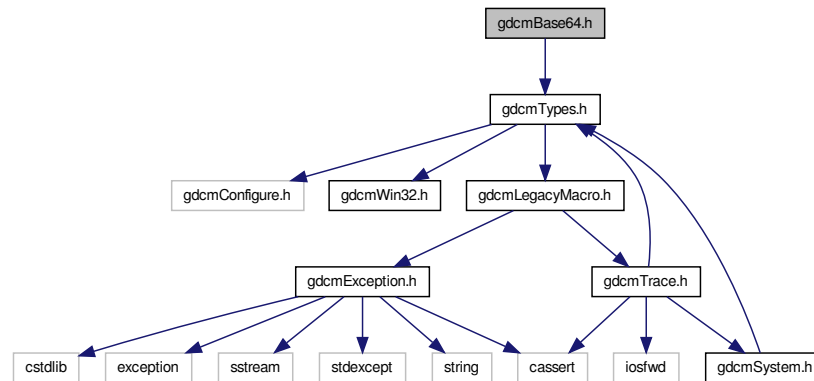
Namespaces

- [gdcm](#)

26.20 gdcmBase64.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmBase64.h`:



Classes

- class [gdcm::Base64](#)

Class for [Base64](#).

Namespaces

- [gdcm](#)

26.21 gdcmBaseCompositeMessage.h File Reference

```

#include "gdcmPresentationDataValue.h"
#include "gdcmBaseRootQuery.h"
#include <vector>

```

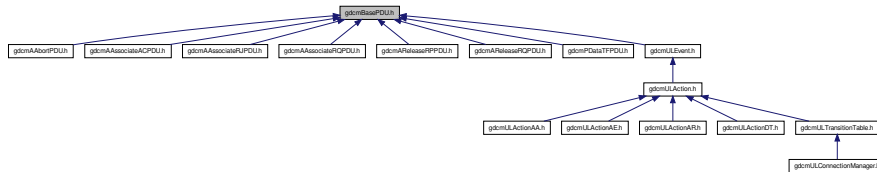
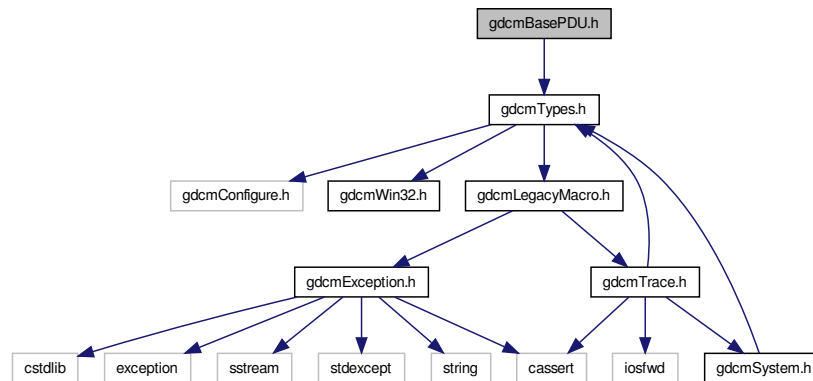
[illegible]

```
graph BT; gdcBase[gdcmBaseCompositeMessage.h] <--> gdcEcho[gdcmCEchoMessages.h]; gdcBase <--> gdcFind[gdcmCFindMessages.h]; gdcBase <--> gdcMove[gdcmCMoveMessages.h]; gdcBase <--> gdcStore[gdcmCStoreMessages.h];
```

- class `gdcm::network::BaseCompositeMessage`

- `gdcm`
- `gdcm::network`

Generated on Sat Dec 21 2013 01:40:25 for GDCM by Doxygen



- class `gdcn::network::BasePDU`
BasePDU base class for PDUs.

- `gdcm`
- `gdcm::network`

```
#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmObject.h"
#include "gdcmQueryPatient.h"
#include "gdcmQueryStudy.h"
#include "gdcmQuerySeries.h"
#include "gdcmQueryImage.h"
```



```

graph TD
    gsdmBasePoolQuery_h[gsdmBasePoolQuery.h] --> gsdmBaseCompositeMessage_h[gsdmBaseCompositeMessage.h]
    gsdmBasePoolQuery_h --> gsdmCompositeNetworkFunctions_h[gsdmCompositeNetworkFunctions.h]
    gsdmBasePoolQuery_h --> gsdmFindPatientPoolQuery_h[gsdmFindPatientPoolQuery.h]
    gsdmBasePoolQuery_h --> gsdmFindStudyPoolQuery_h[gsdmFindStudyPoolQuery.h]
    gsdmBasePoolQuery_h --> gsdmMoveStudyPoolQuery_h[gsdmMoveStudyPoolQuery.h]
    gsdmBasePoolQuery_h --> gsdmQueryFactory_h[gsdmQueryFactory.h]
    gsdmBaseCompositeMessage_h --> gsdmCFindMessages_h[gsdmCFindMessages.h]
    gsdmBaseCompositeMessage_h --> gsdmCEchoMessages_h[gsdmCEchoMessages.h]
    gsdmBaseCompositeMessage_h --> gsdmCStoreMessages_h[gsdmCStoreMessages.h]
    gsdmBaseCompositeMessage_h --> gsdmCMoveMessages_h[gsdmCMoveMessages.h]
    gsdmFindPatientPoolQuery_h --> gsdmMovePatientPoolQuery_h[gsdmMovePatientPoolQuery.h]
    gsdmFindPatientPoolQuery_h --> gsdmMoveStudyPoolQuery_h
  
```

- class `gdcm::BaseRootQuery`

Namespaces

- ## Enumerations

- enum `gdcmm::EQueryLevel` {
 `gdcmm::ePatient` = 0,
 `gdcmm::eStudy` = 1,
 `gdcmm::eSeries` = 2,
 `gdcmm::eImage` = 3 }
- enum `gdcmm::EQueryType` {
 `gdcmm::eFind` = 0,
 `gdcmm::eMove` }

Namespaces

- **gdcm**

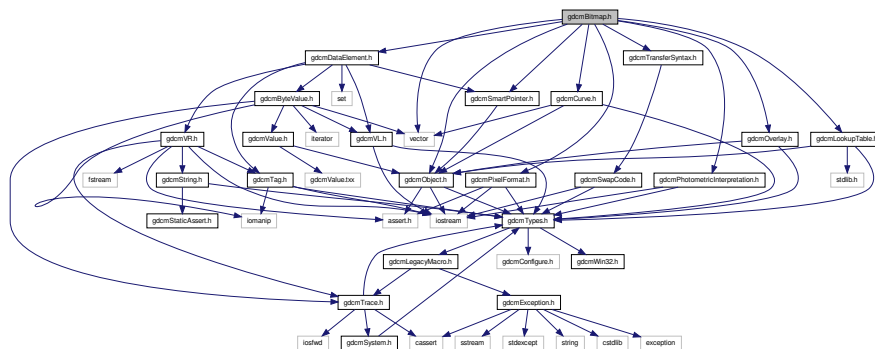
Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const BasicOffsetTable &val)`

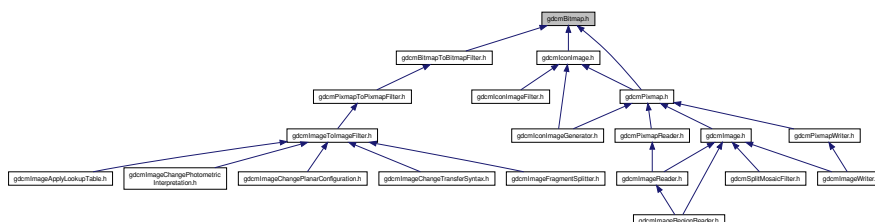
26.25 gdcmBitmap.h File Reference

```
#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:



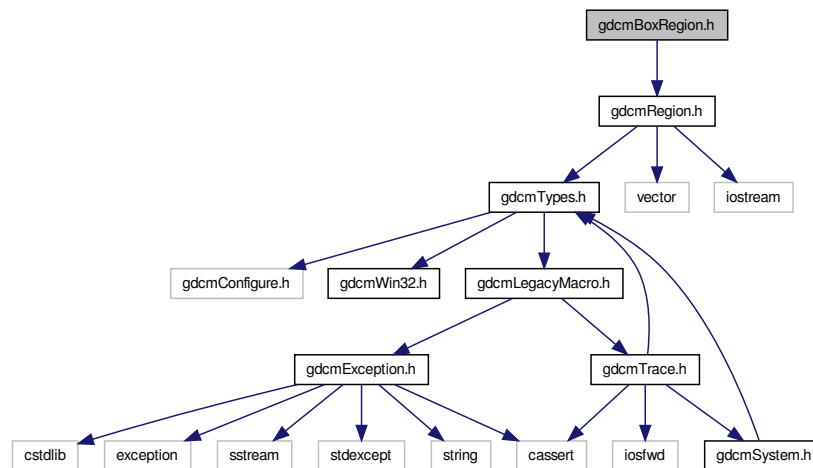
Namespaces

- [gdcm](#)

26.27 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)

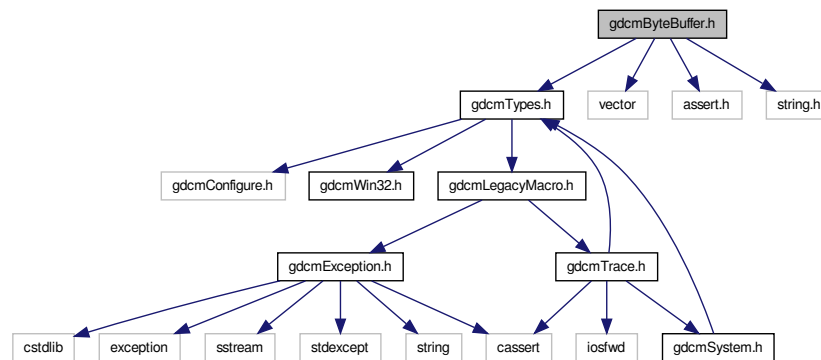
Namespaces

- [gdcm](#)

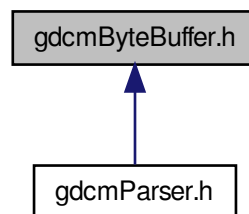
26.28 gdcmByteBuffer.h File Reference

```
#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`

26.29 gdcmByteSwap.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.txx"

```

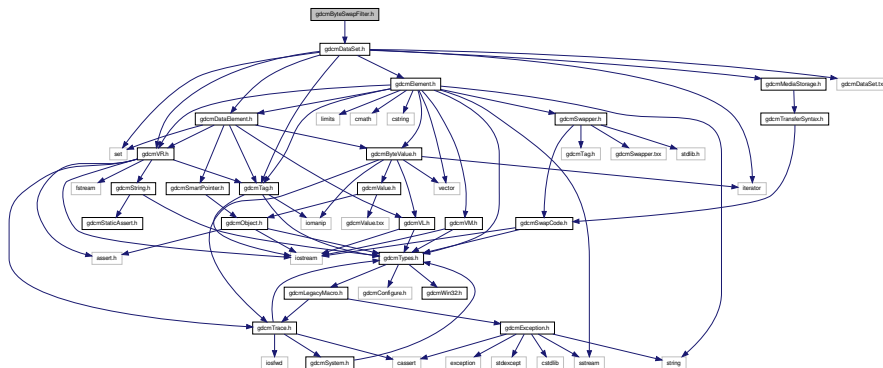
```

graph TD
    gdcmByteSwap.h --> gdcmByteSwap.h
    gdcmByteSwap.h --> gdcmSwapCode.h
    gdcmByteSwap.h --> gdcmByteSwap.box
    gdcmByteSwap.h --> stdlib.h
    gdcmSwapCode.h --> gdcmTypes.h
    gdcmSwapCode.h --> iostream
    gdcmTypes.h --> gdcmConfigure.h
    gdcmTypes.h --> gdcmWin32.h
    gdcmTypes.h --> gdcmLegacyMacro.h
    gdcmTypes.h --> gdcmException.h
    gdcmTypes.h --> gdcmTrace.h
    gdcmTypes.h --> gdcmSystem.h
    gdcmLegacyMacro.h --> gdcmException.h
    gdcmLegacyMacro.h --> gdcmTrace.h
    gdcmException.h --> cstdlib
    gdcmException.h --> exception
    gdcmException.h --> sstream
    gdcmException.h --> stdexcept
    gdcmException.h --> string
    gdcmException.h --> cassert
    gdcmTrace.h --> iosfwd
    gdcmTrace.h --> gdcmSystem.h
    gdcmSystem.h --> gdcmByteSwap.h
  
```

- class `gdcm::ByteSwap< T >`
ByteSwap.

- **gdcm**

```
#include "gdcmDataSet.h"
Include dependency graph for gdcmByteSwapFilter.h:
```



- class `gdcm::ByteSwapFilter`

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

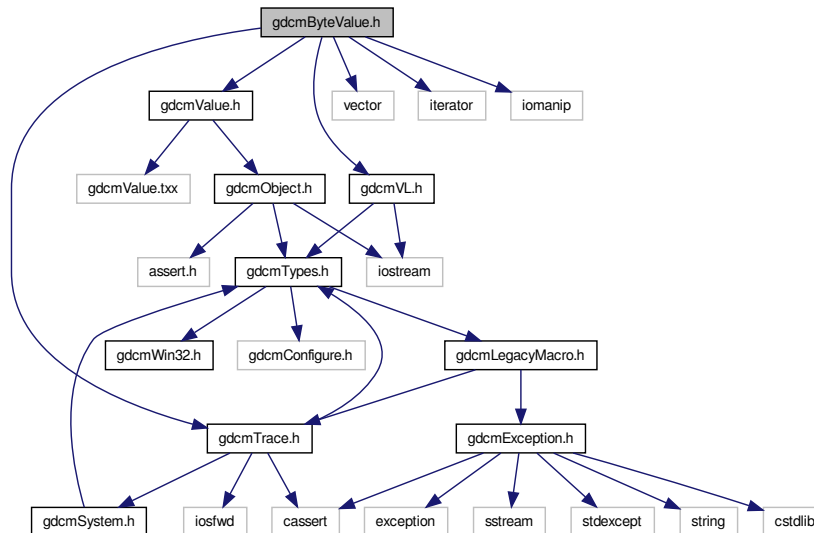
Namespaces

- [gdcm](#)

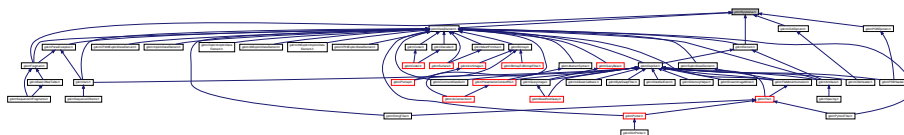
26.31 gdcmByteValue.h File Reference

```
#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)

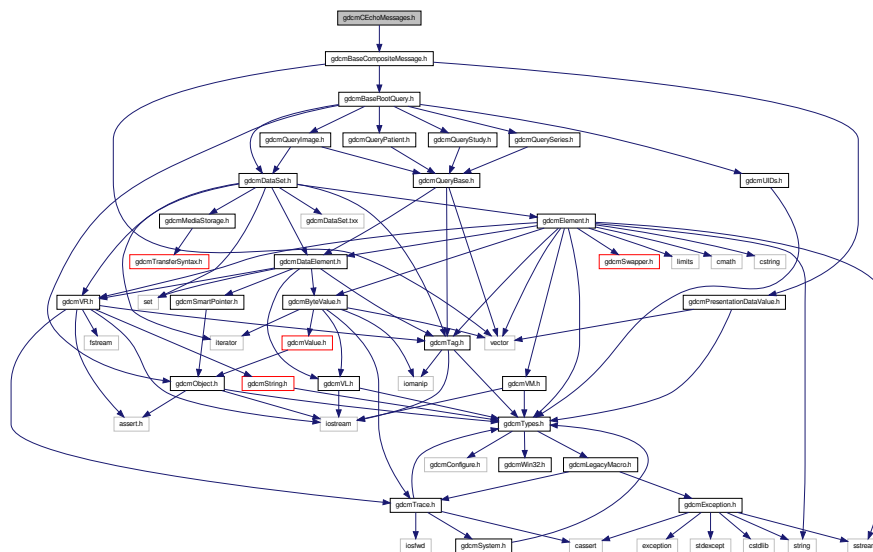
Namespaces

- **gdcm**

26.32 gdcmCEchoMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```

Include dependency graph for gdcmCEchoMessages.h:



Classes

- class `gdcmm::network::CEchoRQ`
CEchoRQ this file defines the messages for the *cecho* action.
- class `gdcmm::network::CEchoRSP`
CEchoRSP this file defines the messages for the *cecho* action.

Namespaces

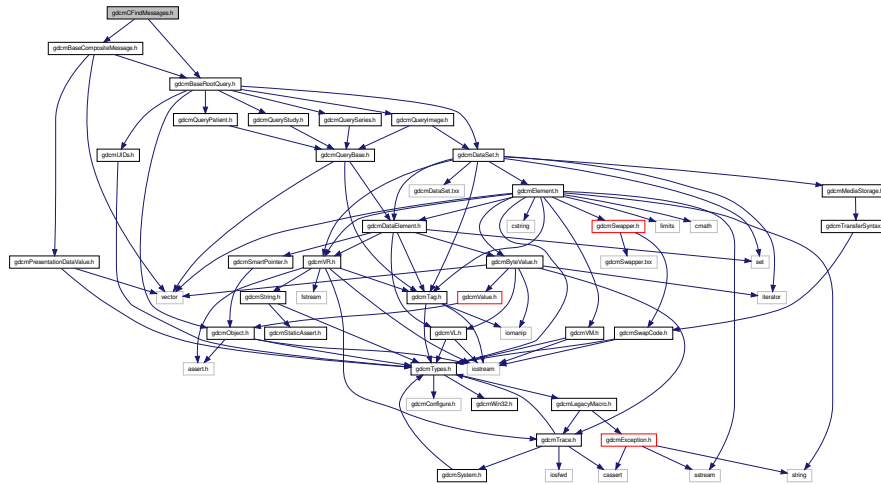
- gdc
- gdc::network

26.33 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`

26.34 gdcmCMoveMessages.h File Reference

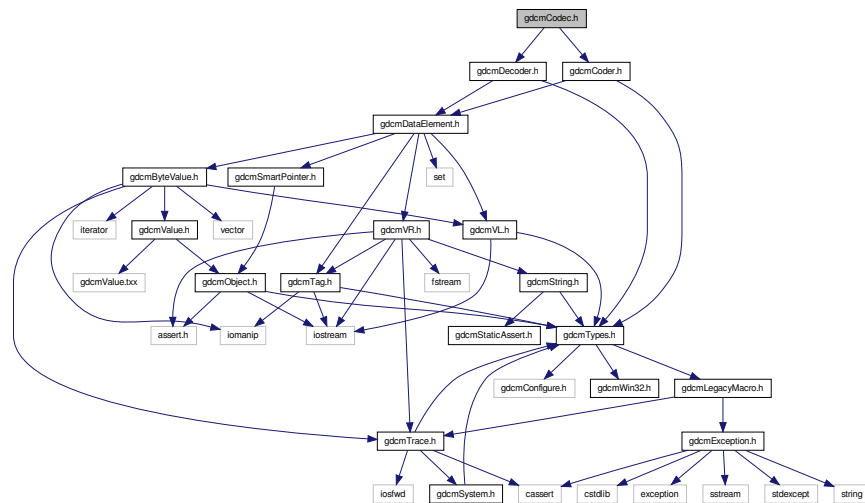
```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```

- 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.

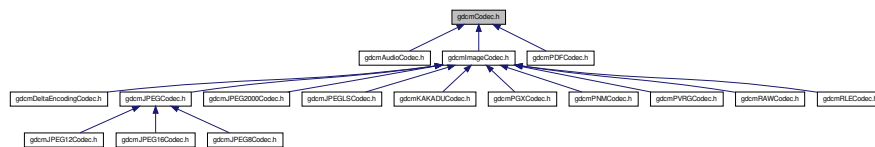
- `gdcm`
- `gdcm::network`

```
#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.

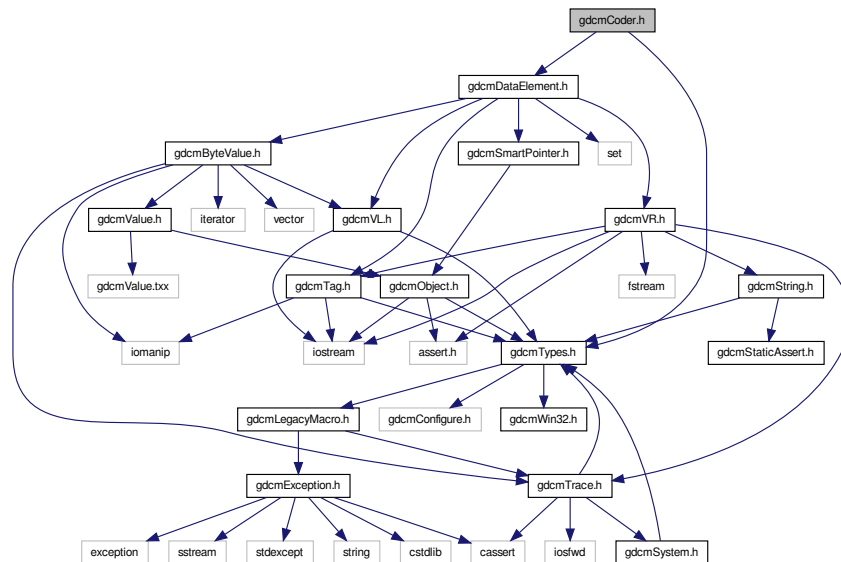
Namespaces

- `gdcm`

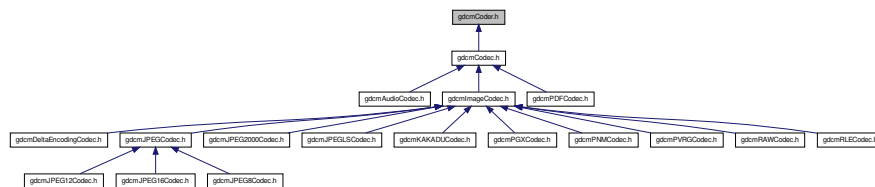
26.36 gdcmCoder.h File Reference

```
#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.

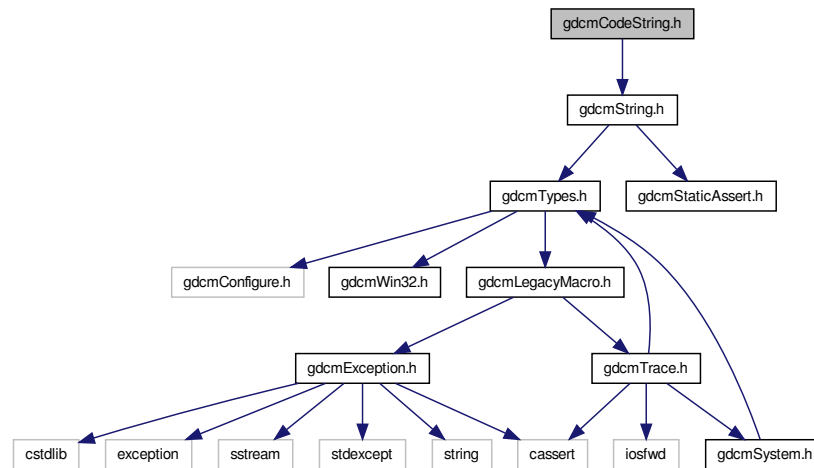
Namespaces

- [gdcm](#)

26.37 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 ctor will properly Trim so that operator== is correct.

Namespaces

- `gdcm`

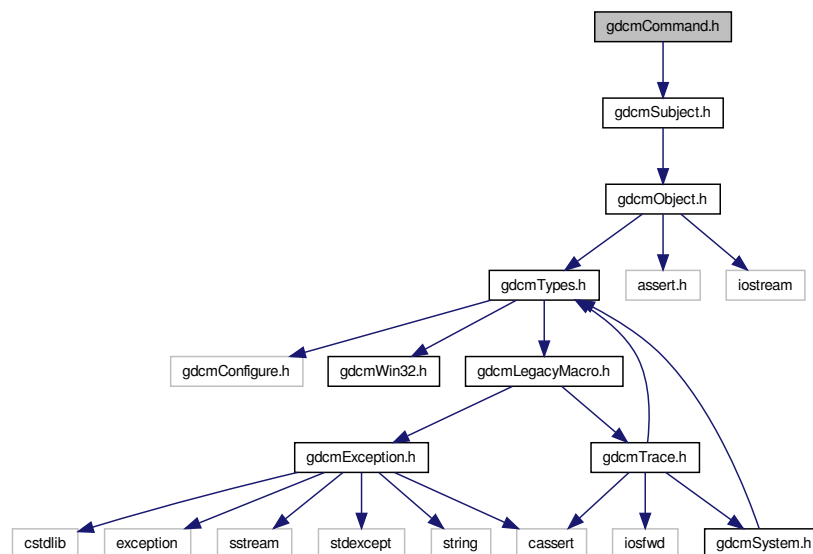
Functions

- `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)`

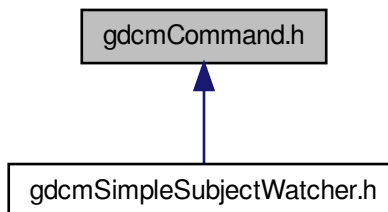
26.38 gdcmCommand.h File Reference

```
#include "gdcmSubject.h"
```

Include dependency graph for gdcMCommand.h:



This graph shows which files directly or indirectly include this file:

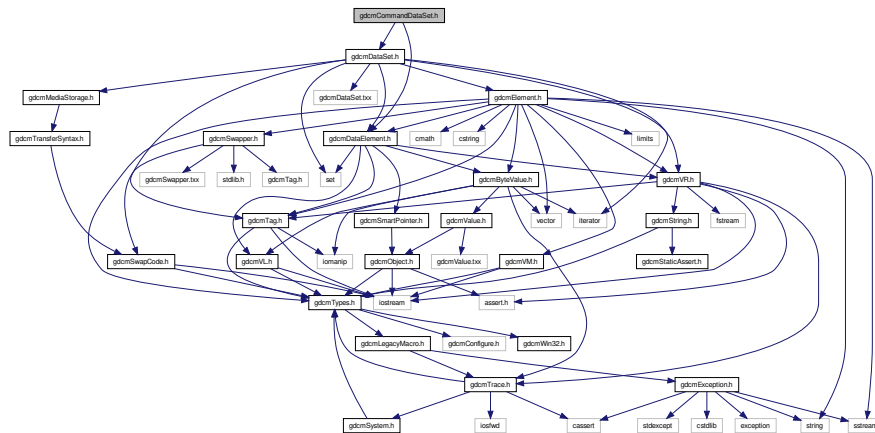


Classes

- 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.

- **gdcm**

```
#include "gdcmDataSet.h"
#include "gdcmDataElement.h"
Include dependency graph for gdcmCommandDataSet.h:
```



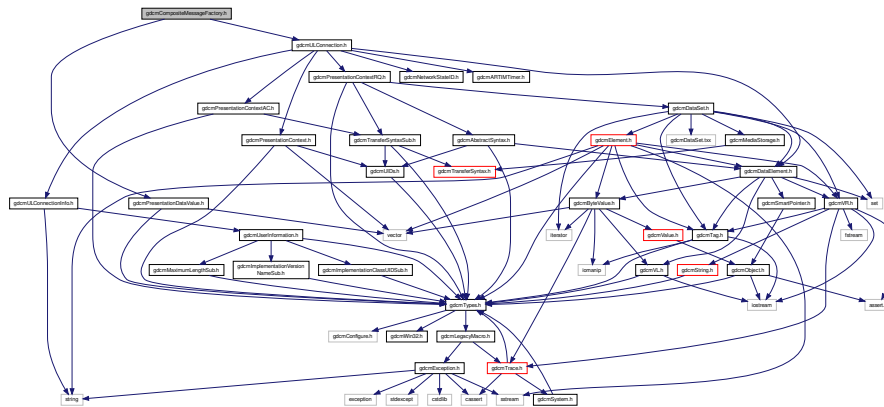
- class `gdcm::CommandDataSet`
Class to represent a `Command DataSet`.

- **gdc**

- `std::ostream & gdcm::operator<< (std::ostream &os, const CommandDataSet &val)`

```
#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](#)

26.41 gdcmCompositeNetworkFunctions.h File Reference

```
#include "gdcmDirectory.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
#include <string>
```


Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities.

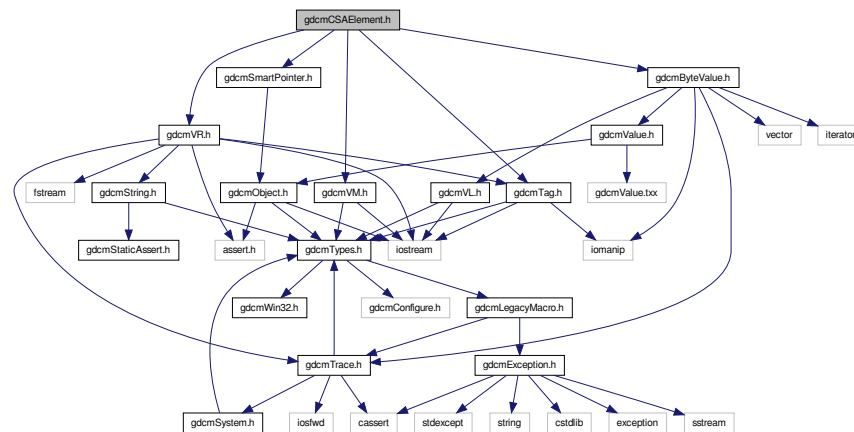
Namespaces

- [gdcm](#)

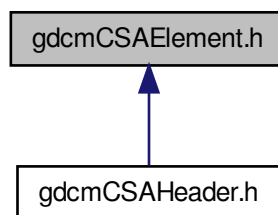
26.46 gdcmCSAElement.h File Reference

```
#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**

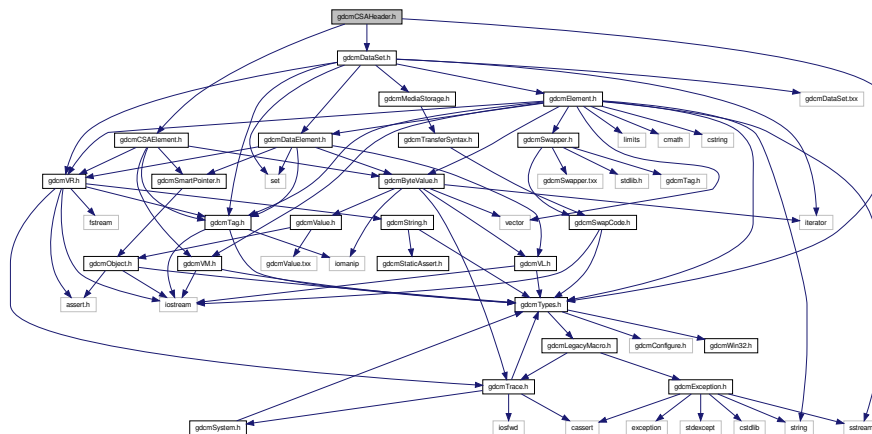
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAElement &val)`

26.47 gdcmCSAHeader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmCSAElement.h"
```

Include dependency graph for gdcmsAHeader.h:



Classes

- class `gdcm::CSAHeader`
Class for CSAHeader.

Namespaces

- gdc

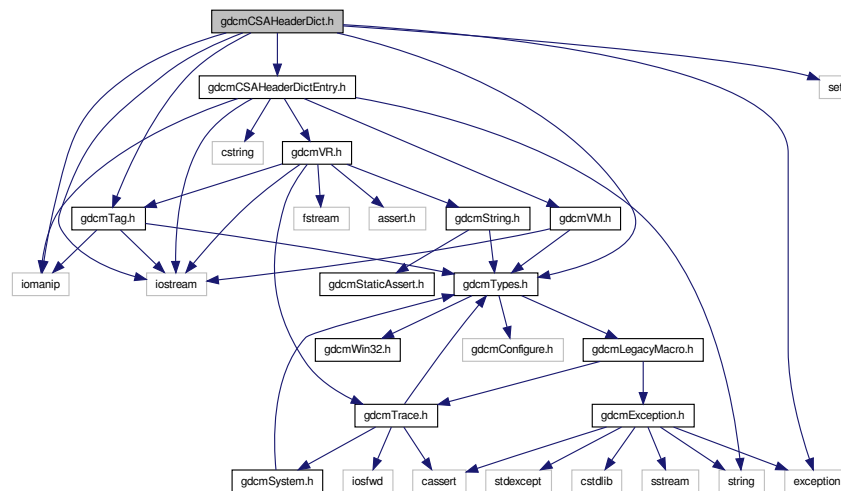
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeader &d)`

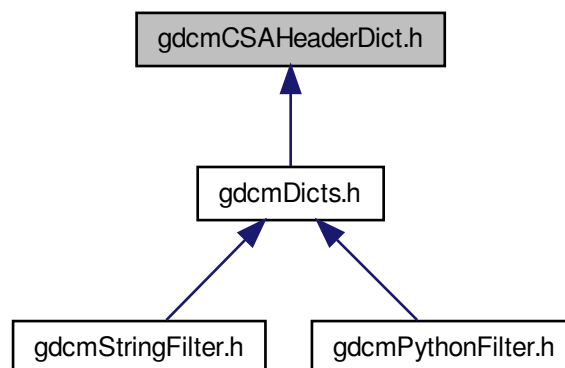
26.48 gdcmCSAHeaderDict.h File Reference

```
#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](#)

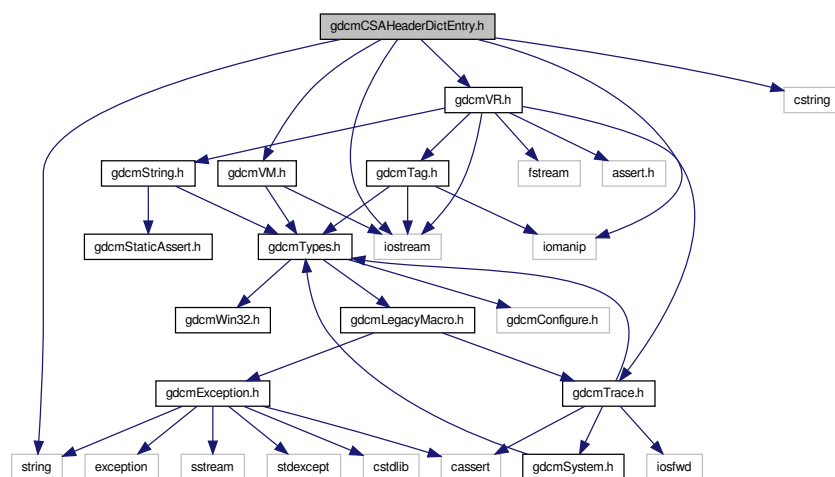
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)`

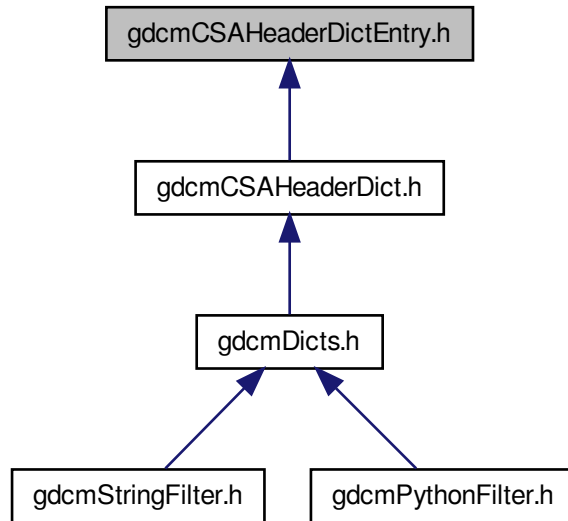
26.49 gdcmCSAHeaderDictEntry.h File Reference

```
#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](#)

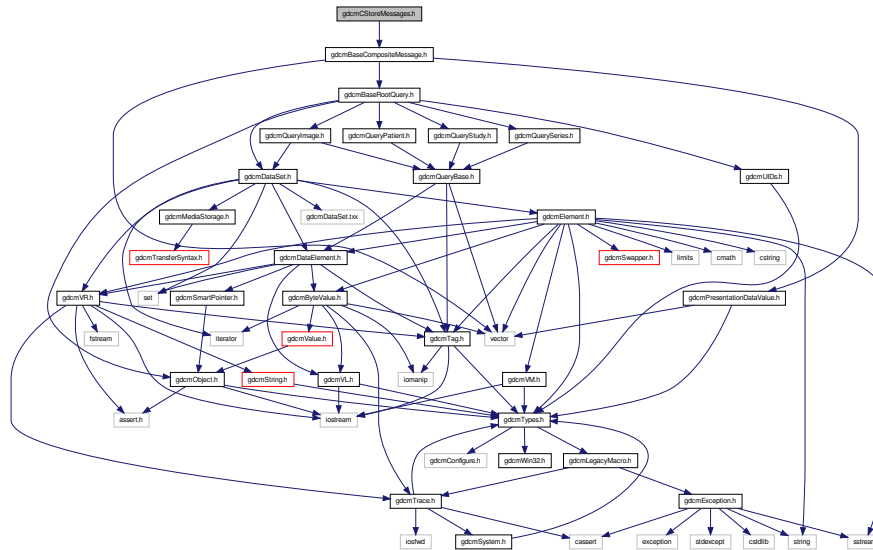
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)`

26.50 gdcmCStoreMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```


Include dependency graph for gdcmCStoreMessages.h:



Classes

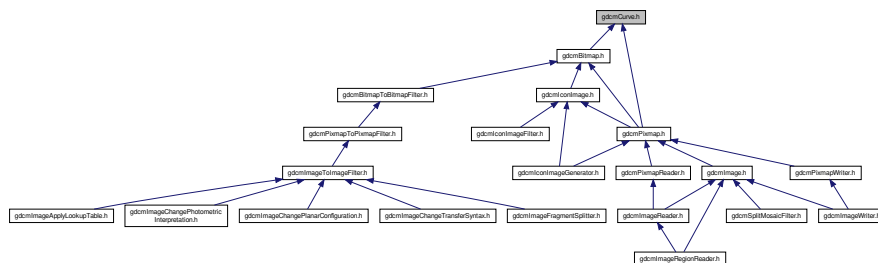
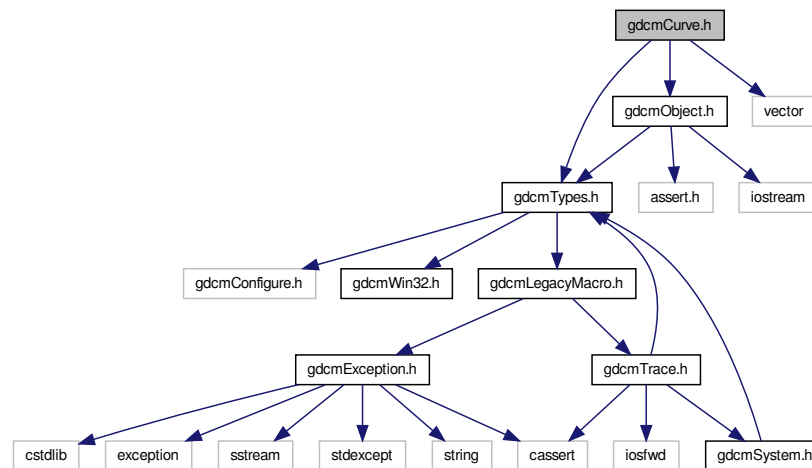
- class [gdcm::network::CStoreRQ](#)
CStoreRQ this file defines the messages for the cecho action.
- class [gdcm::network::CStoreRSP](#)
CStoreRSP this file defines the messages for the cecho action.

Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.51 gdcmCurve.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmObject.h"
#include <vector>
```



- class `gdcm::Curve`

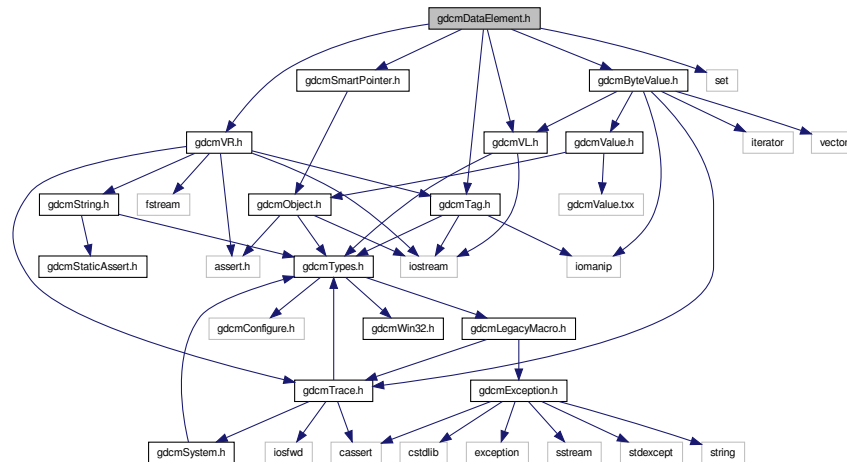
Curve class to handle element 50xx,3000 *Curve* Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

- **gdc**

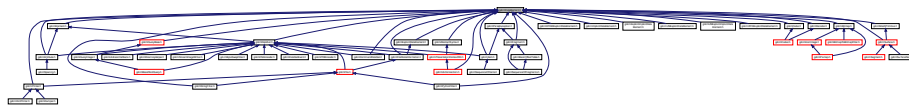
```
#include "gdcmTag.h"
```

```
#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](#)

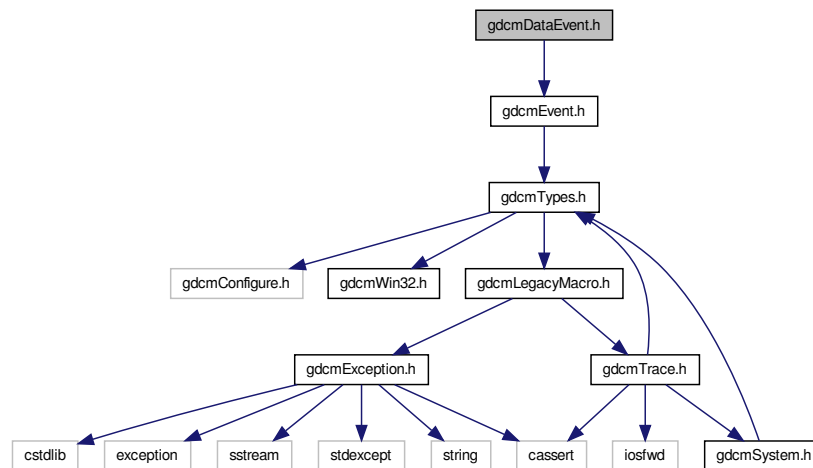
Functions

- bool [gdcm::operator!=](#) (const DataElement &lhs, const DataElement &rhs)
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const DataElement &val)

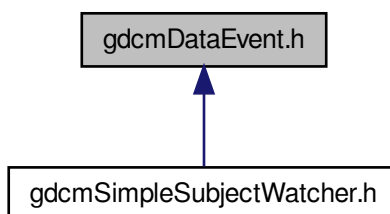
26.53 gdcmDataEvent.h File Reference

```
#include "gdcmEvent.h"
```

Include dependency graph for gdcmDataEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

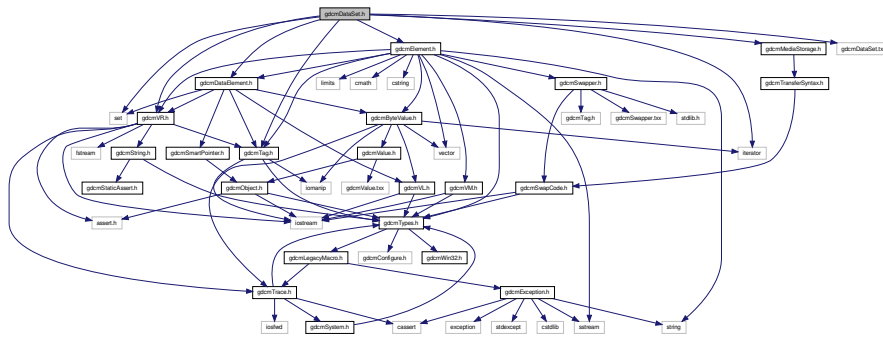
- class [gdcm::DataEvent](#)
DataEvent.

Namespaces

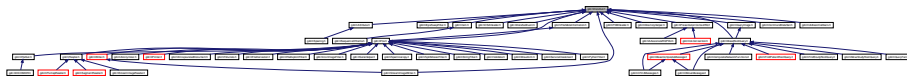
- [gdcm](#)

26.54 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.

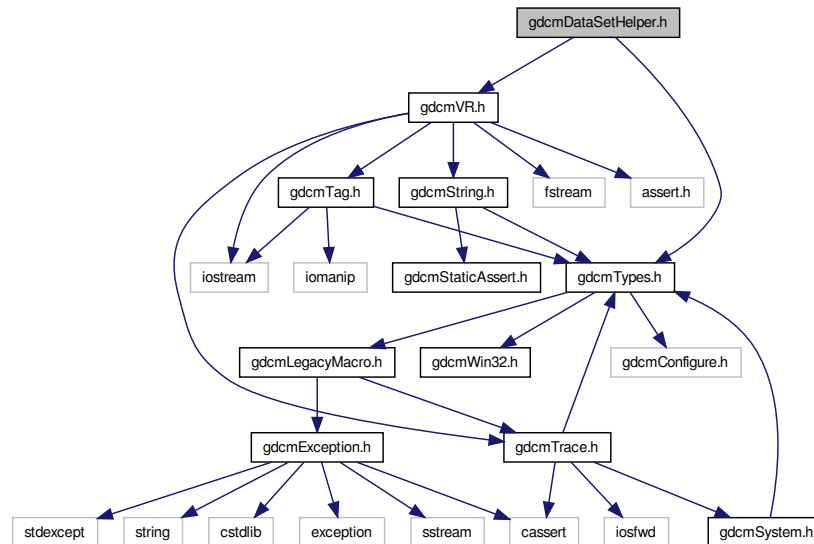
Namespaces

- **gdcm**

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const DataSet &val)`

Include dependency graph for gdcmDataSetHelper.h:



Classes

- class [gdcm::DataSetHelper](#)

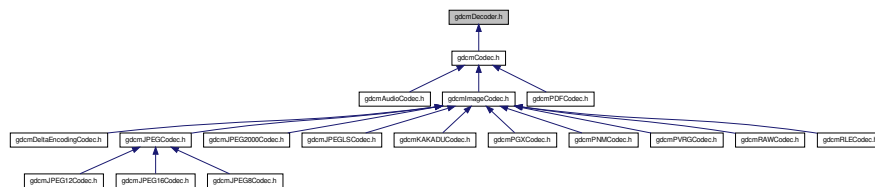
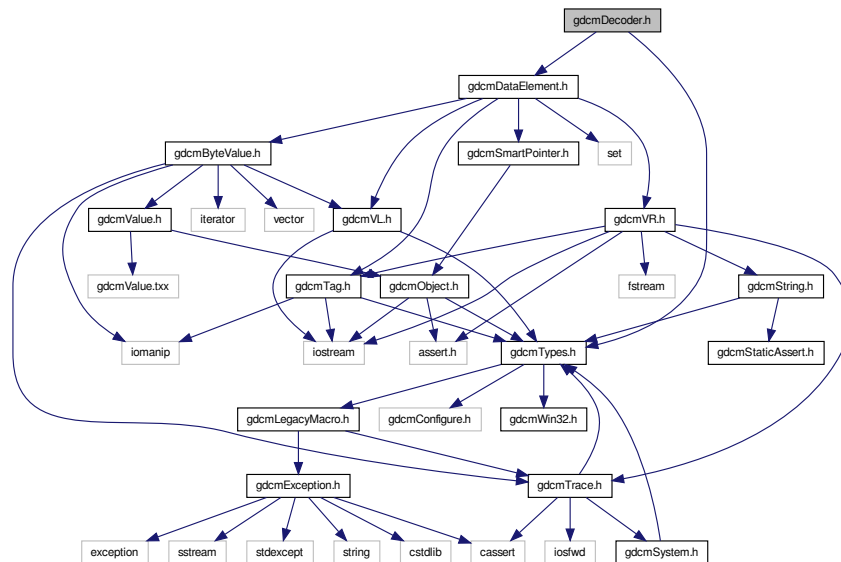
[DataSetHelper](#) (internal class, not intended for user level)

Namespaces

- [gdcm](#)

26.57 gdcmDecoder.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```

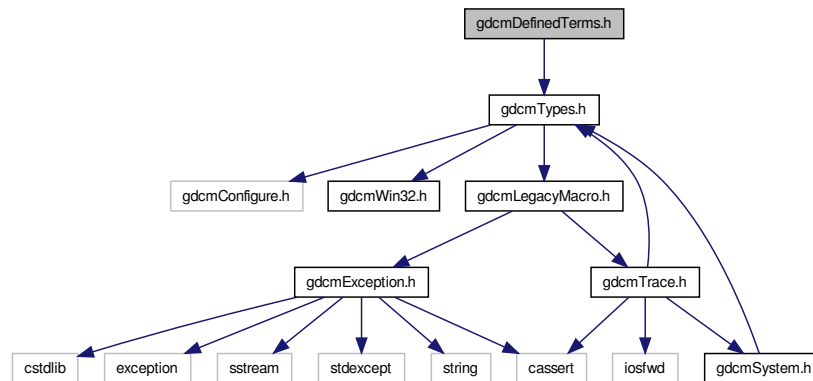


- class `gdcm::Decoder`
Decoder.

- **gdcm**

```
#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.

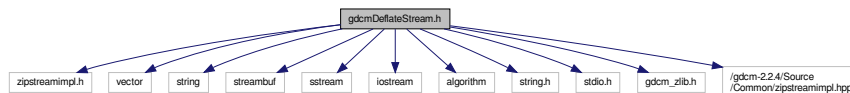
Namespaces

- [gdcm](#)

26.59 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

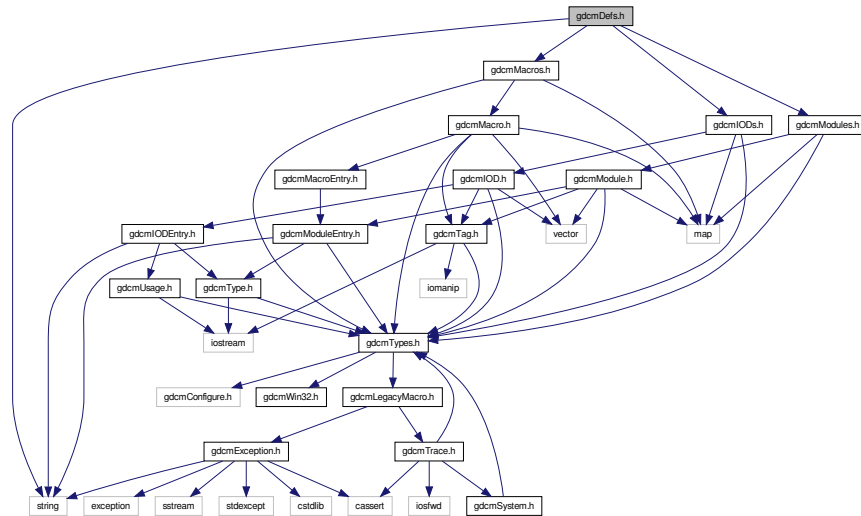
Include dependency graph for gdcmDeflateStream.h:



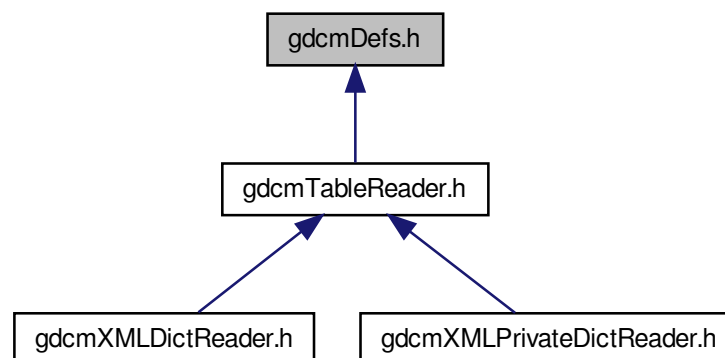
26.60 gdcmDefs.h File Reference

```
#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:

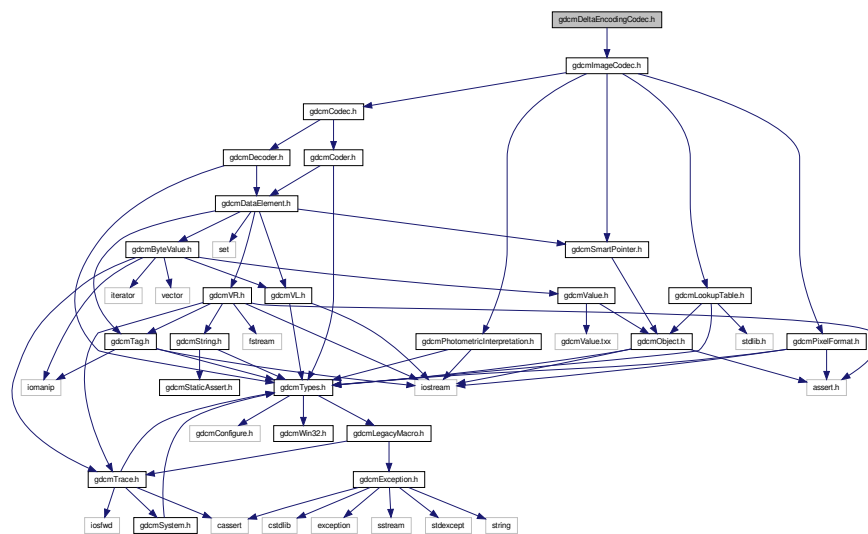


Classes

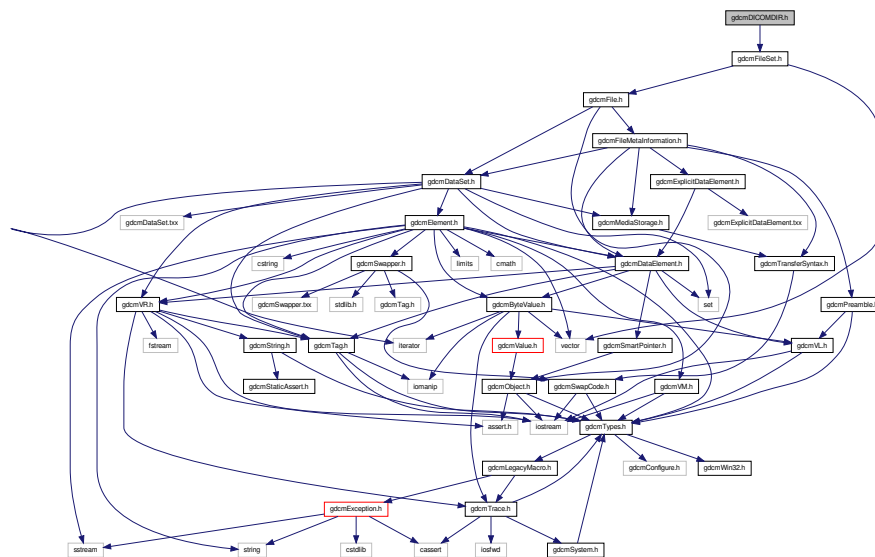
- class [gdcm::Defs](#)

FIXME I do not like the name 'Defs'.

- gdc



Include dependency graph for `gdcmDICOMDIR.h`:



Classes

- class `gdcm::DICOMDIR`

DICOMDIR class.

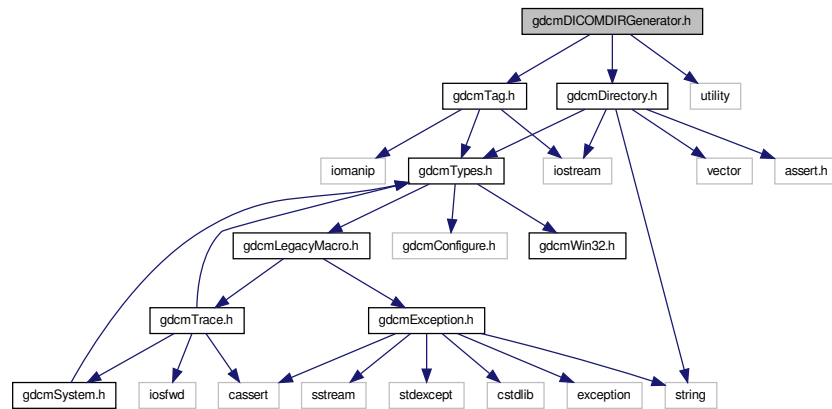
Namespaces

- `gdcm`

26.63 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](#)

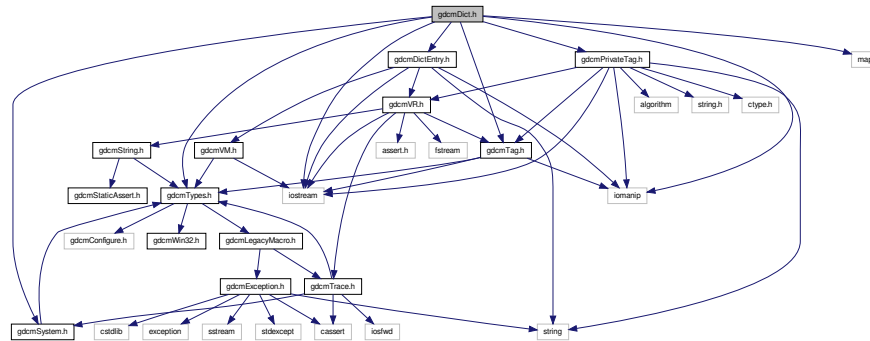
26.64 gdcmDict.h File Reference

```

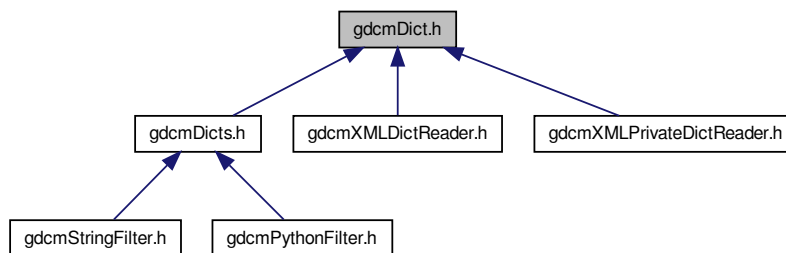
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmDictEntry.h"
#include "gdcmSystem.h"
#include <iostream>
#include <iomanip>
#include <map>

```

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`

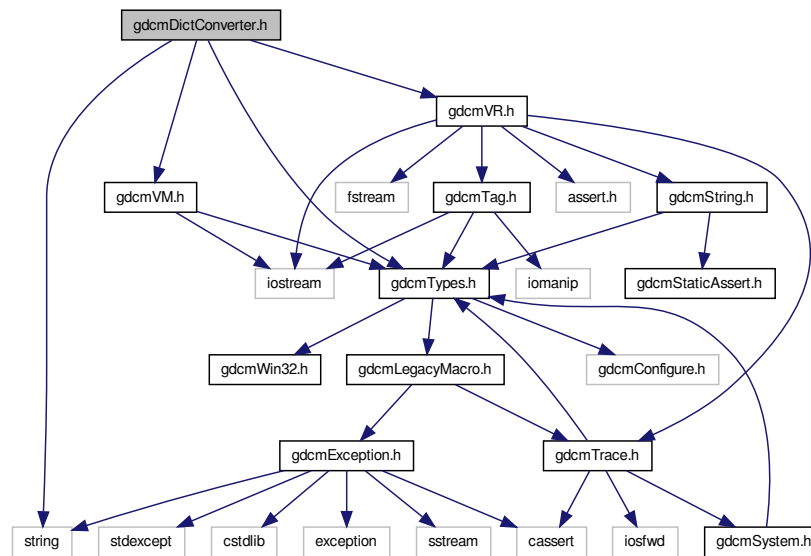
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateDict &val)`

26.65 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:

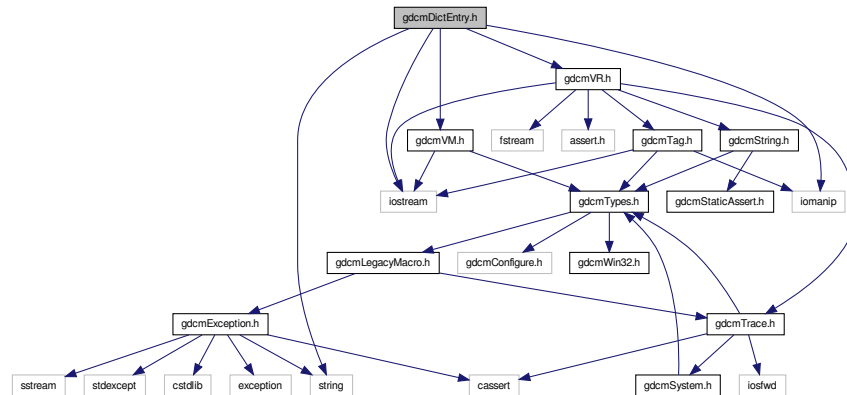
Namespaces

- [gdcm](#)

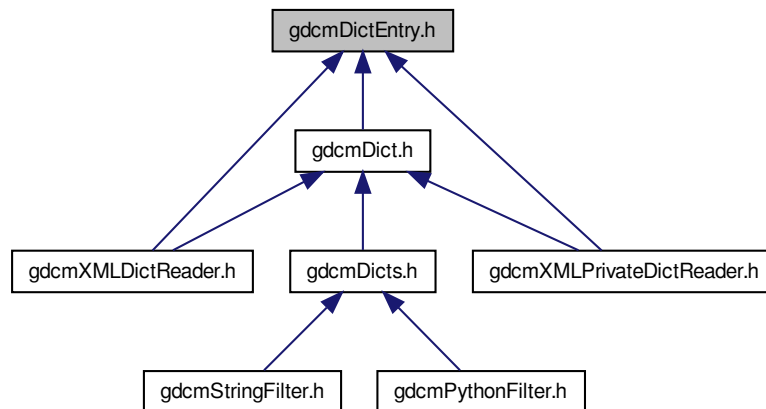
26.66 gdcmDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
```

Include dependency graph for `gdcDictEntry.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdc::DictEntry](#)

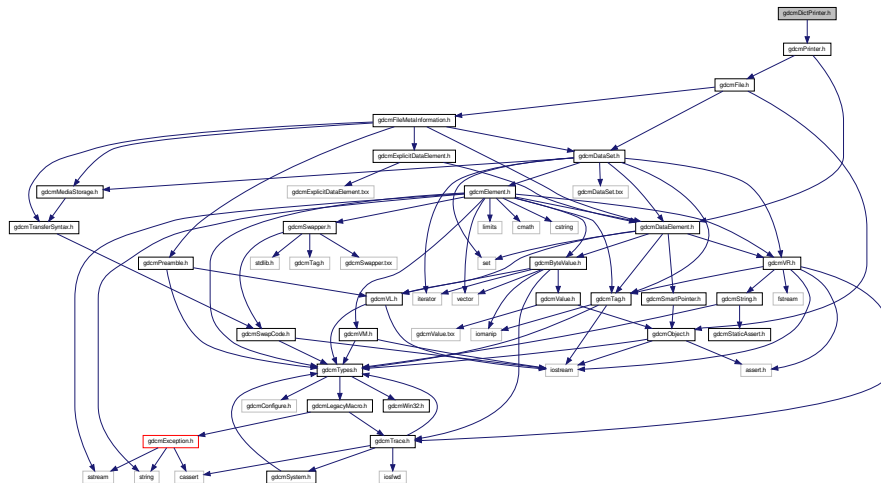
Class to represent an Entry in the [Dict](#) Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from [gdc::Tag](#) to the needed information.

Namespaces

- [gdc](#)

- `std::ostream & gdcmm::operator<< (std::ostream &os, const DictEntry &val)`

```
#include "gdcmPrinter.h"
Include dependency graph for gdcmDictPrinter.h:
```



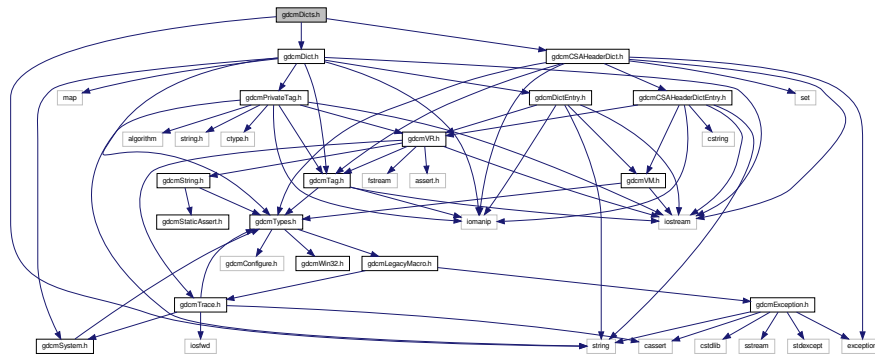
- class `gdcmm::DictPrinter`

DictPrinter class.

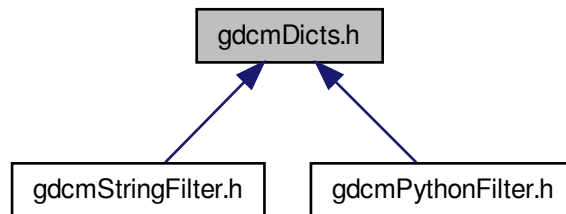
- gdc

```
#include "gdcDict.h"
#include "gdcCSAHeaderDict.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

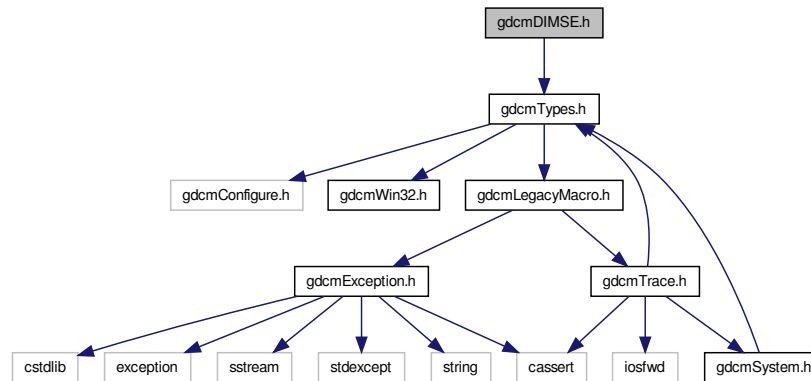
- `std::ostream & gdcm::operator<< (std::ostream &os, const Dicts &d)`

26.69 gdcmDiff.man File Reference

26.70 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 \(PART 1\)](#)*

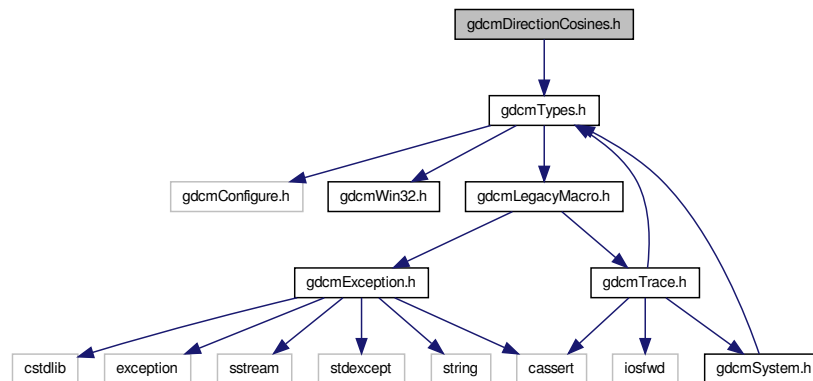
Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.71 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmDirectionCosines.h`:



Classes

- class `gdcm::DirectionCosines`
class to handle `DirectionCosines`

Namespaces

- `gdcm`

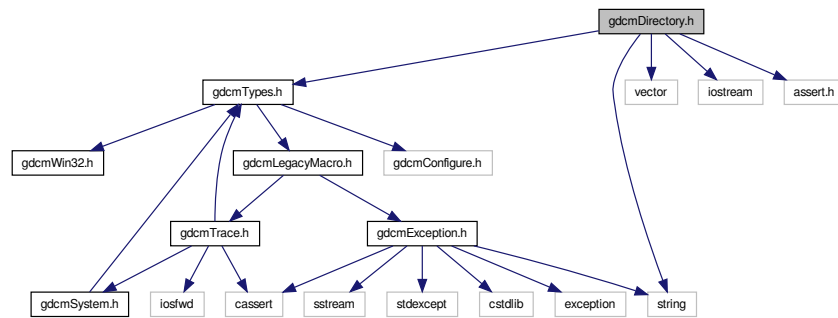
26.72 `gdcmDirectory.h` File Reference

```

#include "gdcmTypes.h"
#include <string>
#include <vector>
#include <iostream>
#include <assert.h>

```

Include dependency graph for `gdcmDirectory.h`:



```

classDiagram
    class gdcmsdkDirectory_h["gdcmsdk.gdcmsdkDirectory.h"]
    class dicomdirgenerator_h["gdcmsdk.DICOMDIRGenerator.h"]
    class directoryhelper_h["gdcmsdk.DirectoryHelper.h"]
    class sorter_h["gdcmsdk.Sorter.h"]
    class scanner_h["gdcmsdk.Scanner.h"]
    class compositenetworkfunctions_h["gdcmsdk.CompositeNetworkFunctions.h"]
    class presentationcontextgenerator_h["gdcmsdk.PresentationContextGenerator.h"]
    class ipsorter_h["gdcmsdk.IPPSorter.h"]

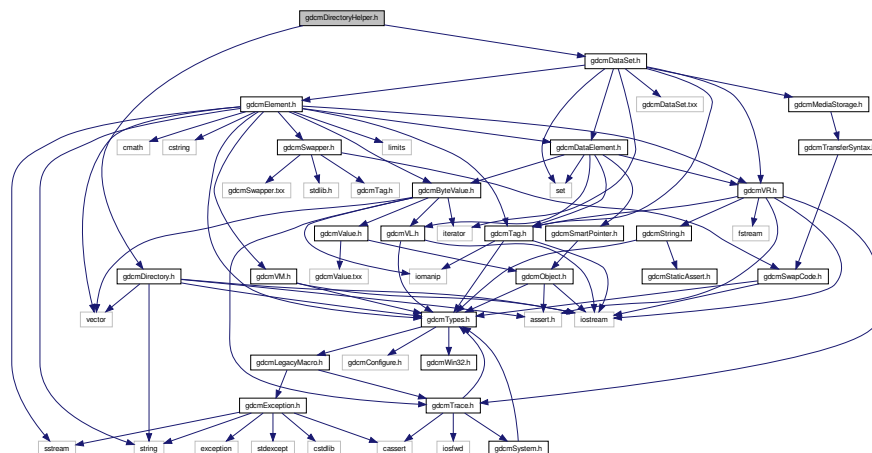
    gdcmsdkDirectory_h <|-- dicomdirgenerator_h
    gdcmsdkDirectory_h <|-- directoryhelper_h
    gdcmsdkDirectory_h <|-- sorter_h
    gdcmsdkDirectory_h <|-- scanner_h
    gdcmsdkDirectory_h <|-- compositenetworkfunctions_h
    gdcmsdkDirectory_h <|-- presentationcontextgenerator_h
    sorter_h <|-- ipsorter_h
  
```

- class `gdcm::Directory`
Class for manipulation directories.

- **gdcm**

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Directory &d)`

```
#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.

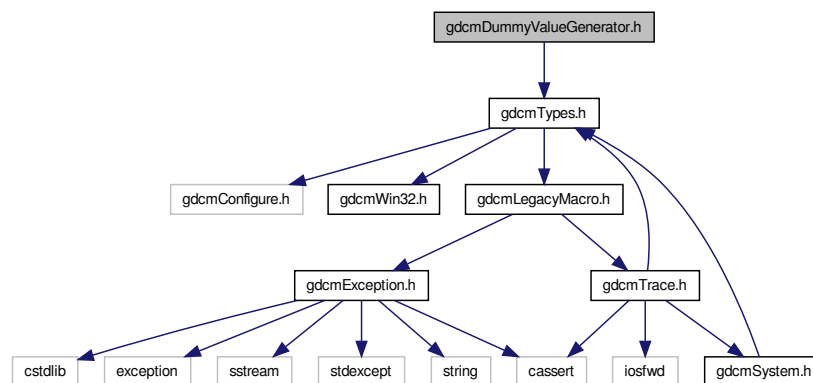
Namespaces

- [gdcm](#)

26.74 gdcmDummyValueGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDummyValueGenerator.h:



Classes

- class [gdcm::DummyValueGenerator](#)

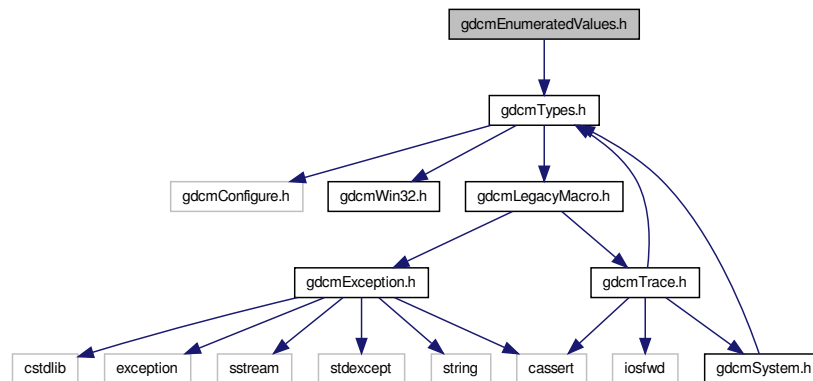
Class for generating dummy value.

Namespaces

- [gdcm](#)

26.75 gdcmdump.man File Reference

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:

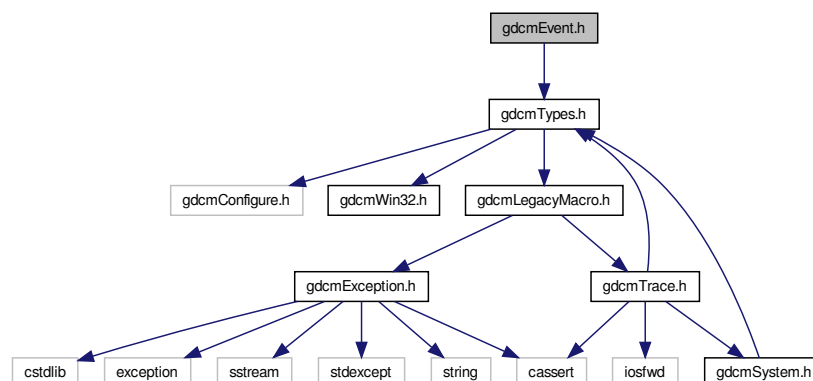
Namespaces

- [gdcm](#)

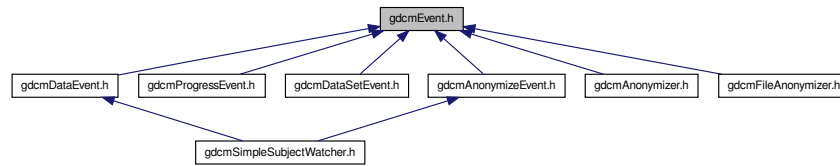
26.80 gdcmEvent.h File Reference

```
#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.

26.80.1 Macro Definition Documentation

26.80.1.1 `#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 ::gdc::Event* e) const \
    { return dynamic_cast<const Self*>(e) ? true : false; } \
    virtual ::gdc::Event* MakeObject() const \
    { return new Self; } \
    classname(const Self&s) : super(s){}; \
private: \
    void operator=(const Self&); \
}

```

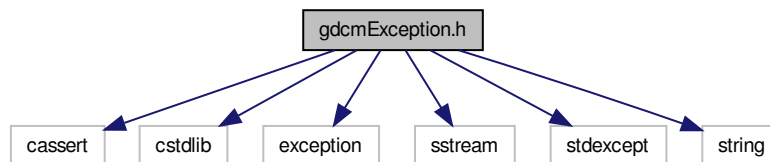
26.81 gdcException.h File Reference

```

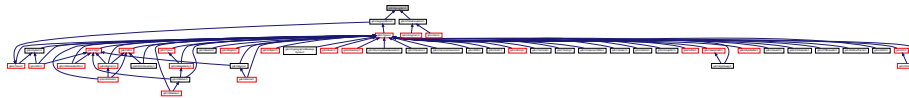
#include <cassert>
#include <cstdlib>
#include <exception>
#include <sstream>
#include <stdexcept>
#include <string>

```

Include dependency graph for gdcException.h:



This graph shows which files directly or indirectly include this file:



Classes

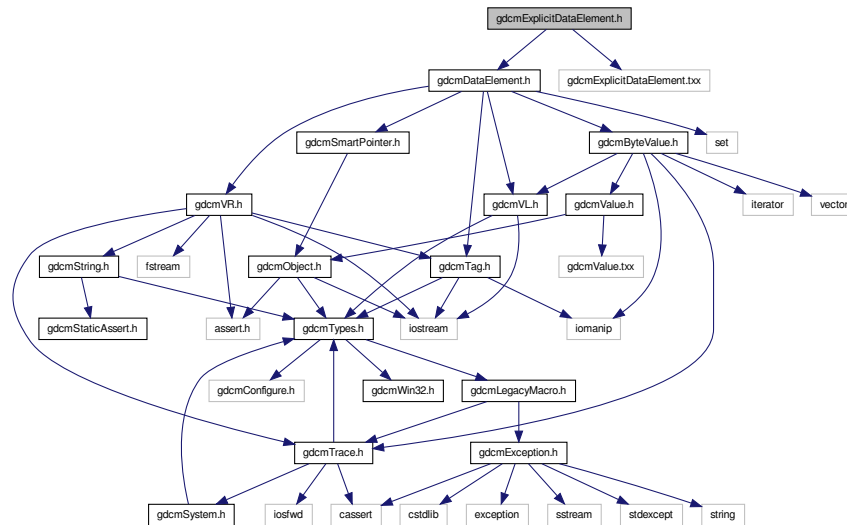
- class [gdc::Exception](#)
Exception.

Namespaces

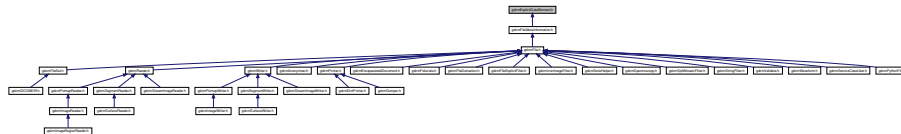
- [gdc](#)

26.82 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

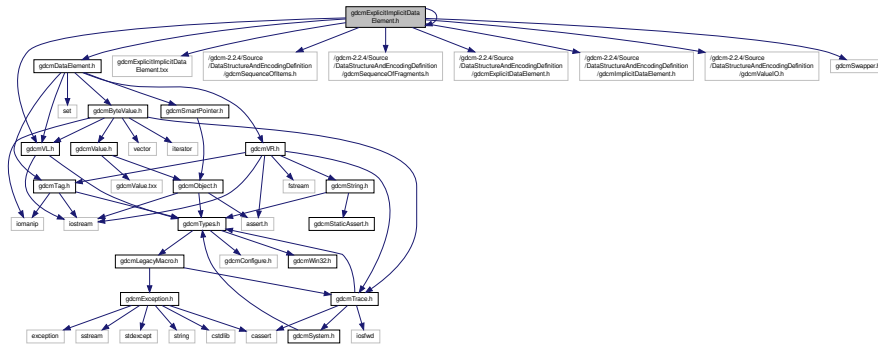
- class [gdcm::ExplicitDataElement](#)
Class to read/write a [DataElement](#) as *Explicit Data Element*.

Namespaces

- [gdcm](#)

26.83 gdcmExplicitImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmExplicitImplicitDataElement.txx"
```

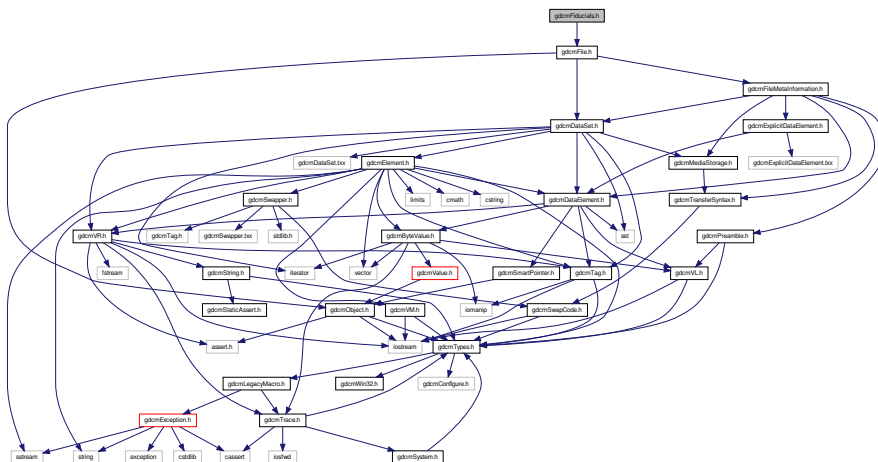


- class `gdcm::ExplicitImplicitDataElement`

Namespaces

- **gdcm**

```
#include "gdcmFile.h"
```



Classes

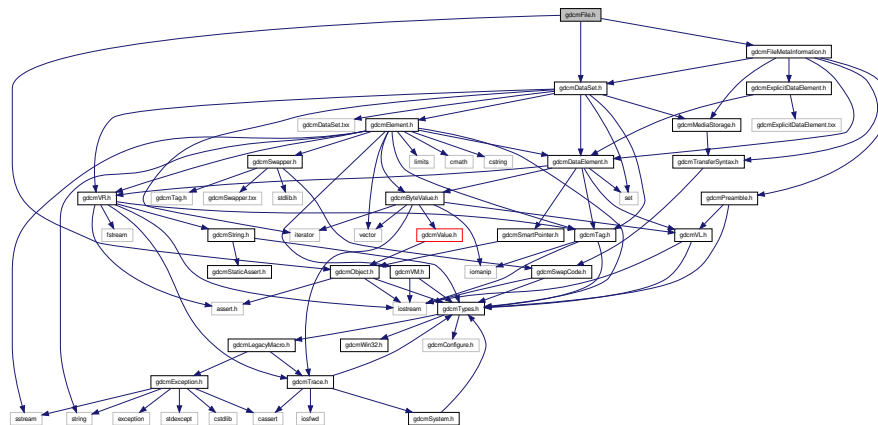
- class [gdcm::Fiducials](#)
Fiducials.

Namespaces

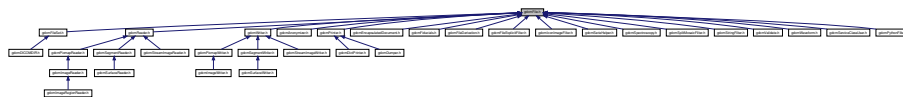
- [gdcm](#)

26.85 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 be written, read and/or deleted.

Namespaces

- [gdcm](#)

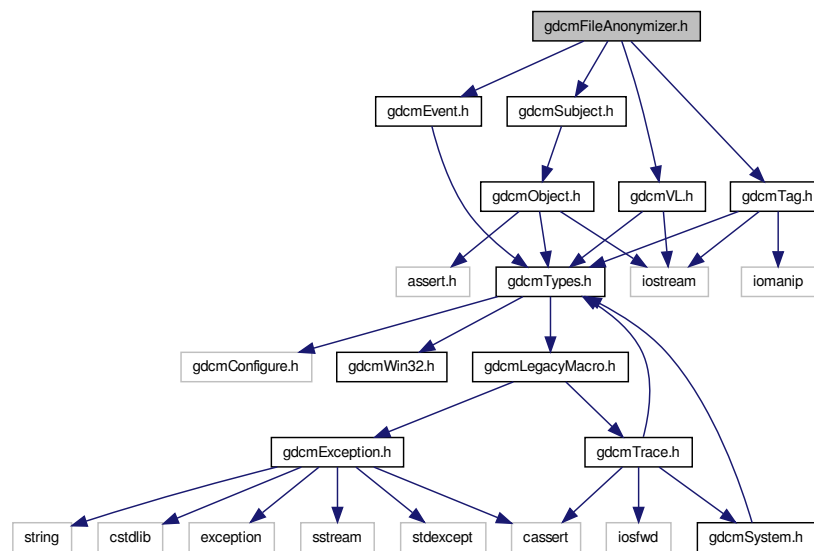
Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const File &val)`

26.86 gdcmFileAnonymizer.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmTag.h"
#include "gdcmVL.h"
```

Include dependency graph for `gdcmFileAnonymizer.h`:



Classes

- class `gdcmm::FileAnonymizer`
FileAnonymizer.

Namespaces

- `gdcmm`

26.87 gdcmFileDerivation.h File Reference

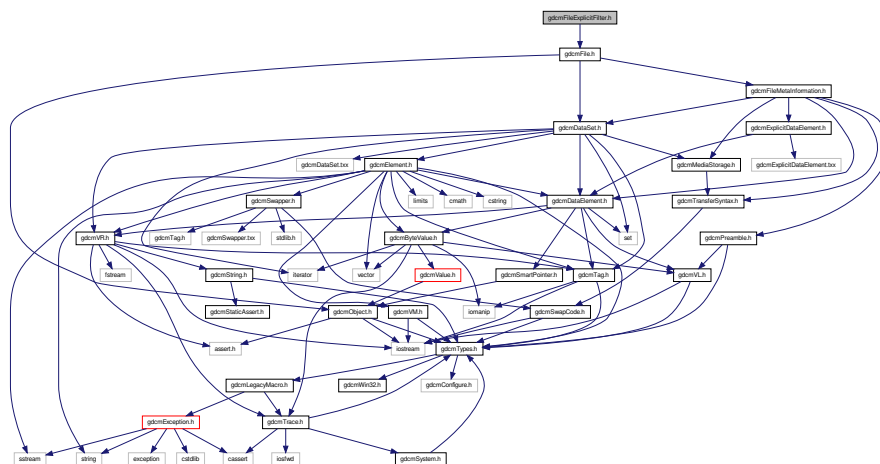
```
#include "gdcmFile.h"
```


- class `gdcm::FileDerivation`

Namespaces

- ## 26.88 gdcmFileExplicitFilter.h File Reference

Include dependency graph for `gdcmFileExplicitFilter.h`:



Namespaces

- [gdcm](#)

Functions

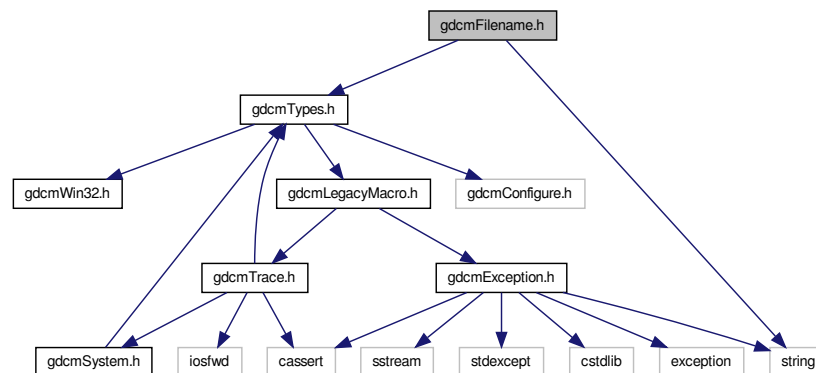
- `std::ostream & gdcm::operator<< (std::ostream &os, const FileMetaInformation &val)`

26.90 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](#)

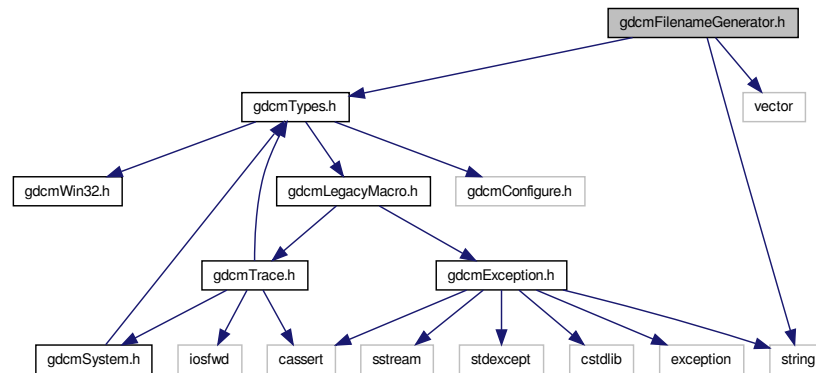
26.91 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <string>
```

```
#include <vector>
```

Include dependency graph for `gdcmFilenameGenerator.h`:



Classes

- class `gdcm::FilenameGenerator`

FilenameGenerator.

Namespaces

- `gdcm`

26.92 gdcmFileSet.h File Reference

```
#include "gdcmFile.h"
#include <vector>
```

```
graph BT; gdcmDICONDIR.h --> gdcmFileSet.h
```

- class `gdcm::FileSet`

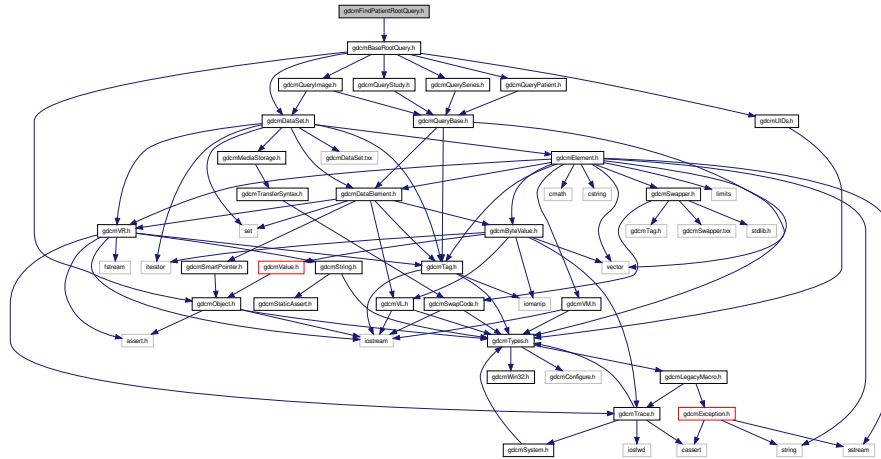
- **gdcm**

- `std::ostream & gdcmm::operator<< (std::ostream &os, const FileSet &f)`

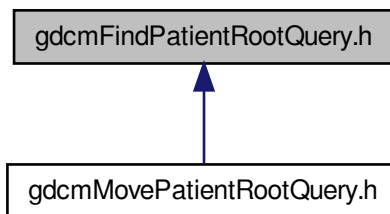
26.93 gdcmFindPatientRootQuery.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmFindPatientRootQuery.h:



This graph shows which files directly or indirectly include this file:



Classes

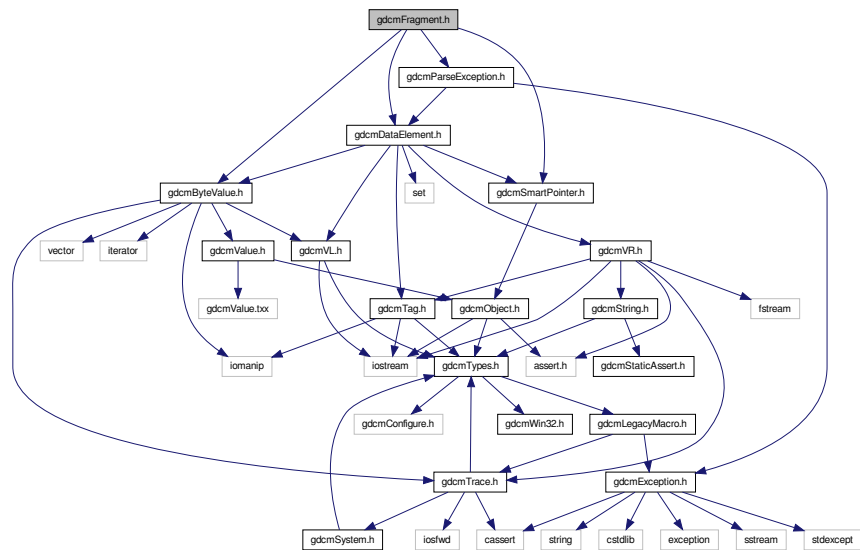
- class [gdcm::FindPatientRootQuery](#)

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

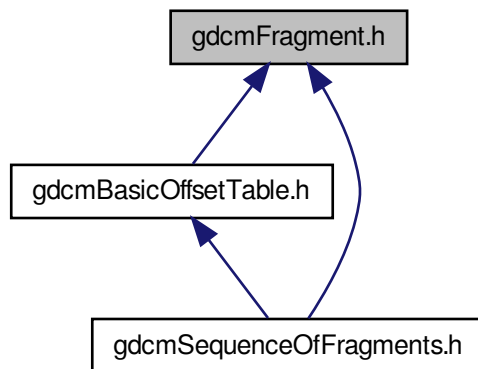
Namespaces

- [gdcm](#)

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

- [gdc](#)

Functions

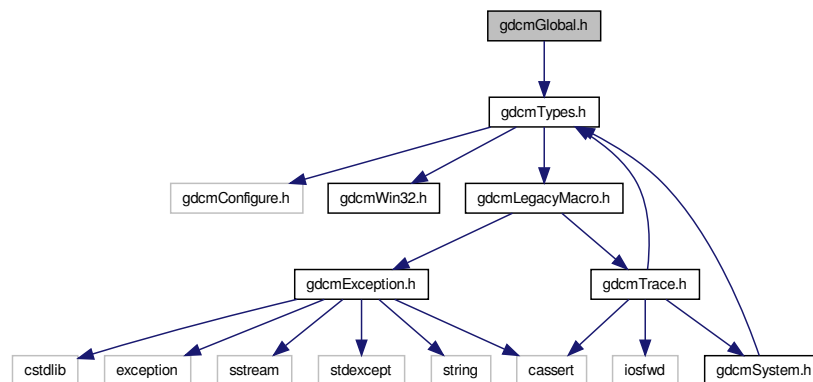
- `std::ostream & gdc::operator<< (std::ostream &os, const Fragment &val)`

26.96 gdcmgendir.man File Reference

26.97 gdcGlobal.h File Reference

```
#include "gdcTypes.h"
```

Include dependency graph for gdcGlobal.h:



Classes

- class [gdc::Global](#)
Global.

Namespaces

- [gdc](#)

Functions

- `std::ostream & gdc::operator<< (std::ostream &os, const Global &g)`

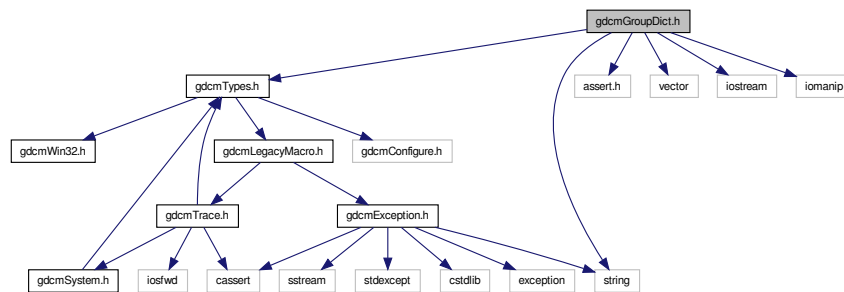
Variables

- static Global [gdcm::GlobalInstance](#)

26.98 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](#)

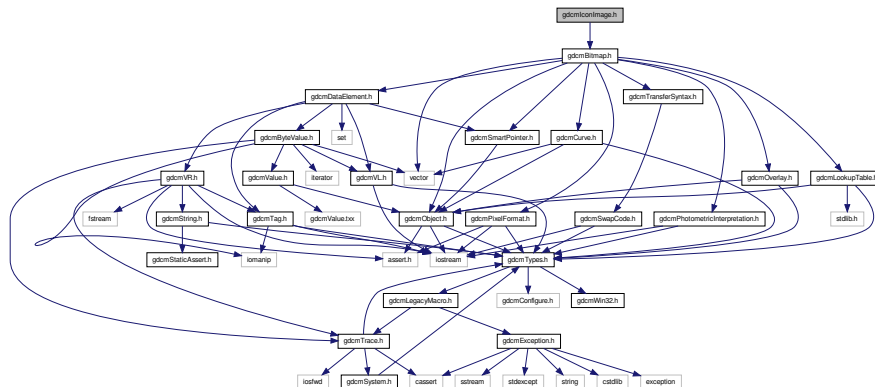
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)`

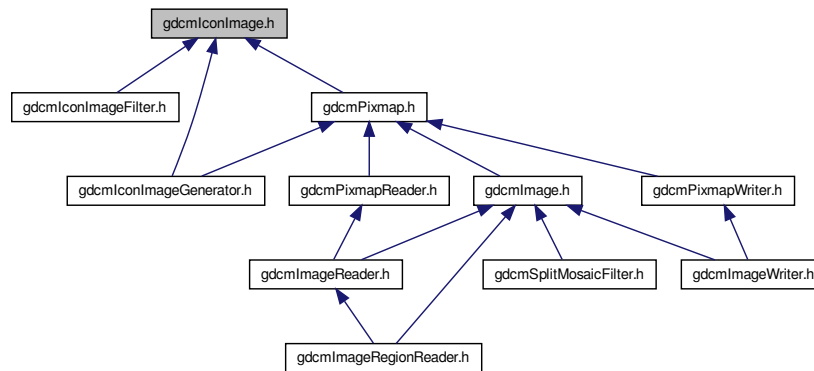
26.99 gdcmIconImage.h File Reference

```
#include "gdcmBitmap.h"
```

Include dependency graph for gdcmlconImage.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- **gdcm**

Typedefs

- typedef Bitmap **gdcm::IconImage**

26.100 gdcmlconImageFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```


[illegible]

- class `gdcm::IconImageGenerator`

`IconImageGenerator` This filter will generate a valid Icon from the Pixel Data element (an instance of `gdcm::Pixmap`). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

- IconImageGenerator** This filter will generate a valid Icon from the Pixel Data element (an instance of `gdcm::Pixmap`). To generate a valid Icon, one is only allowed the following Photometric Interpretation:*

```
#include "gdcmPixmap.h"
#include <vector>
```

Generated on Sat Dec 21 2013 01:40:25 for GDCM by Doxygen

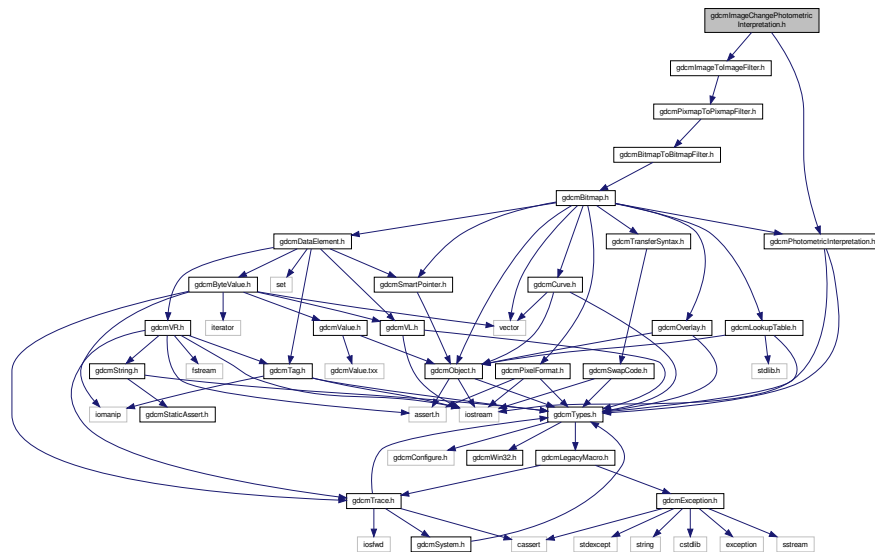
- class `gdcm::ImageApplyLookupTable`

aces

- gdc

```
#include "gdcImageToImageFilter.h"
#include "gdcPhotometricInterpretation.h"
```

Include dependency graph for `gdcmImageChangePhotometricInterpretation.h`:



Classes

- class `gdcm::ImageChangePhotometricInterpretation`

ImageChangePhotometricInterpretation class Class to change the Photometric Interpretation of an input DICOM.

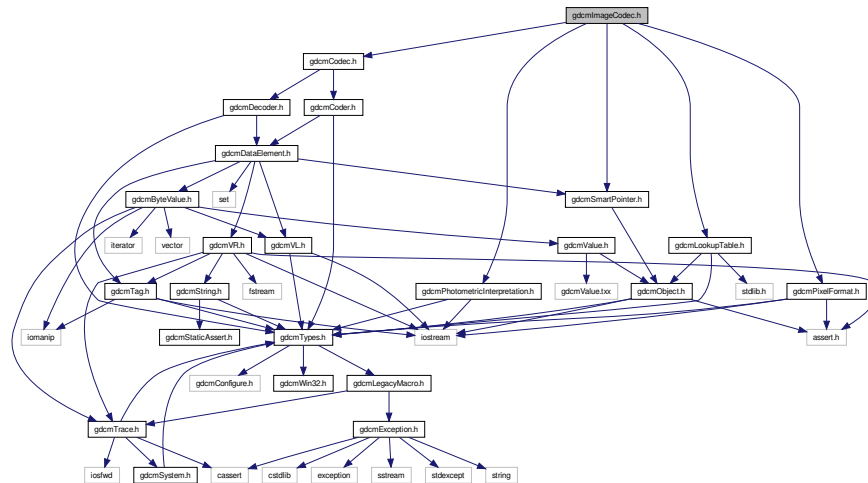
Namespaces

- `gdcm`

26.105 gdcmImageChangePlanarConfiguration.h File Reference

```
#include "gdcmImageToImageFilter.h"
```


Include dependency graph for gdcmImageCodec.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::ImageCodec](#)

ImageCodec.

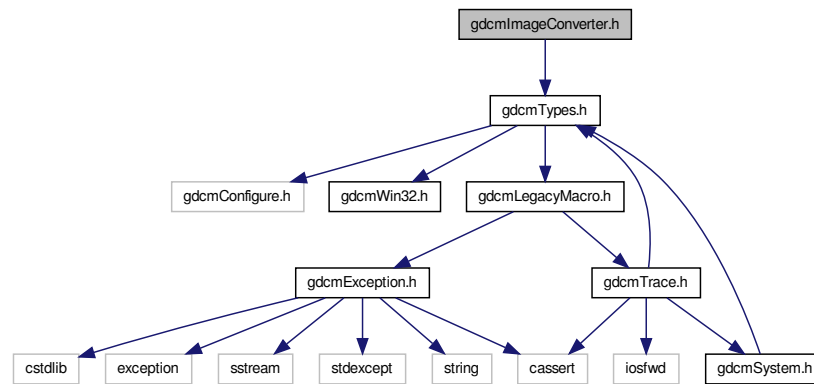
Namespaces

- [gdcm](#)

26.108 gdcmImageConverter.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmImageConverter.h`:



Classes

- class `gdcm::ImageConverter`

Image Converter.

Namespaces

- `gdcm`

26.109 gdcmImageFragmentSplitter.h File Reference

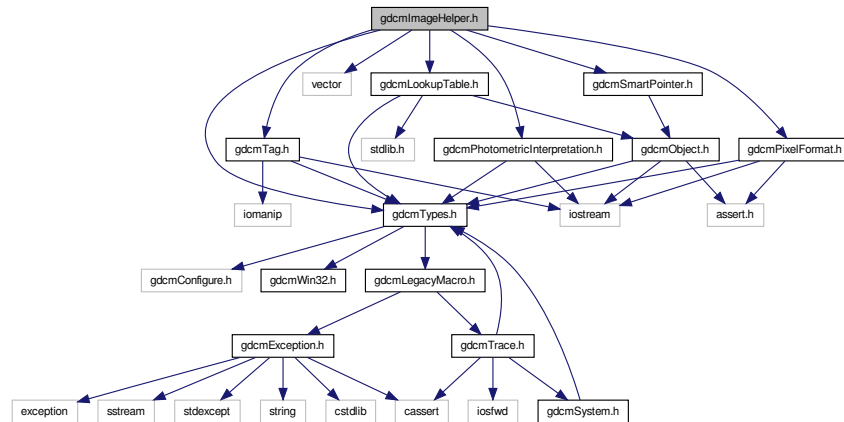
```
#include "gdcmImageToImageFilter.h"
```

- class `gdcm::ImageFragmentSplitter`

Namespaces

- **gdcm**

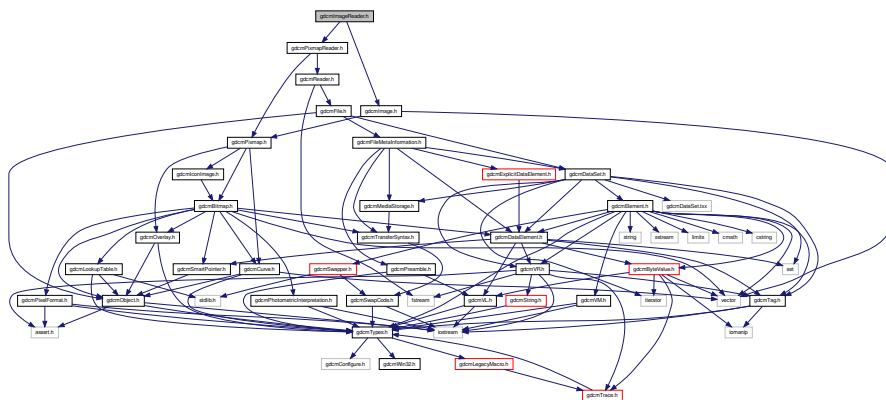
```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include <vector>
#include "gdcmPixelFormat.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmSmartPointer.h"
#include "gdcmLookupTable.h"
```



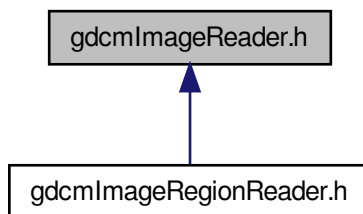
- class `gdcm::ImageHelper`
ImageHelper (internal class, not intended for user level)

- **gdcm**

```
#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*](#).

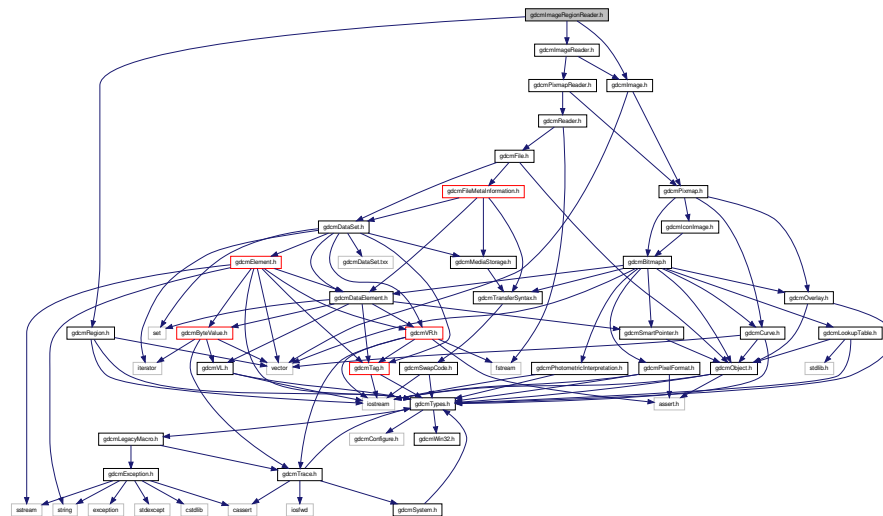
Namespaces

- [gdcm](#)

26.112 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`

26.113 gdcmImageToImageFilter.h File Reference

```
#include "gdcmPixmapToPixmapFilter.h"
```



```

classDiagram
    class gdcmImageApplyLookupTable
    class gdcmImageChangePhotometricInterpretation
    class gdcmImageChangePlaneConfiguration
    class gdcmImageChangeTransferSyntax
    class gdcmImageFragmentSplitter
    class gdcmImageToImageFilter

    gdcmImageApplyLookupTable --|> gdcmImageToImageFilter
    gdcmImageChangePhotometricInterpretation --|> gdcmImageToImageFilter
    gdcmImageChangePlaneConfiguration --|> gdcmImageToImageFilter
    gdcmImageChangeTransferSyntax --|> gdcmImageToImageFilter
    gdcmImageFragmentSplitter --|> gdcmImageToImageFilter
  
```

- class `gdcm::ImageToImageFilter`

Namespaces

- **gdcm**

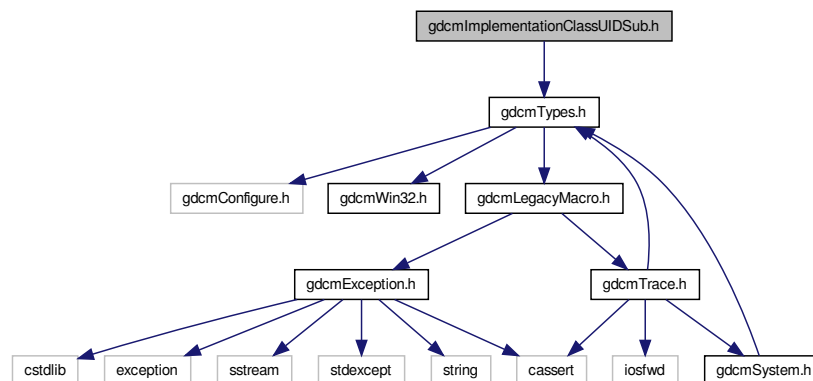
```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
```

- class `gdcm::ImageWriter`
ImageWriter.

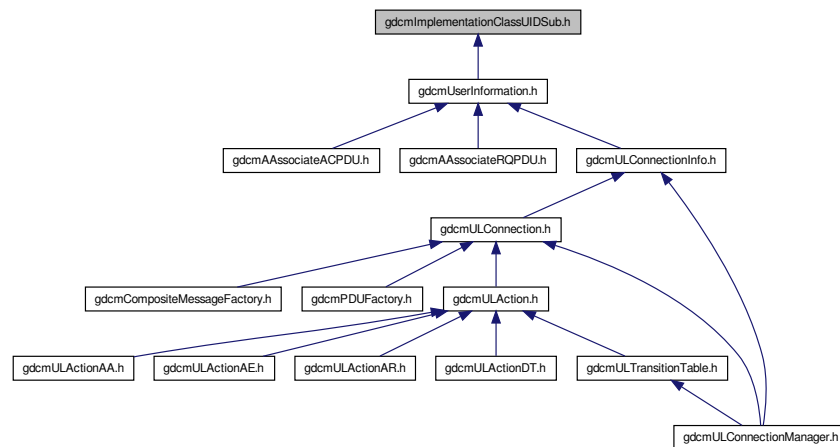
- **gdcm**

26.116 gdcmImplementationClassUIDSub.h File Reference

Include dependency graph for gdcmlImplementationClassUIDSub.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcml::network::ImplementationClassUIDSub](#)

[ImplementationClassUIDSub](#) PS 3.7 [Table D.3-1](#) IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE--RQ)

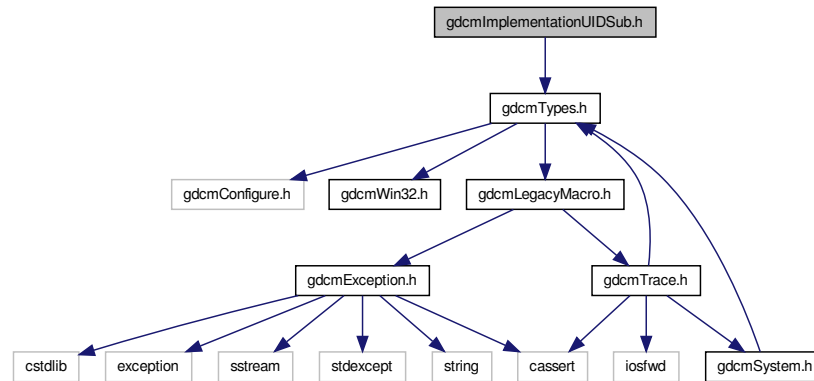
Namespaces

- [gdcml](#)
- [gdcml::network](#)

26.117 gdcmlImplementationUIDSub.h File Reference

```
#include "gdcmlTypes.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)

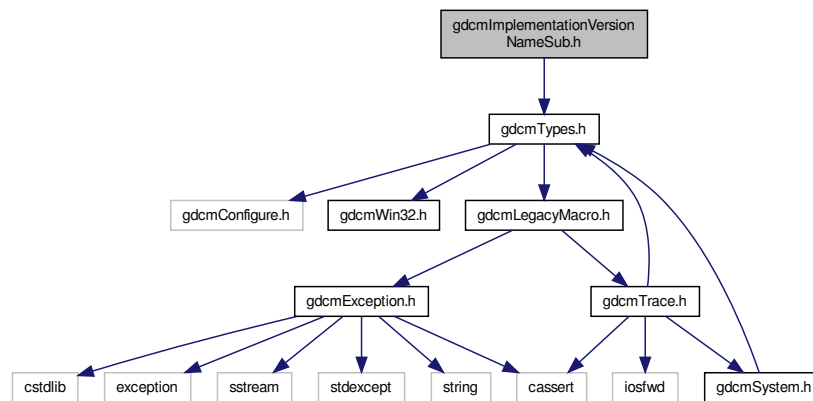
Namespaces

- `gdcm`
- `gdcm::network`

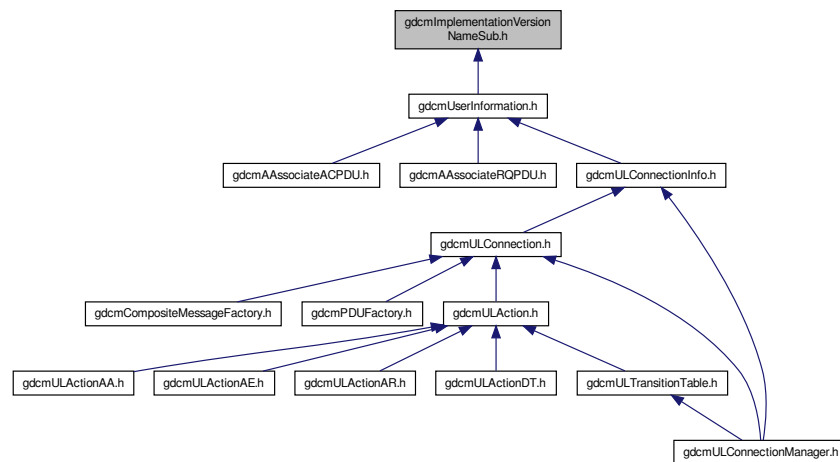
26.118 gdcmImplementationVersionNameSub.h File Reference

```
#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](#)

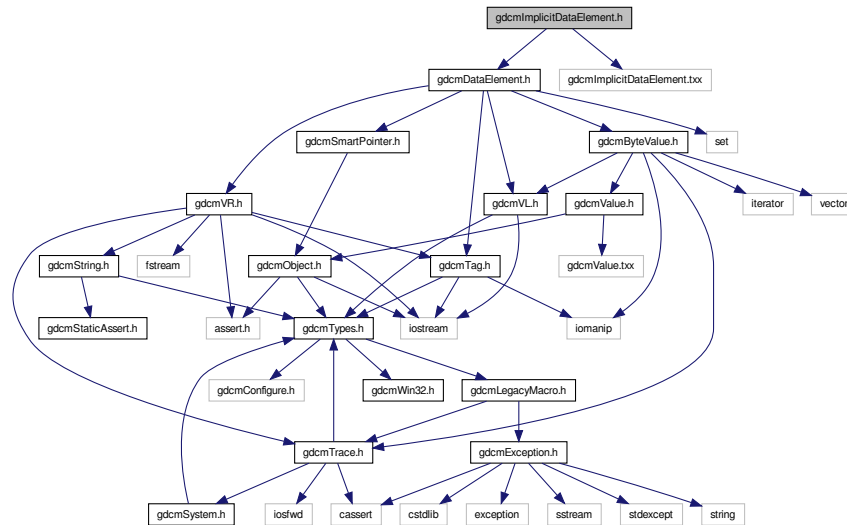
ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.119 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

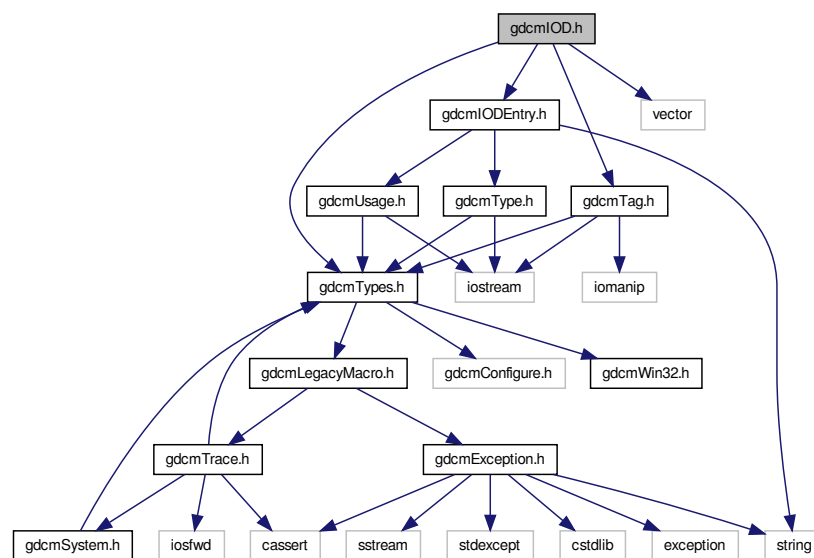
- [gdcm](#)

26.120 gdcminfo.man File Reference

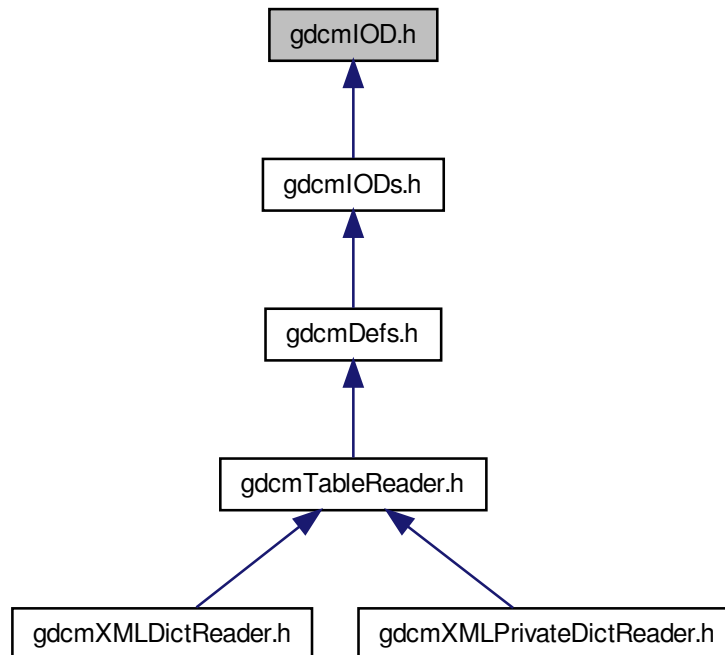
26.121 gdcmIOD.h File Reference

```
#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 [gdcml::IOD](#)
Class for representing a [IOD](#).

Namespaces

- [gdcml](#)

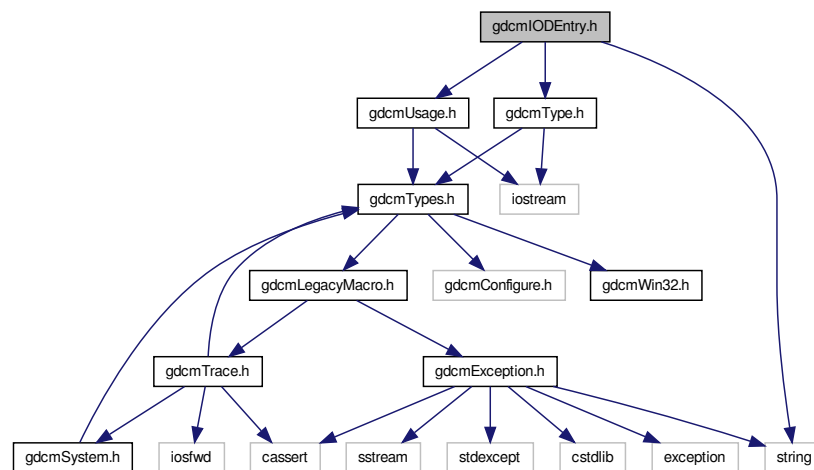
Functions

- `std::ostream & gdcml::operator<< (std::ostream &_os, const IOD &_val)`

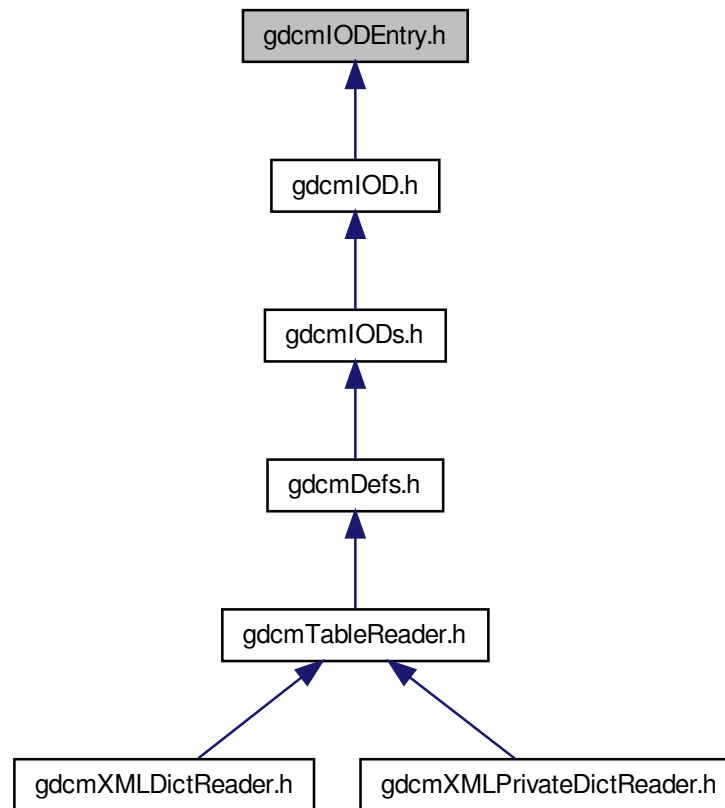
26.122 gdcmlIOEntry.h File Reference

```
#include "gdcmlUsage.h"  
#include "gdcmlType.h"  
#include <string>
```


Include dependency graph for gdcmIODEntry.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcml::IODEntry`
Class for representing a `IODEntry`.

Namespaces

- `gdcml`

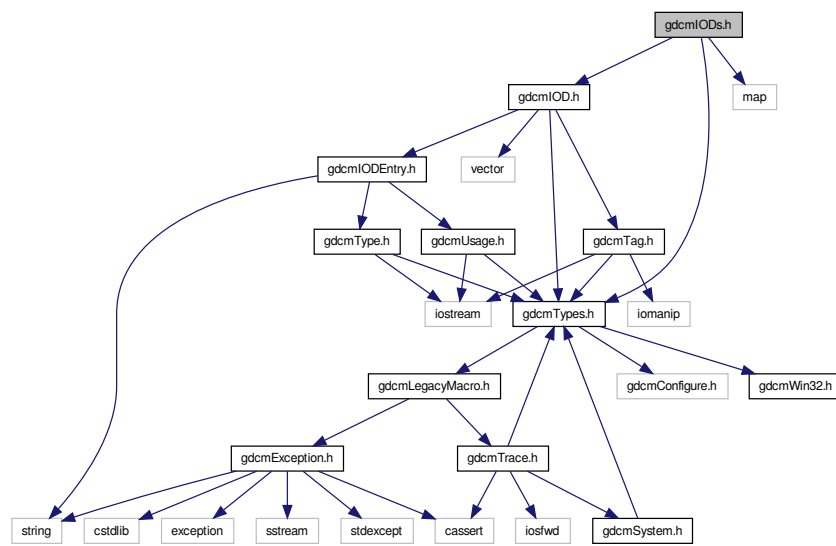
Functions

- `std::ostream & gdcml::operator<< (std::ostream &_os, const IODEntry &_val)`

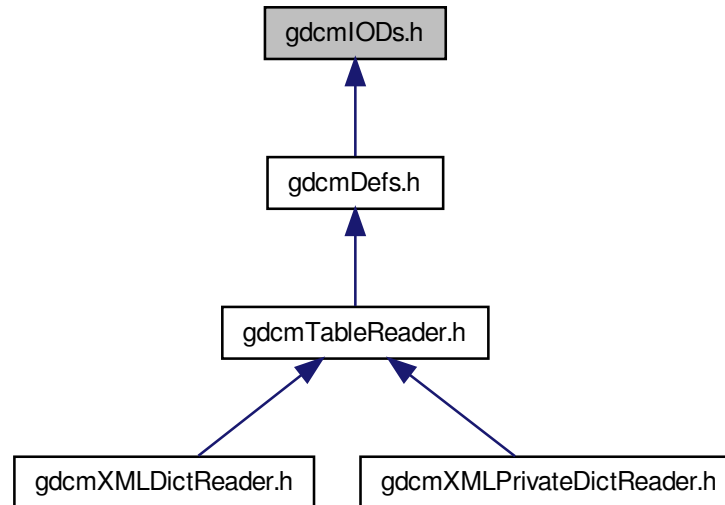
26.123 gdcmIODs.h File Reference

```
#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 [gdcmlODs](#)

Class for representing a [IODs](#).

Namespaces

- [gdcml](#)

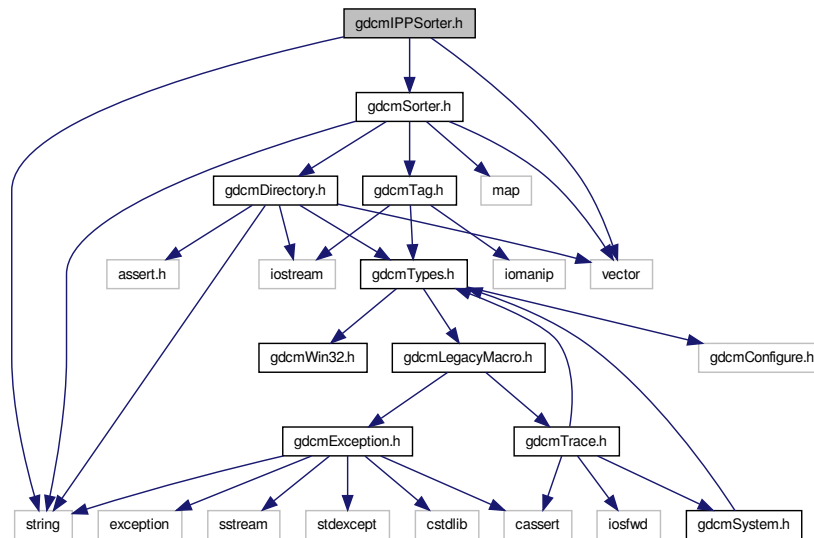
Functions

- `std::ostream & gdcml::operator<< (std::ostream &_os, const IODs &_val)`

26.124 gdcmlPPSorter.h File Reference

```
#include "gdcmlSorter.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

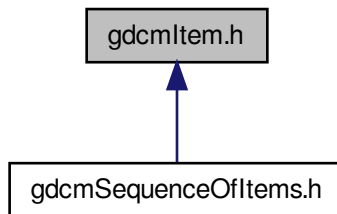
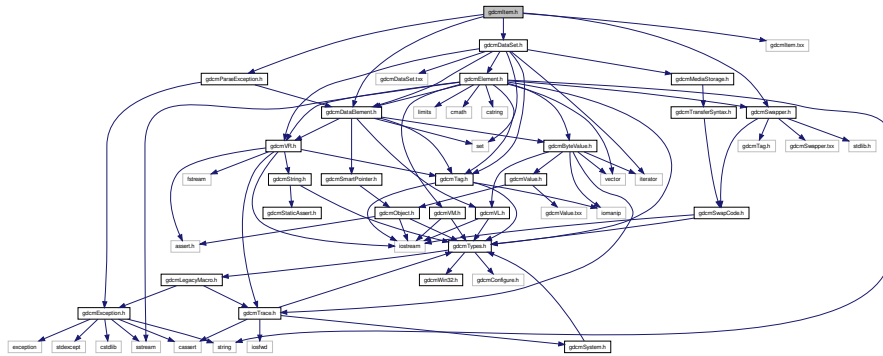
- [gdcm](#)

26.125 gdcmItem.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"
#include "gdcmItem.txx"

```



Class to represent an **Item** A component of the value of a Data **Element** that is of **Value** Representation Sequence of Items. An **Item** contains a Data Set . See PS 3.5 7.5.1 **Item** Encoding Rules Each **Item** of a Data **Element** of **VR** SQ shall be encoded as a DICOM Standard Data **Element** with a specific Data **Element** Tag of **Value** (FFFF,E000). The **Item** Tag is followed by a 4 byte **Item** Length field encoded in one of the following two ways Explicit/ Implicit.

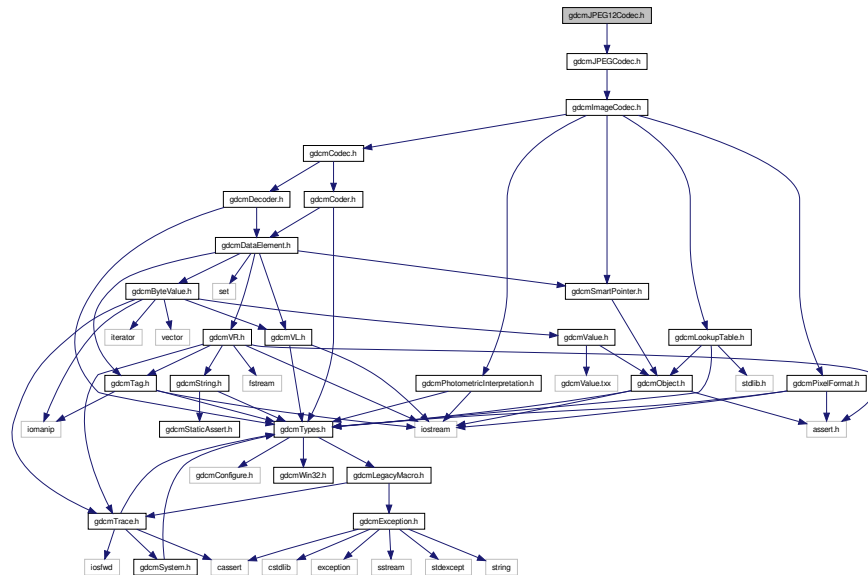
- ③

- `std::ostream & gdcm::operator<< (std::ostream &os, const Item &val)`

26.126 gdcmJPEG12Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG12Codec.h:



Classes

- class [gdcm::JPEG12Codec](#)

Class to do JPEG 12bits (lossy & lossless)

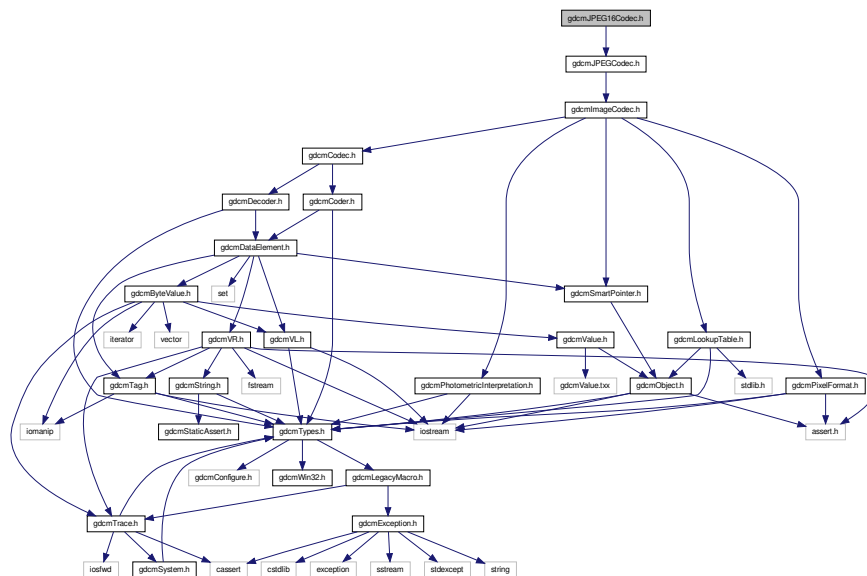
Namespaces

- [gdcm](#)

26.127 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](#)

26.128 gdcmJPEG2000Codec.h File Reference

```
#include "gdcmImageCodec.h"
```

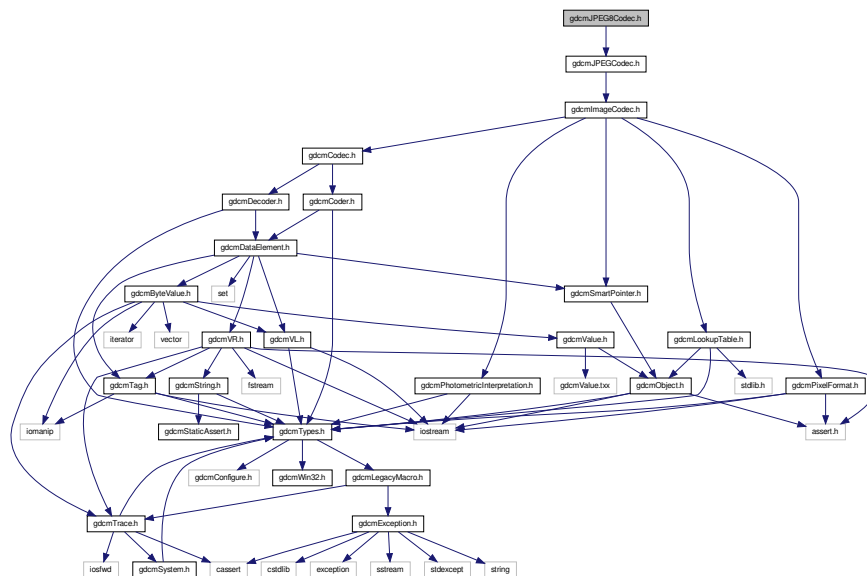

- class `gdcm::JPEG2000Codec`

Namespaces

- gdc

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for `gdcmJPEG8Codec.h`:



Classes

- class [gdcm::JPEG8Codec](#)

Class to do JPEG 8bits (lossy & lossless)

Namespaces

- [gdcm](#)

26.130 gdcmJPEGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

[illegible]

```
graph BT; gdcMJ12[gdcMJPEG12Codec.h] --> gdcMJ[gdcMJPEGCodec.h]; gdcMJ16[gdcMJPEG16Codec.h] --> gdcMJ; gdcMJ8[gdcMJPEG8Codec.h] --> gdcMJ;
```

- class `gdcm::JPEGCodec`

Namespaces

- [gdcm](#)

```
#include "gdcmImageCodec.h"
```

[illegible]

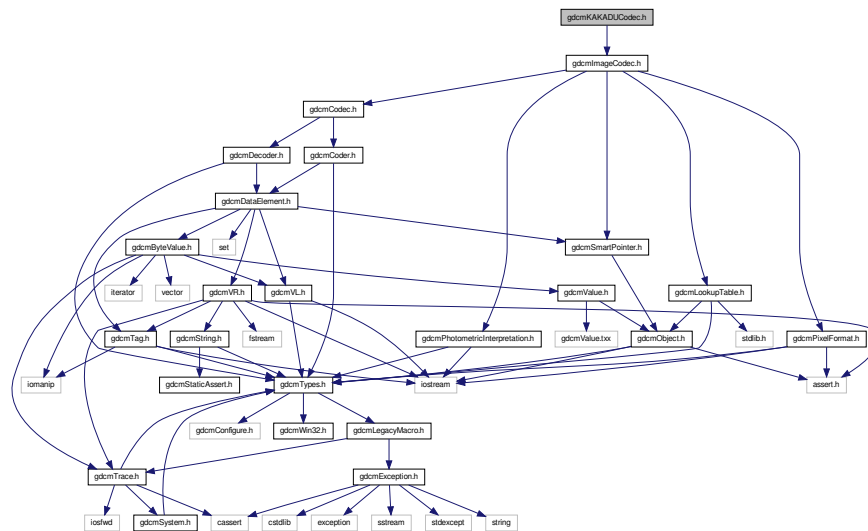
- class `gdcm::JPEGLSCodec`

Namespaces

- **gdcm**

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmKAKADUCodec.h:



Classes

- class [gdcm::KAKADUCodec](#)

KAKADUCodec.

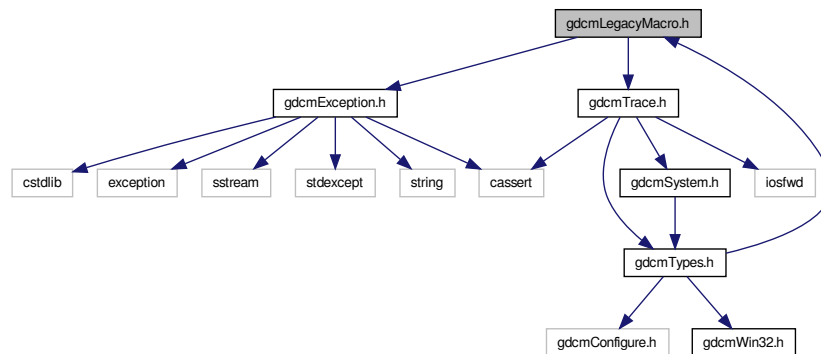
Namespaces

- [gdcm](#)

26.133 gdcmLegacyMacro.h File Reference

```
#include "gdcmException.h"
#include "gdcmTrace.h"
```

Include dependency graph for `gdcmlLegacyMacro.h`:



This graph shows which files directly or indirectly include this file:



Macros

- `#define GDCM_LEGACY(method) method;`
- `#define GDCM_LEGACY_BODY(method, version) gdcmlWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")`
- `#define GDCM_LEGACY_REPLACED_BODY(method, version, replace) gdcmlWarningMacro(#method " was deprecated for " version " and will be removed in a future version. Use " #replace " instead.")`

26.133.1 Macro Definition Documentation

26.133.1.1 `#define GDCM_LEGACY(method) method;`

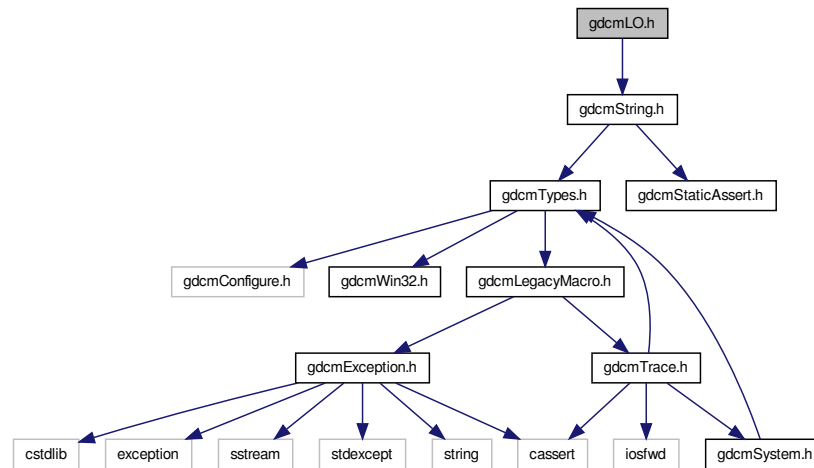
26.133.1.2 `#define GDCM_LEGACY_BODY(method, version) gdcmlWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")`

26.133.1.3 `#define GDCM_LEGACY_REPLACED_BODY(method, version, replace) gdcmlWarningMacro(#method " was deprecated for " version " and will be removed in a future version. Use " #replace " instead.")`

26.134 gdcmlLO.h File Reference

```
#include "gdcmlString.h"
```

Include dependency graph for gdcmLO.h:



Classes

- class `gdcm::LO`

LO.

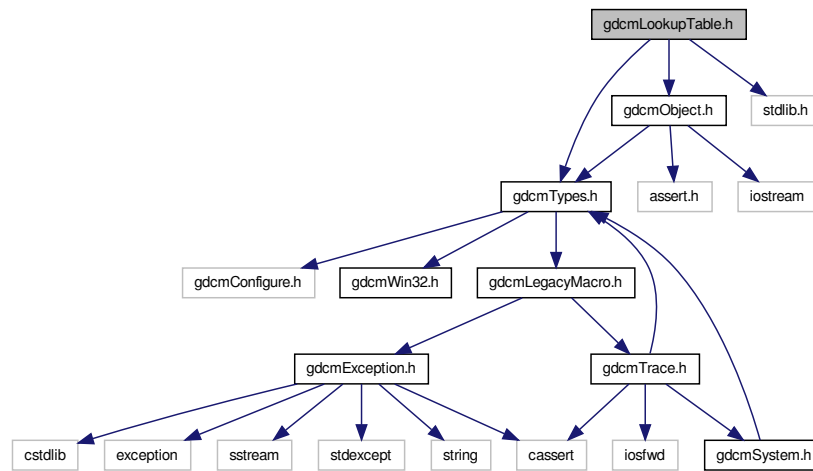
Namespaces

- `gdcm`

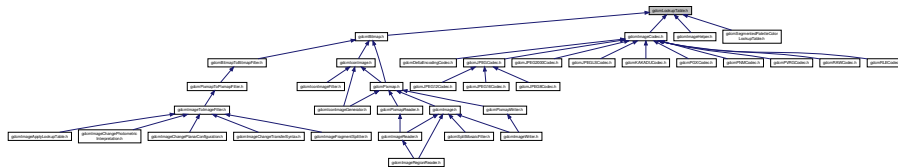
26.135 gdcmLookupTable.h File Reference

```
#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](#)

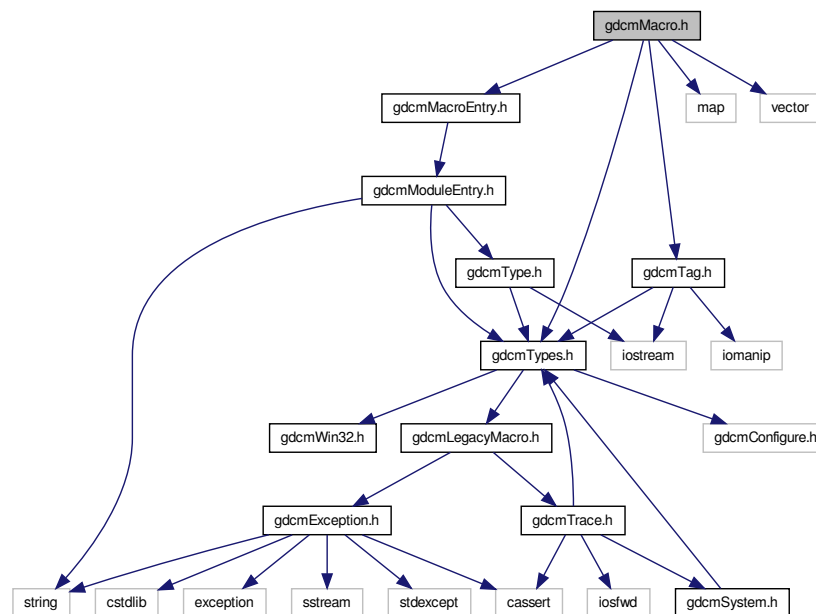
26.136 gdcmMacro.h File Reference

```

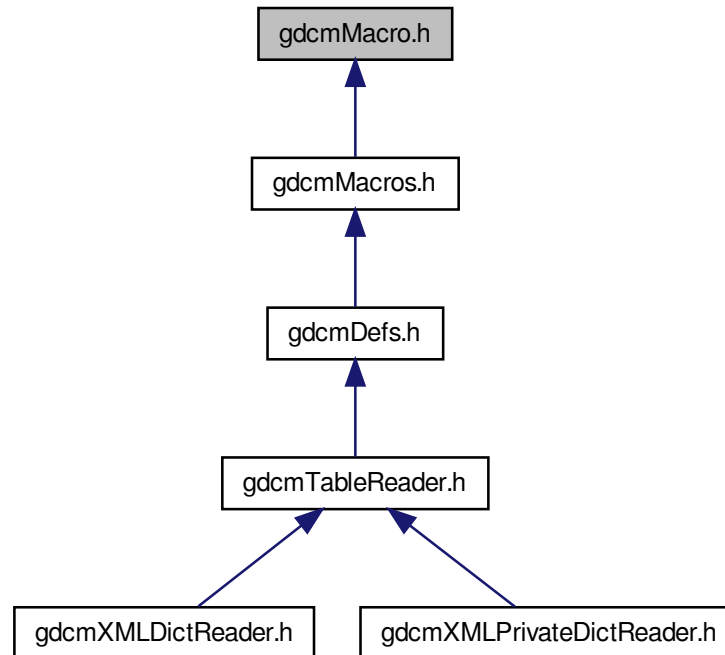
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmMacroEntry.h"
#include <map>
#include <vector>

```


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](#)

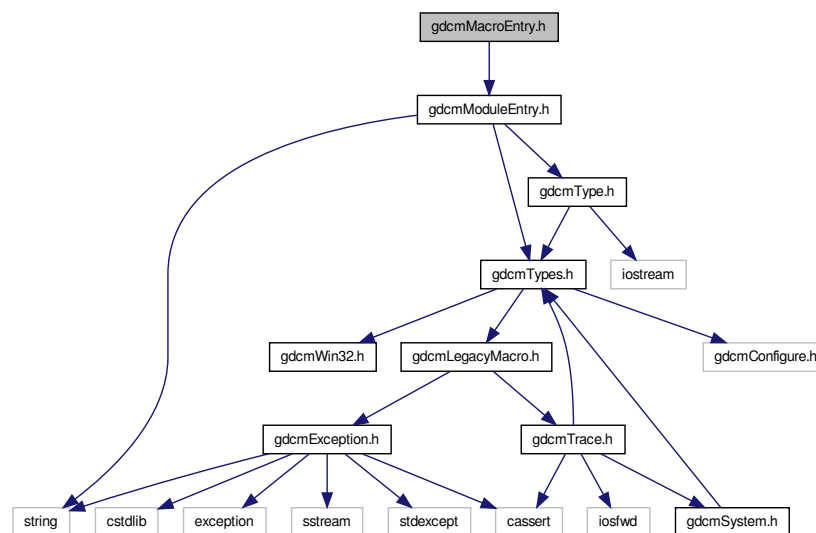
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)`

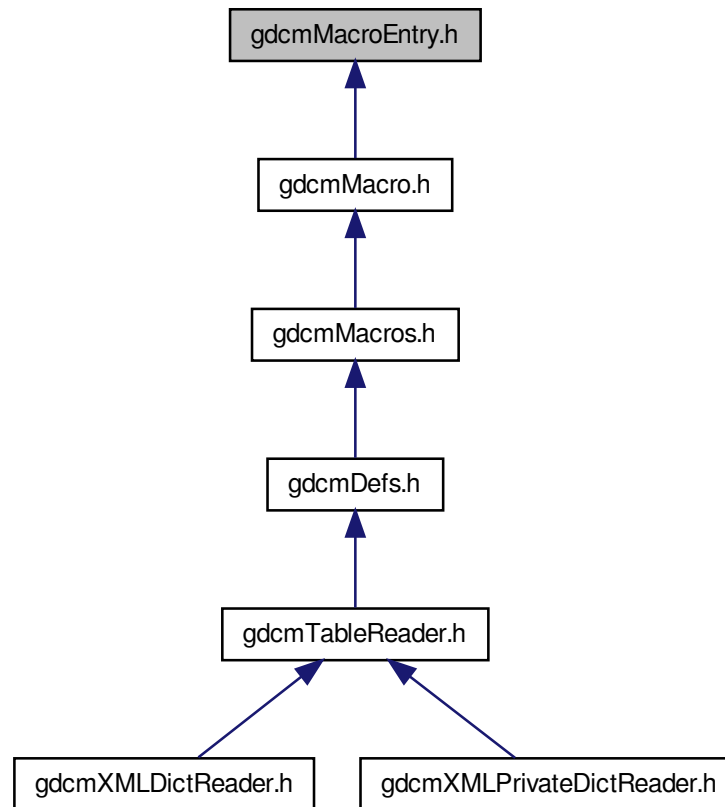
26.137 gdcMacroEntry.h File Reference

```
#include "gdcModuleEntry.h"
```

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define GDCMMACROENTRY_H`

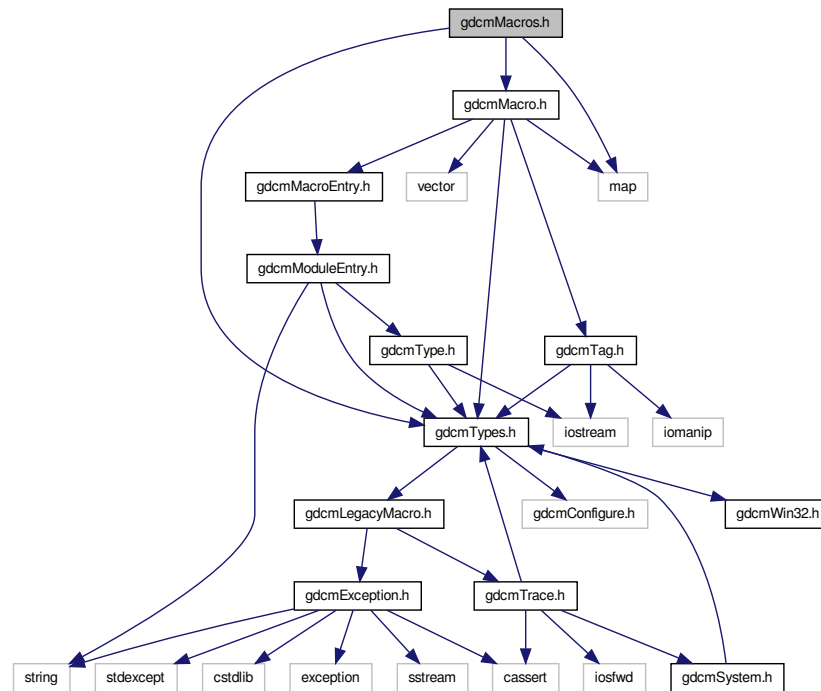
26.137.1 Macro Definition Documentation

26.137.1.1 `#define GDCMMACROENTRY_H`

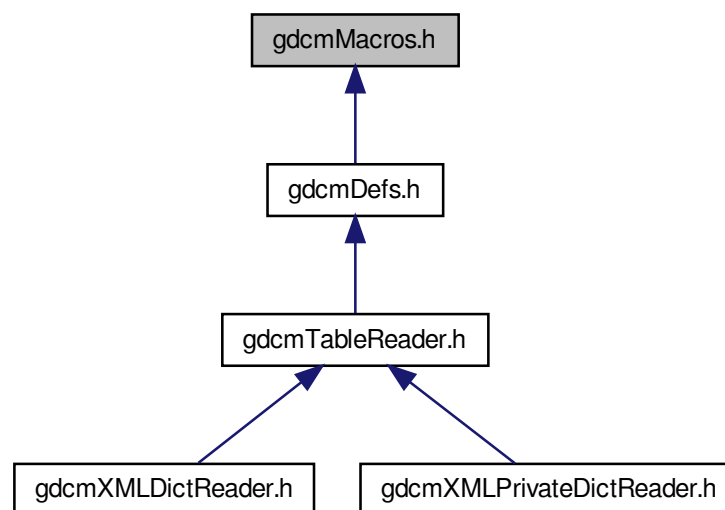
26.138 gdcMacros.h File Reference

```
#include "gdcTypes.h"  
#include "gdcMacro.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](#)

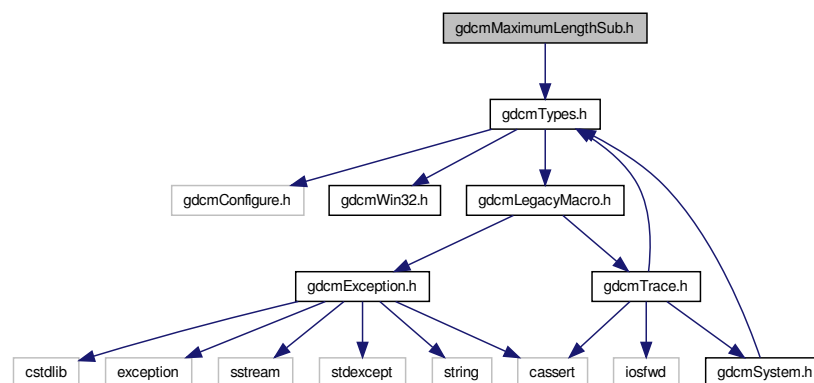
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)`

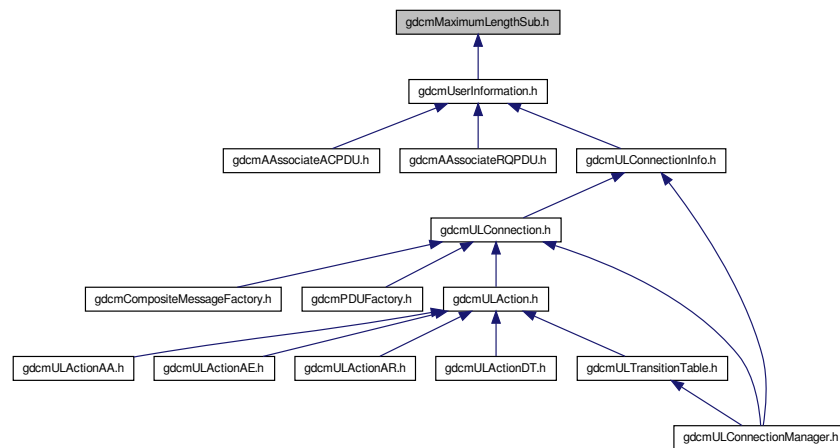
26.139 gdcmMaximumLengthSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmMaximumLengthSub.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcmd::network::MaximumLengthSub](#)

MaximumLengthSub Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

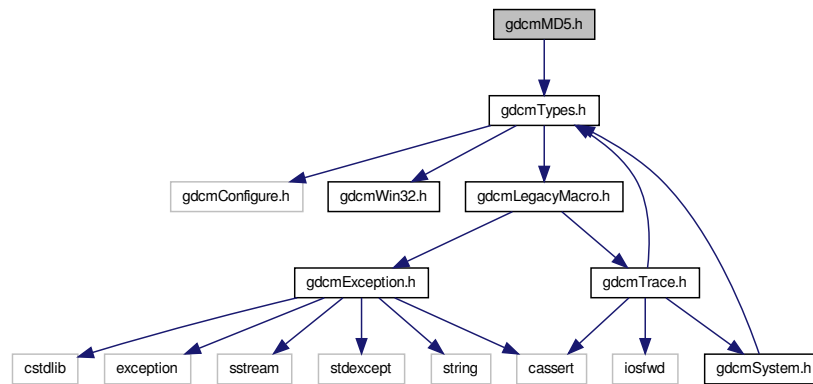
Namespaces

- [gdcmd](#)
- [gdcmd::network](#)

26.140 gdcmd5.h File Reference

```
#include "gdcmdTypes.h"
```

Include dependency graph for `gdcmd5.h`:



Classes

- class `gdcmd5::MD5`

Class for `MD5`.

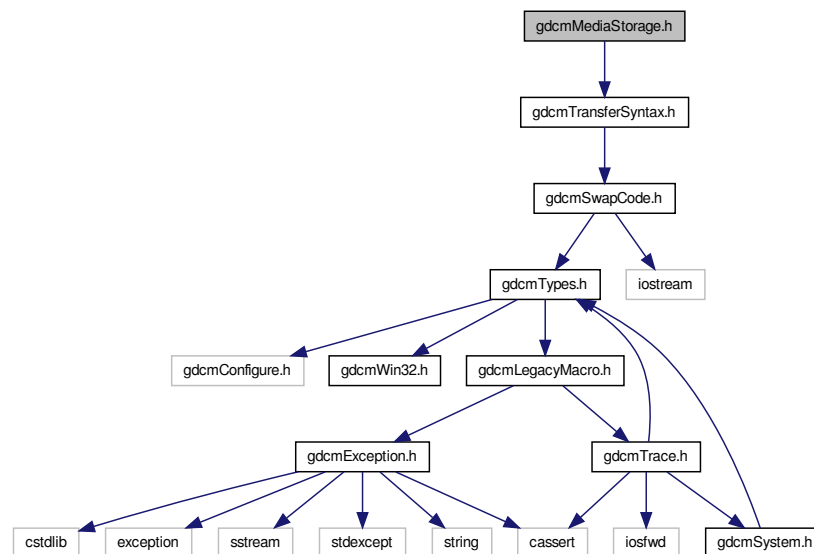
Namespaces

- `gdcmd5`

26.141 `gdcmd5MediaStorage.h` File Reference

```
#include "gdcmd5TransferSyntax.h"
```


Include dependency graph for gdcmMediaStorage.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::MediaStorage`
MediaStorage.

Namespaces

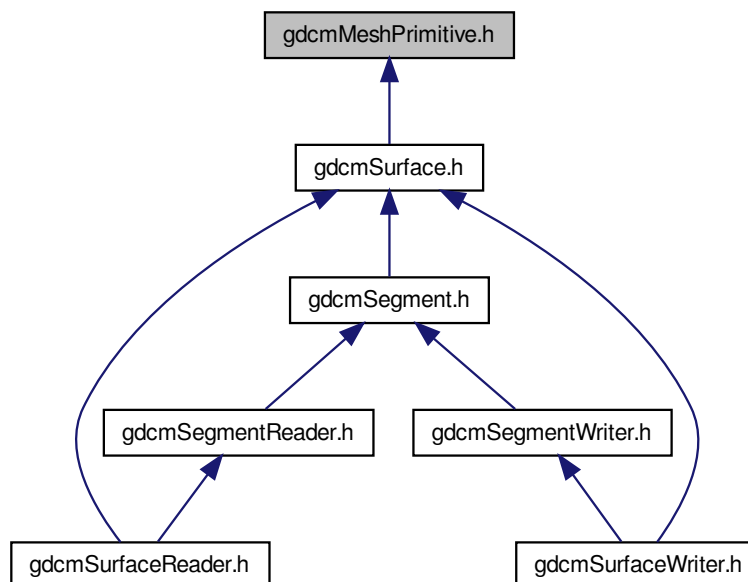
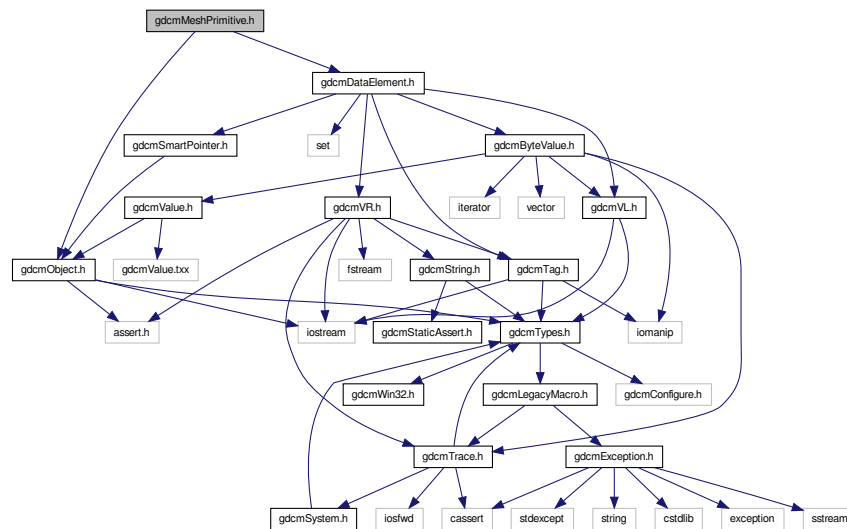
- `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)`

26.142 gdcmMeshPrimitive.h File Reference

```
#include <gdcmObject.h>
#include <gdcmDataElement.h>
```



Classes

- class [gdcm::MeshPrimitive](#)

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

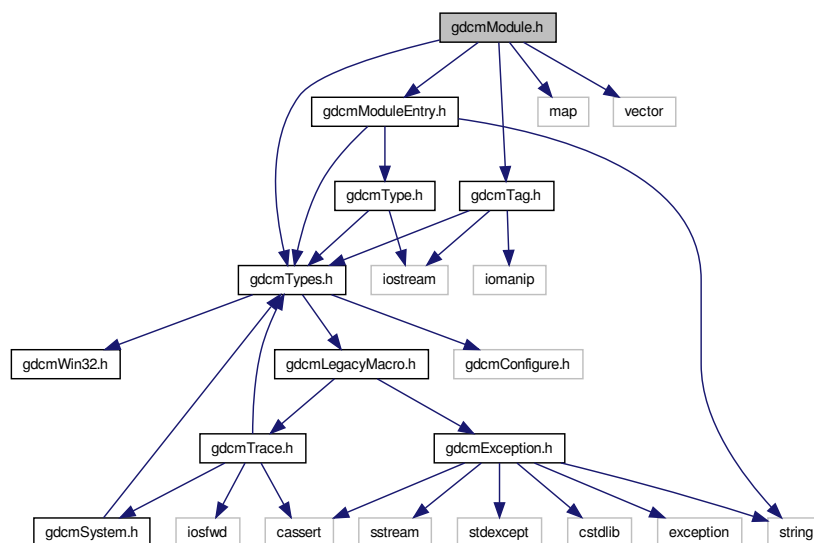
Namespaces

- [gdcm](#)

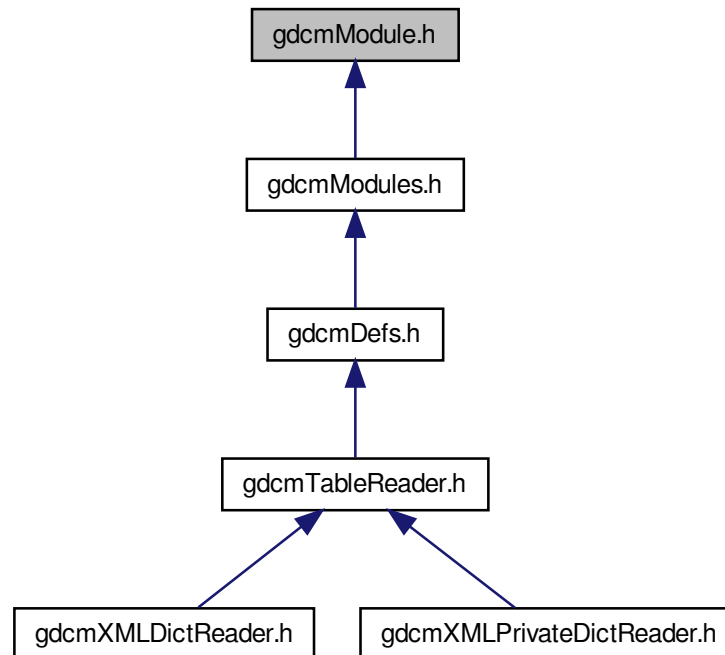
26.143 gdcmModule.h File Reference

```
#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

- [gdc](#)

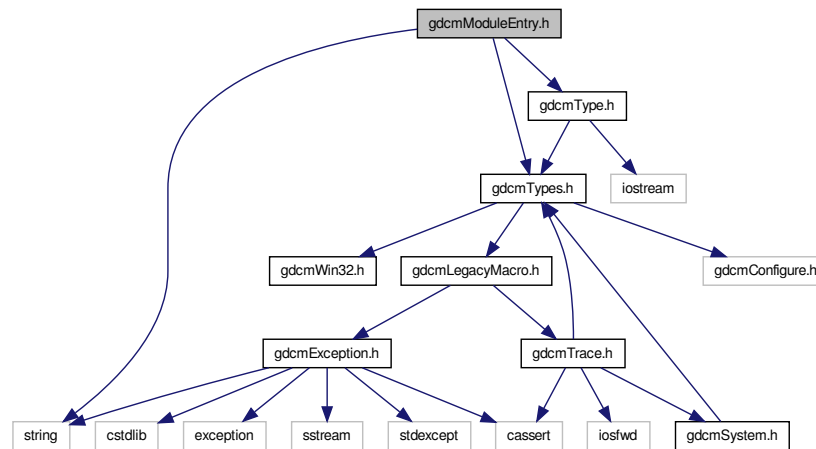
Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)`

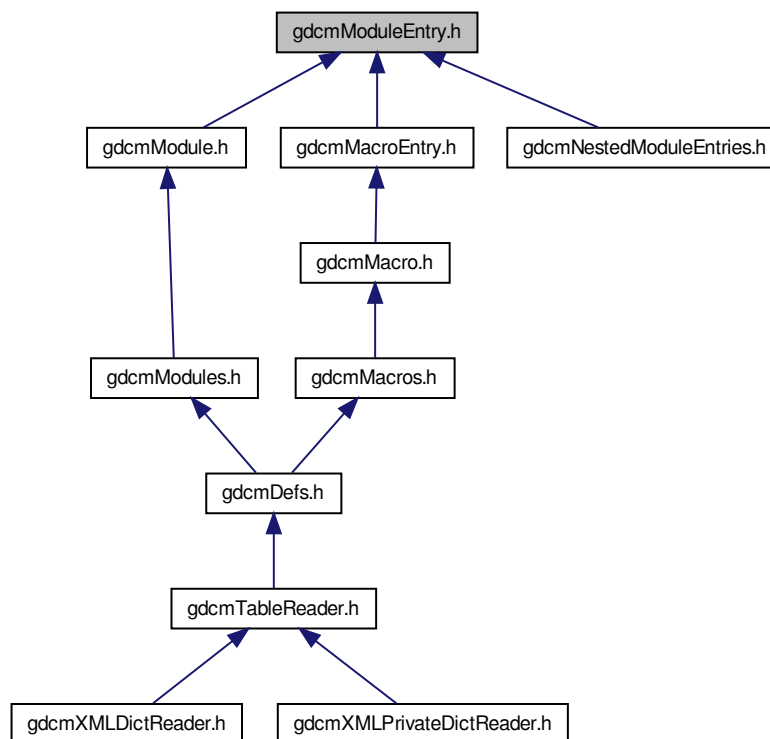
26.144 gdcModuleEntry.h File Reference

```
#include "gdcTypes.h"  
#include "gdcType.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](#)

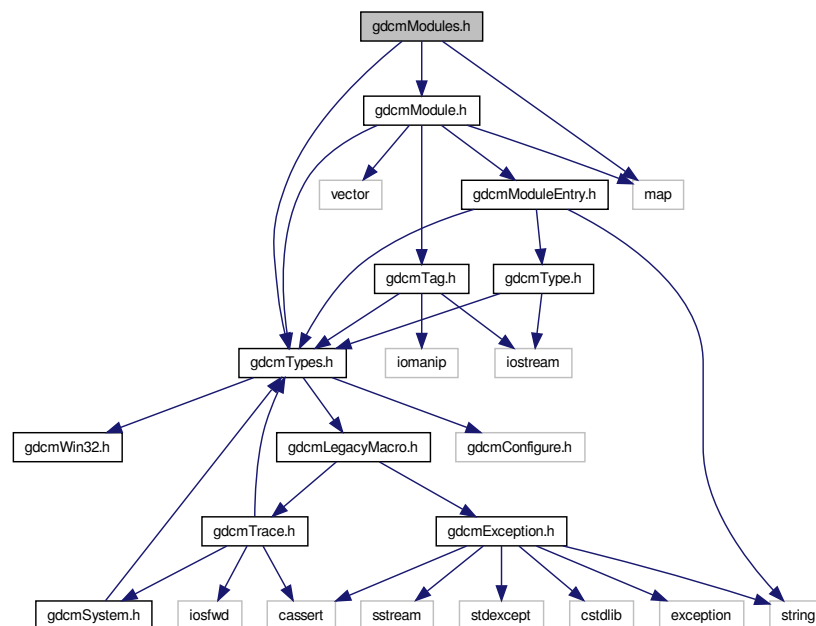
Functions

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

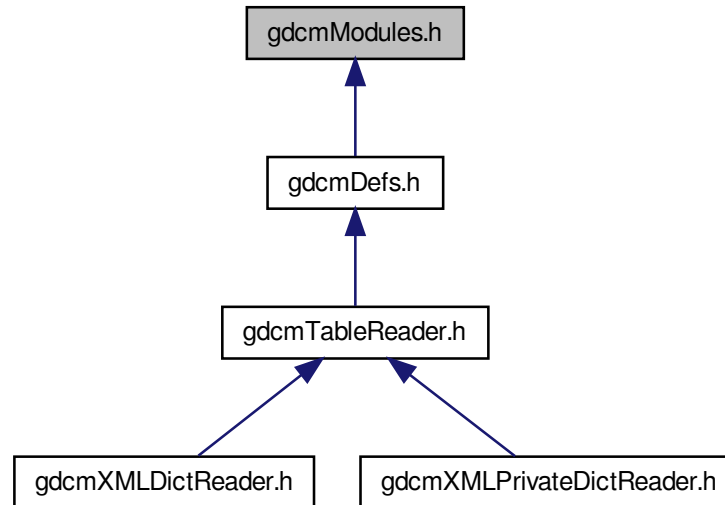
26.145 gdcmModules.h File Reference

```
#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

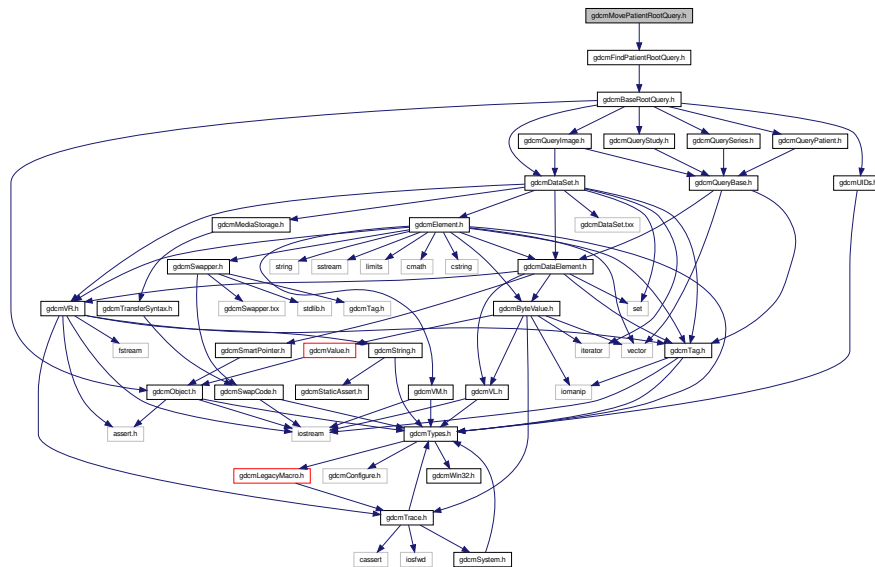
- [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)`

26.146 gdcmMovePatientRootQuery.h File Reference

```
#include "gdcmFindPatientRootQuery.h"
```



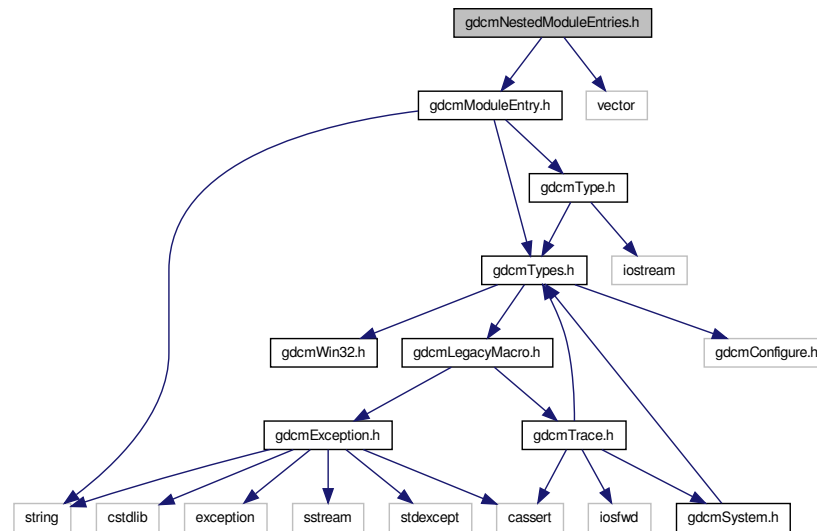
- class `gdcm::MoveStudyRootQuery`

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

- gdcm

```
#include "gdcmModuleEntry.h"
#include <vector>
```

Include dependency graph for `gdcnNestedModuleEntries.h`:



Classes

- class [gdcn::NestedModuleEntries](#)
Class for representing a *NestedModuleEntries*.

Namespaces

- [gdcn](#)

Typedefs

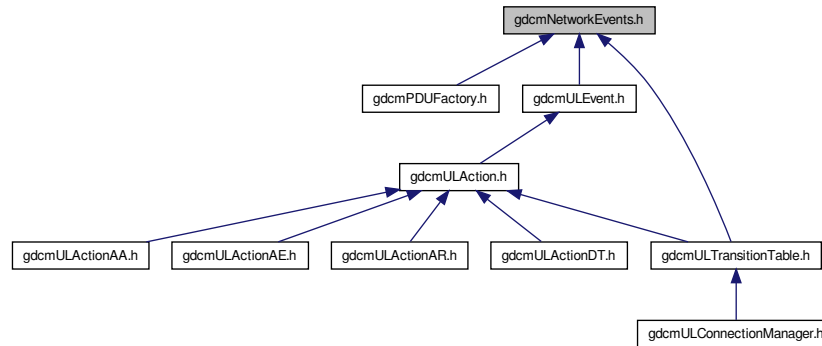
- typedef NestedModuleEntries [gdcn::NestedMacroEntries](#)

Functions

- `std::ostream & gdcn::operator<< (std::ostream &_os, const NestedModuleEntries &_val)`

26.149 gdcNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [gdc](#)
- [gdc::network](#)

Enumerations

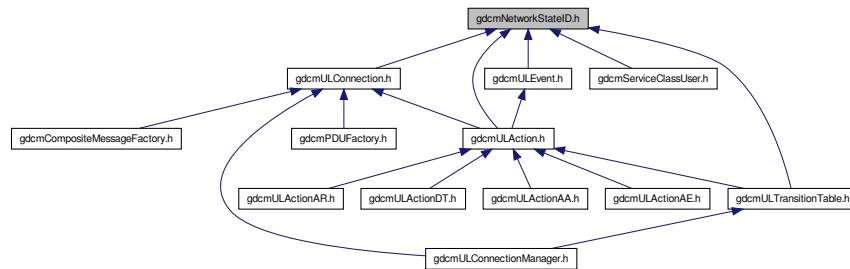
- `enum gdc::network::EEventID {`
`gdc::network::eAASSOCIATERequestLocalUser = 0,`
`gdc::network::eTransportConnConfirmLocal,`
`gdc::network::eASSOCIATE_ACPDUreceived,`
`gdc::network::eASSOCIATE_RJPDUreceived,`
`gdc::network::eTransportConnIndicLocal,`
`gdc::network::eAASSOCIATE_RQPDUreceived,`
`gdc::network::eAASSOCIATEResponseAccept,`
`gdc::network::eAASSOCIATEResponseReject,`
`gdc::network::ePDATArequest,`
`gdc::network::ePDATATFPDU,`
`gdc::network::eARELEASERequest,`
`gdc::network::eARELEASE_RQPDUReceivedOpen,`
`gdc::network::eARELEASE_RPPDUReceived,`
`gdc::network::eARELEASEResponse,`
`gdc::network::eAABORTRequest,`
`gdc::network::eAABORTPDUReceivedOpen,`
`gdc::network::eTransportConnectionClosed,`
`gdc::network::eARTIMTimerExpired,`
`gdc::network::eUnrecognizedPDUReceived,`
`gdc::network::eEventDoesNotExist }`

Variables

- `const int gdc::network::cMaxEventID = eEventDoesNotExist`

26.150 gdcmlNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [gdcml](#)
- [gdcml::network](#)

Enumerations

- `enum gdcml::network::EStateID {`
`gdcml::network::eStaDoesNotExist = 0,`
`gdcml::network::eSta1Idle = 1,`
`gdcml::network::eSta2Open = 2,`
`gdcml::network::eSta3WaitLocalAssoc = 4,`
`gdcml::network::eSta4LocalAssocDone = 8,`
`gdcml::network::eSta5WaitRemoteAssoc = 16,`
`gdcml::network::eSta6TransferReady = 32,`
`gdcml::network::eSta7WaitRelease = 64,`
`gdcml::network::eSta8WaitLocalRelease = 128,`
`gdcml::network::eSta9ReleaseCollisionRqLocal = 256,`
`gdcml::network::eSta10ReleaseCollisionAc = 512,`
`gdcml::network::eSta11ReleaseCollisionRq = 1024,`
`gdcml::network::eSta12ReleaseCollisionAcLocal = 2048,`
`gdcml::network::eSta13AwaitingClose = 4096 }`

Functions

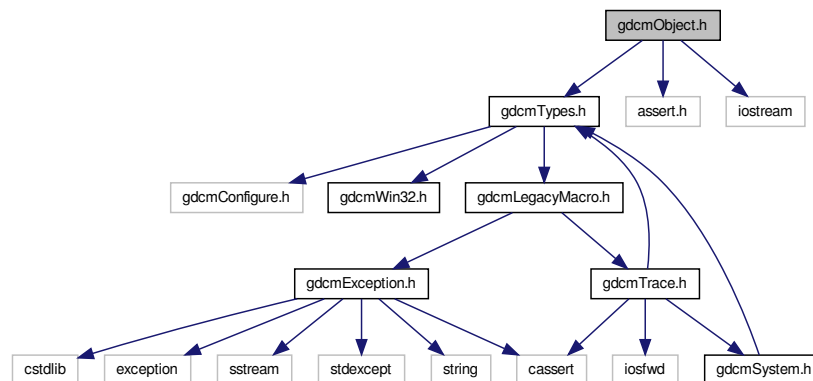
- `int gdcml::network::GetStateIndex (EStateID inState)`

Variables

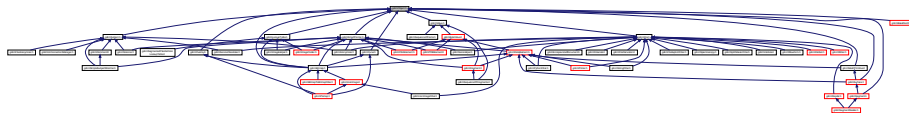
- `const int gdcml::network::cMaxStateID = 13`

26.151 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](#)

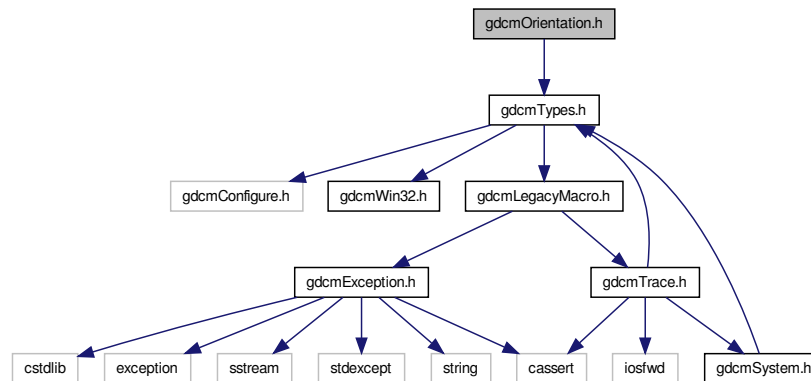
Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)`

26.152 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)`

26.153 gdcmOverlay.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmObject.h"
```

```

graph TD
    gdcmOverlay.h[gdcmOverlay.h] --> gdcmObject.h[gdcmObject.h]
    gdcmOverlay.h --> gdcmTypes.h[gdcmTypes.h]
    gdcmObject.h --> gdcmTypes.h
    gdcmObject.h --> assert.h[assert.h]
    gdcmObject.h --> iostream[iostream]
    gdcmTypes.h --> gdcmConfigure.h[gdcmConfigure.h]
    gdcmTypes.h --> gdcmWin32.h[gdcmWin32.h]
    gdcmTypes.h --> gdcmLegacyMacro.h[gdcmLegacyMacro.h]
    gdcmTypes.h --> gdcmException.h[gdcmException.h]
    gdcmTypes.h --> gdcmTrace.h[gdcmTrace.h]
    gdcmLegacyMacro.h --> gdcmLegacyMacro.h
    gdcmException.h --> cstdlib[cstdlib]
    gdcmException.h --> exception[exception]
    gdcmException.h --> sstream[sstream]
    gdcmException.h --> stdexcept[stdexcept]
    gdcmException.h --> string[string]
    gdcmException.h --> cassert[cassert]
    gdcmException.h --> iosfwd[iosfwd]
    gdcmException.h --> gdcmSystem.h[gdcmSystem.h]
    gdcmTrace.h --> gdcmSystem.h
  
```

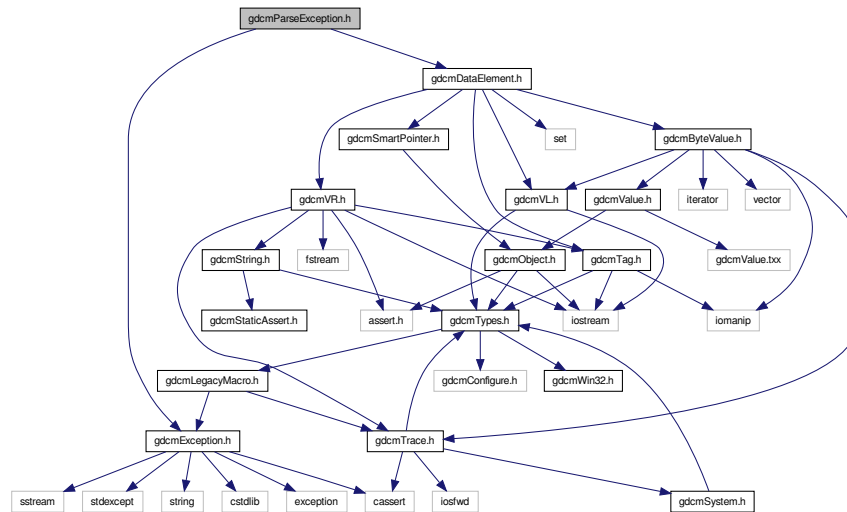
[illegible]

- class `gdcm::Overlay`
Overlay class.

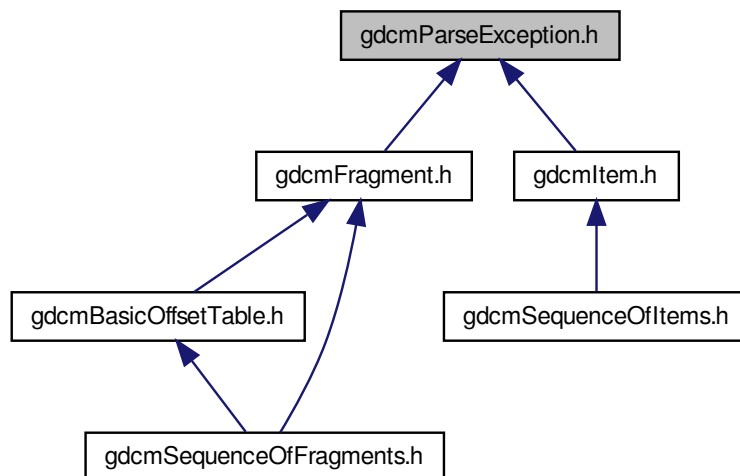
- **gdc**

```
#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.

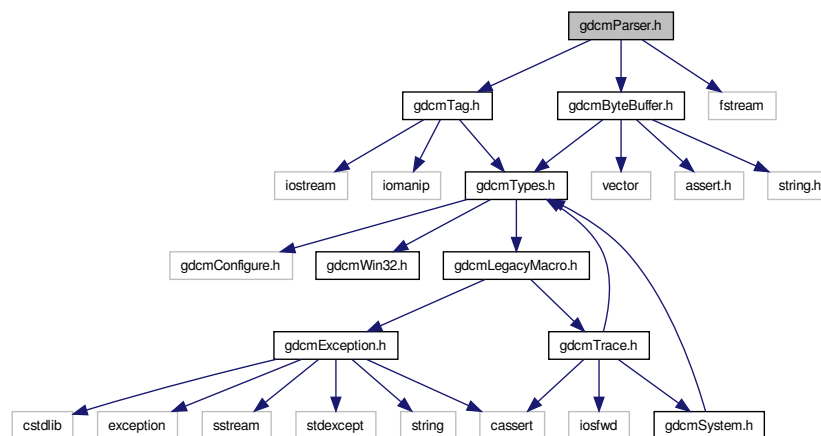
Namespaces

- [gdcm](#)

26.155 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)

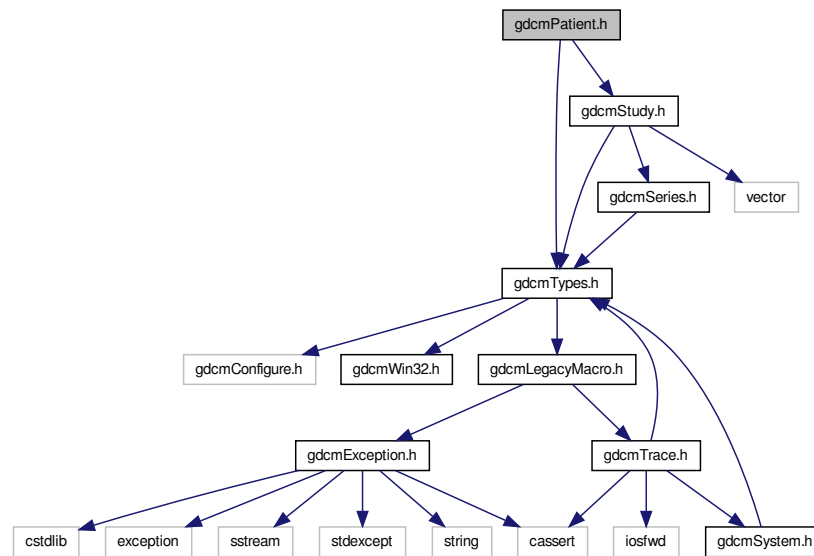
Namespaces

- [gdcm](#)

26.156 gdcmPatient.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmStudy.h"
```

Include dependency graph for gdcmPidient.h:



Classes

- class [gdcmPid::Patient](#)

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

Namespaces

- [gdcmPid](#)

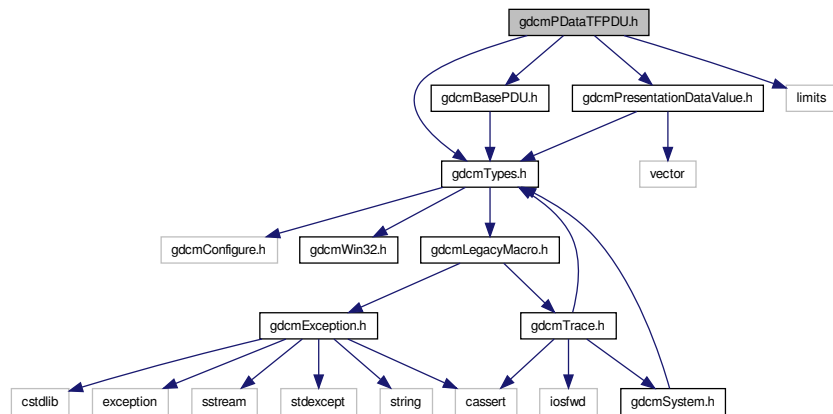
26.157 gdcmPidDataTFPDU.h File Reference

```

#include "gdcmPidTypes.h"
#include "gdcmPidPresentationDataValue.h"
#include "gdcmPidBasePDU.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](#)

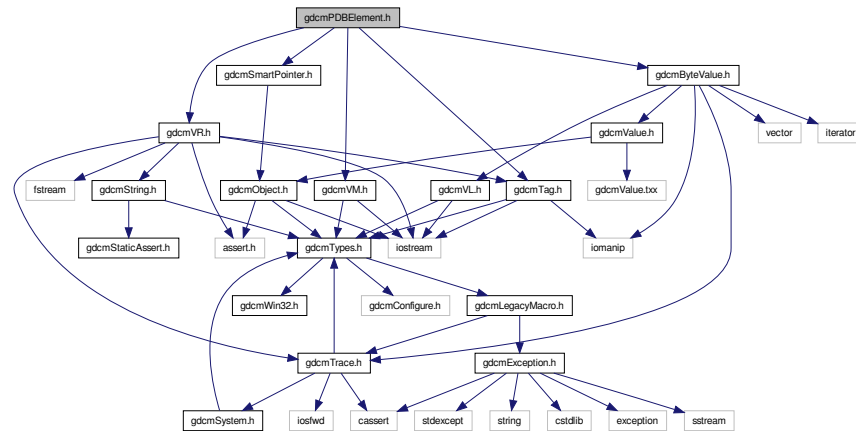
26.158 gdcmPDBelement.h File Reference

```

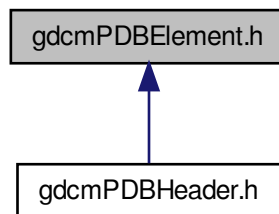
#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

```

Include dependency graph for `gdcnPDBElement.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcn::PDBElement`
Class to represent a PDB [Element](#).

Namespaces

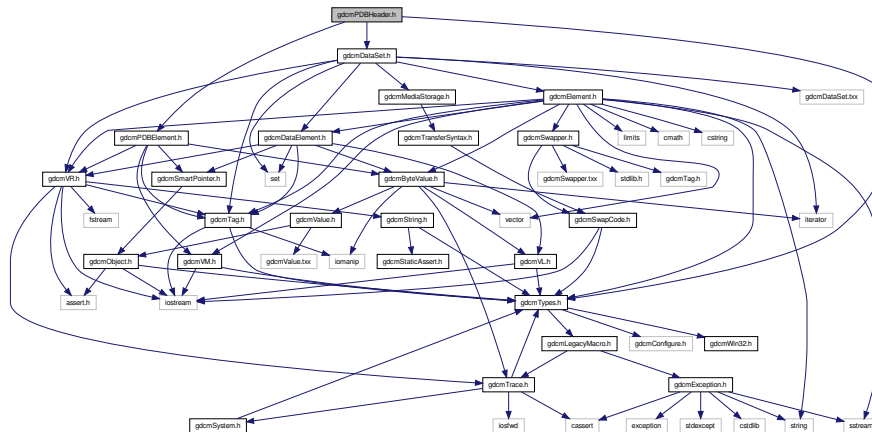
- `gdcn`

Functions

- `std::ostream & gdcn::operator<< (std::ostream &os, const PDBElement &val)`

26.159 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

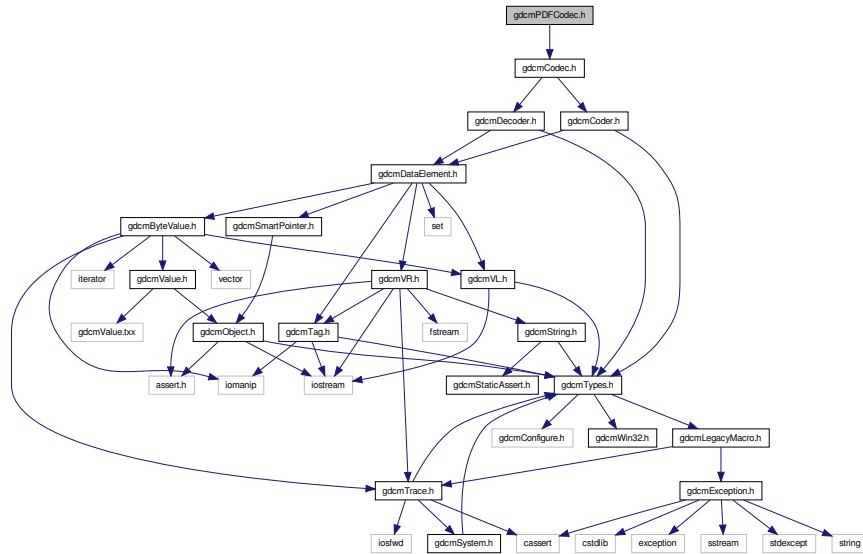
- `std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)`

26.160 gdcmpdf.man File Reference

26.161 gdcmPDFCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for `gdcmPDFCodec.h`:



Classes

- class [gdcm::PDFCodec](#)

PDFCodec class.

Namespaces

- [gdcm](#)

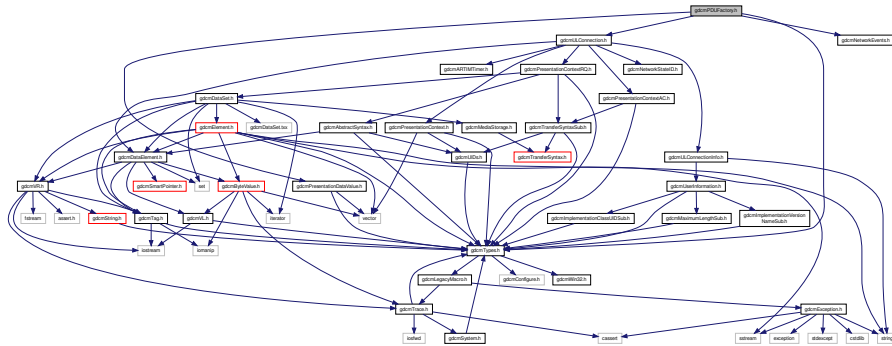
26.162 gdcmPDUFactory.h File Reference

```

#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.

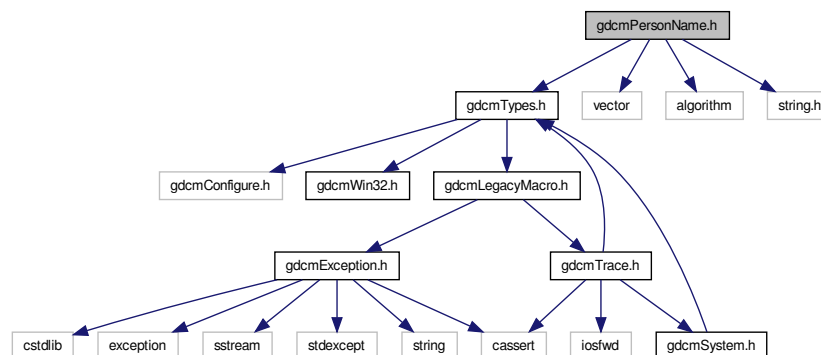
Namespaces

- [gdcm](#)
- [gdcm::network](#)

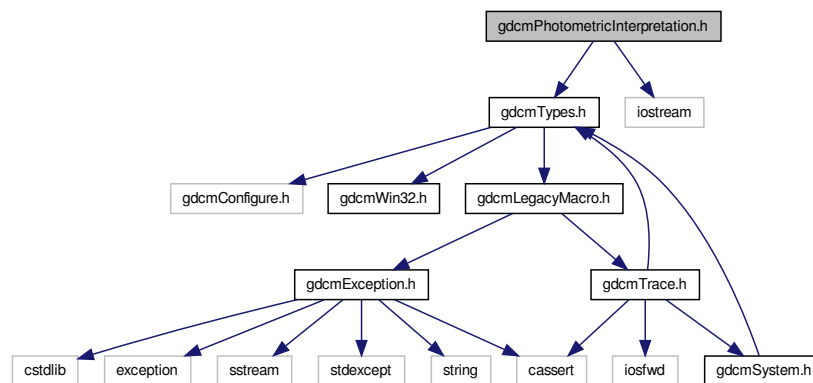
26.163 gdcmPersonName.h File Reference

```
#include "gdcmTypes.h"
#include <vector>
#include <algorithm>
#include <string.h>
```

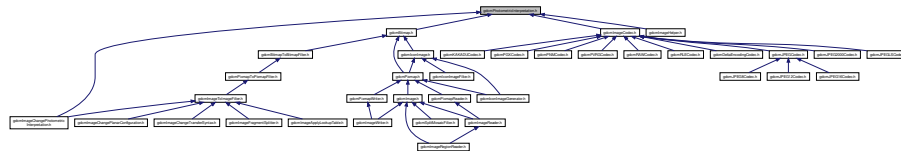
Include dependency graph for gdcmPersonName.h:



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)`

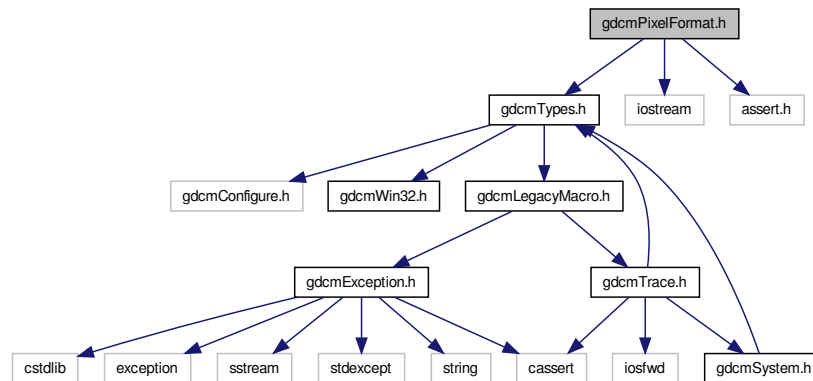
26.166 gdcmPixelFormat.h File Reference

```

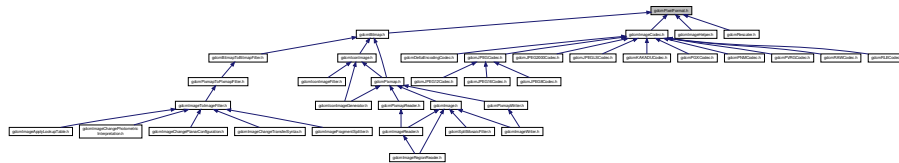
#include "gdcmTypes.h"
#include <iostream>
#include <assert.h>

```

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)`

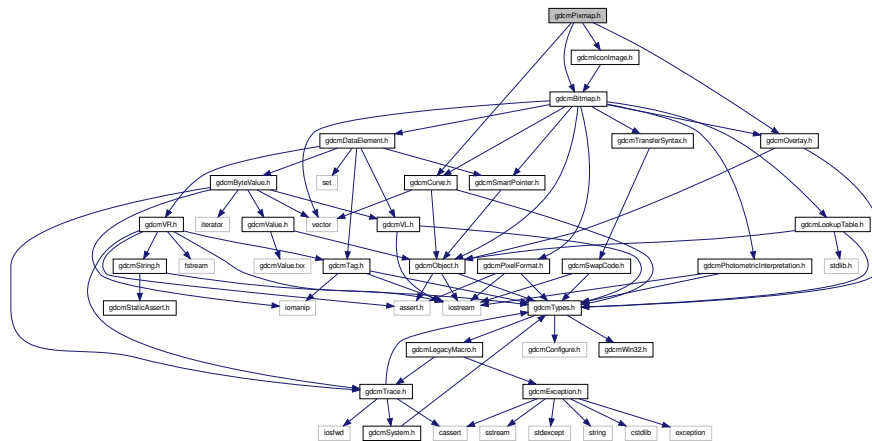
26.167 gdcmPixmap.h File Reference

```

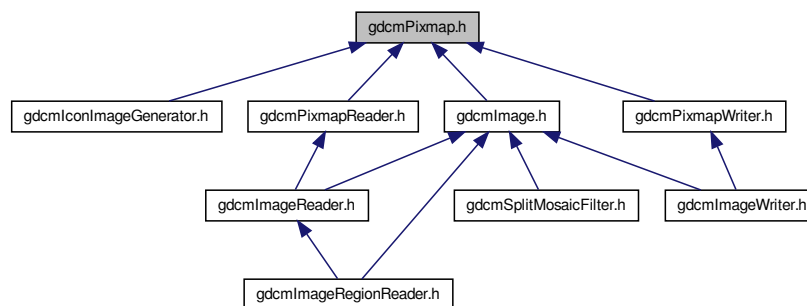
#include "gdcmBitmap.h"
#include "gdcmCurve.h"
#include "gdcmIconImage.h"
#include "gdcmOverlay.h"

```

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

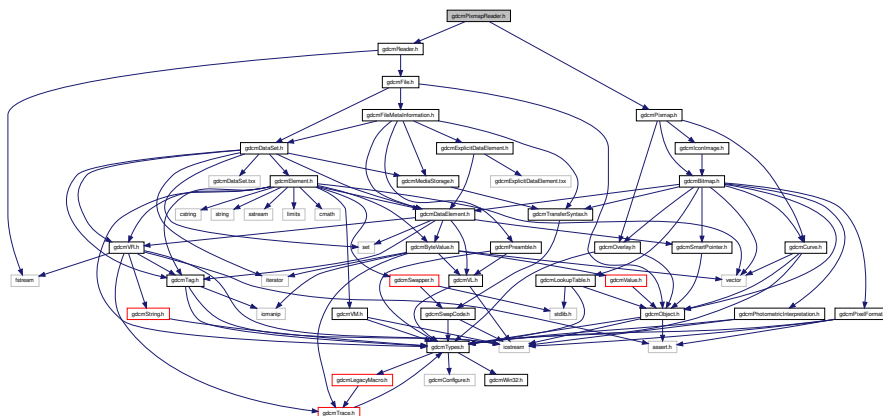
- [gdcm](#)

26.168 gdcmPixmapReader.h File Reference

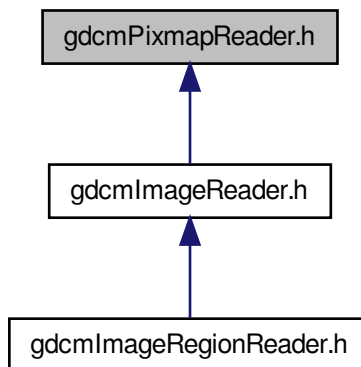
```
#include "gdcmReader.h"
```

```
#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.

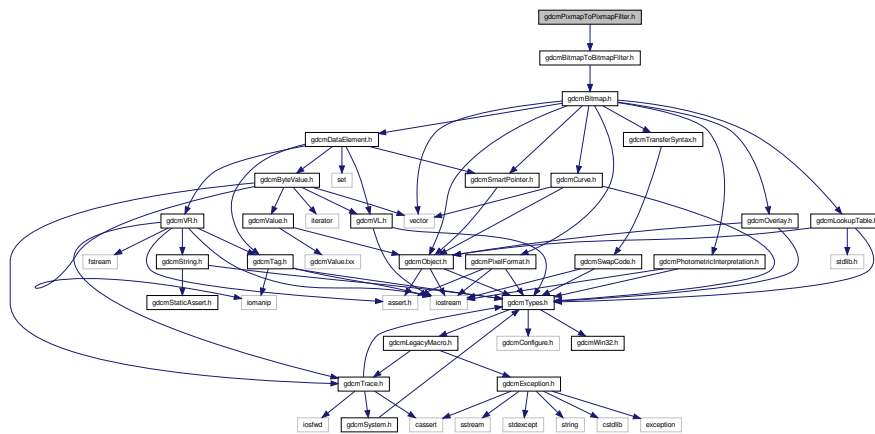
Namespaces

- [gdcm](#)

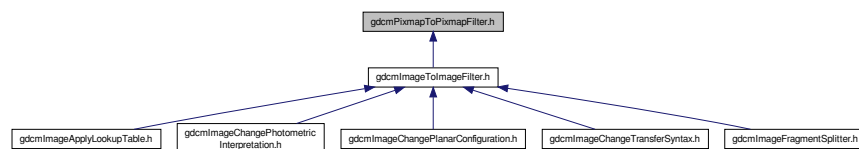
26.169 gdcmPixmapToPixmapFilter.h File Reference

```
#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](#)

26.170 gdcmPixmapWriter.h File Reference

```
#include "gdcmWriter.h"
```

```
#include "gdcmPixmap.h"
```

The graph illustrates the intricate dependencies within the glibc project. Key components include:

- Internal Headers:** A large cluster of nodes at the top, such as `glibc/symtab.h`, `glibc/compat.h`, and `glibc/compat.h`, which serve as the foundation for the library's internal structure.
- System Headers:** Nodes like `glibc/compat.h` and `glibc/compat.h` represent dependencies on system-level headers.
- User-space Libraries:** Nodes like `glibc/compat.h` and `glibc/compat.h` represent dependencies on user-space libraries.
- Build Configuration:** Nodes like `glibc/compat.h` and `glibc/compat.h` represent build configuration files.

The red highlighting indicates the current state of the build, showing which files are being processed or are the most relevant to the current context.

```
graph BT; gdcmImageWriter.h --> gdcmPixmapWriter.h
```

- class `gdcm::PixmapWriter`
PixmapWriter This class will takes two inputs:

- **gdcm**

```
#include "gdcmImageCodec.h"
```

[illegible]

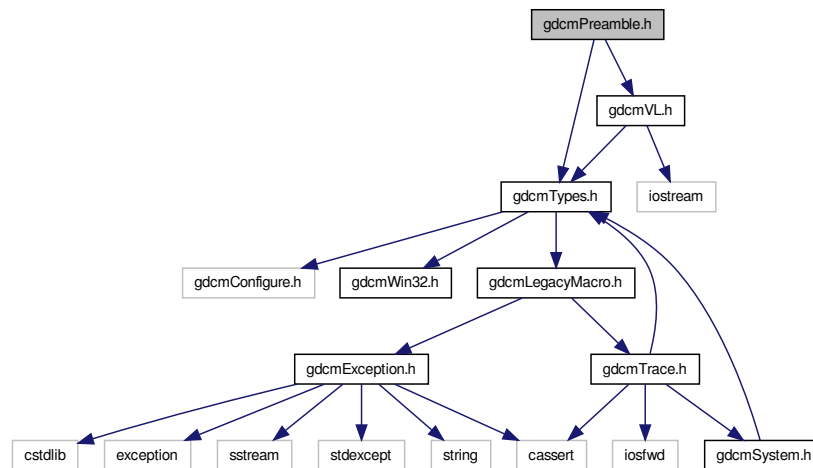
- class `gdcm::PNMCodec`

aces

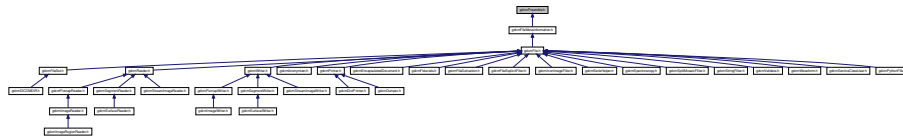
- **gdcm**

```
#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)`

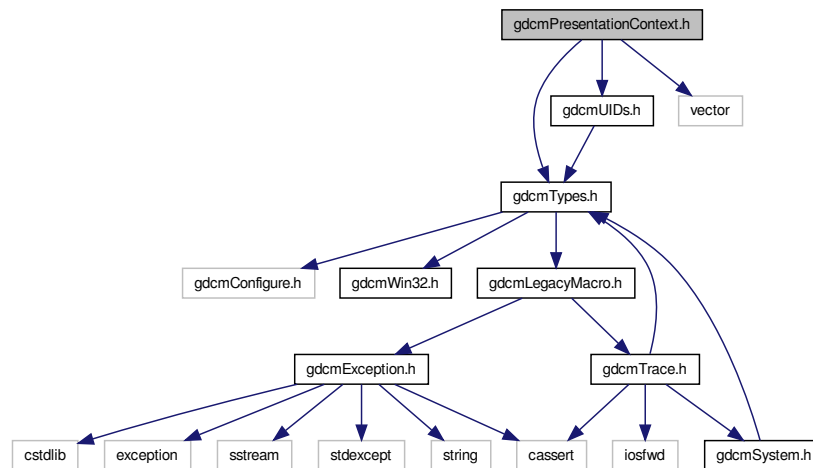
26.173 gdcmPresentationContext.h File Reference

```

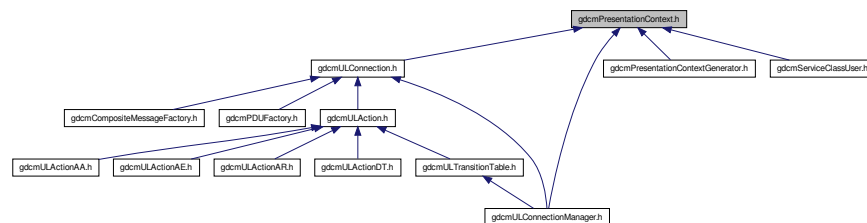
#include "gdcmTypes.h"
#include "gdcmUIDs.h"
#include <vector>

```


Include dependency graph for gdcmPresentationContext.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gdcm::PresentationContext](#)
PresentationContext.

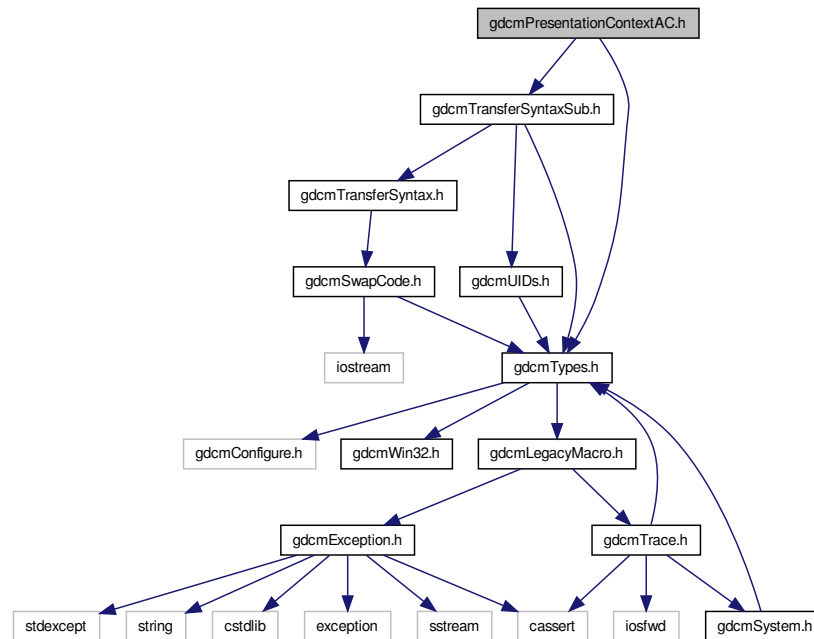
Namespaces

- [gdcm](#)

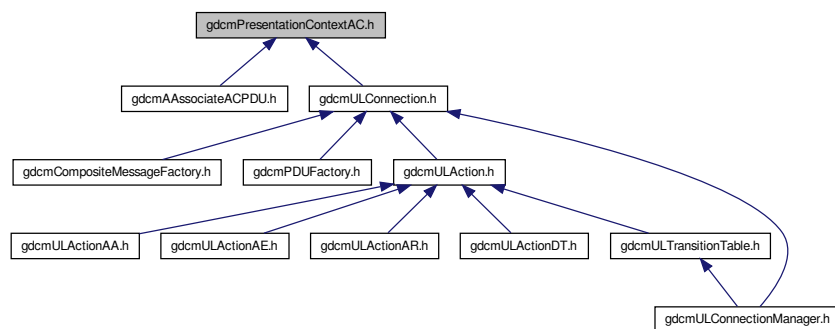
26.174 gdcmPresentationContextAC.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTransferSyntaxSub.h"
```

Include dependency graph for `gdcmPidentationContextAC.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmPid::network::PresentationContextAC`

PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.

Namespaces

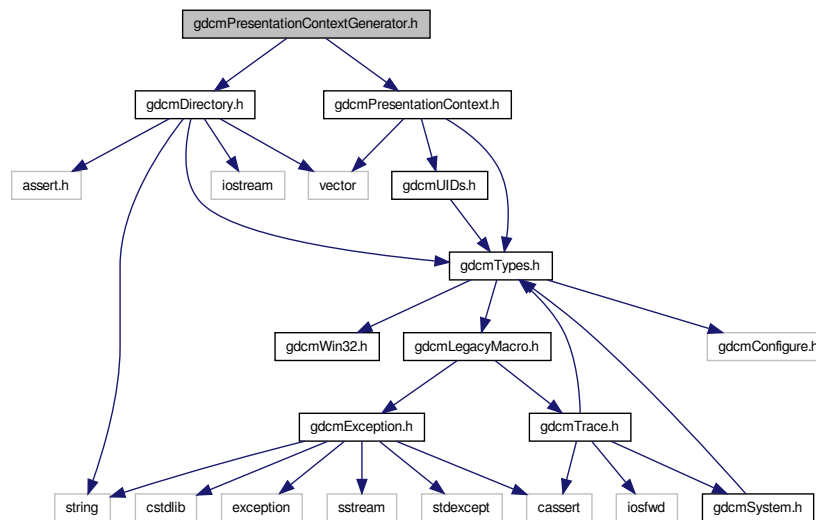
- [gdcm](#)
- [gdcm::network](#)

26.175 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](#)

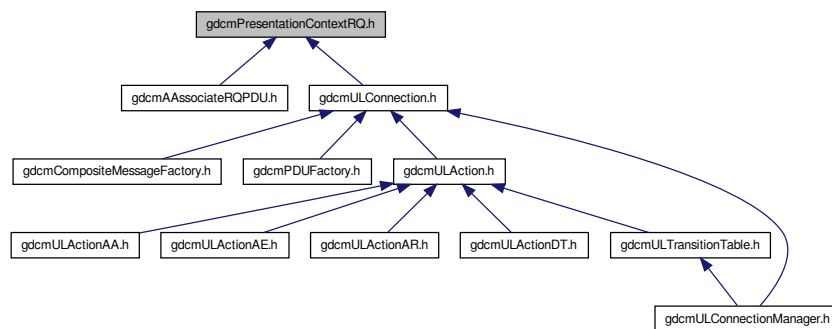
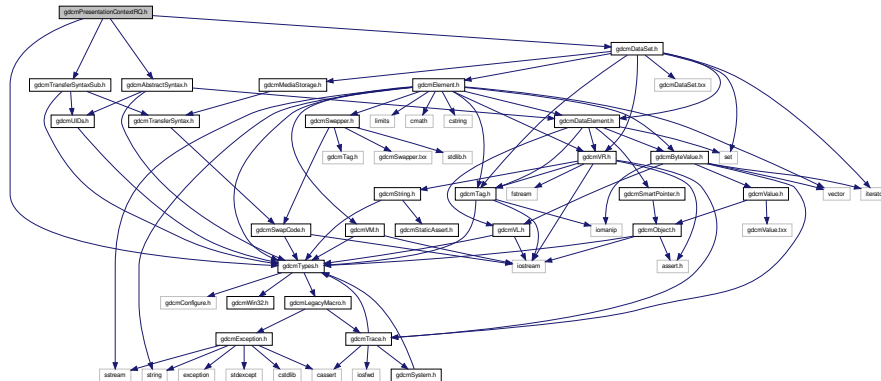
26.176 gdcmPresentationContextRQ.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmAbstractSyntax.h"
```

```
#include "gdcmTransferSyntaxSub.h"
```

```
#include "gdcmDataSet.h"
```

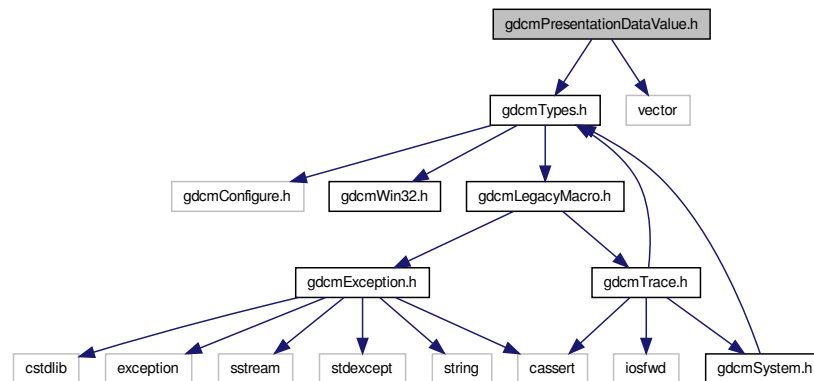


- class `gdcn::network::PresentationContextRQ`
- PresentationContextRQ Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.*

- `gdcm`
- `gdcm::network`

```
#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](#)

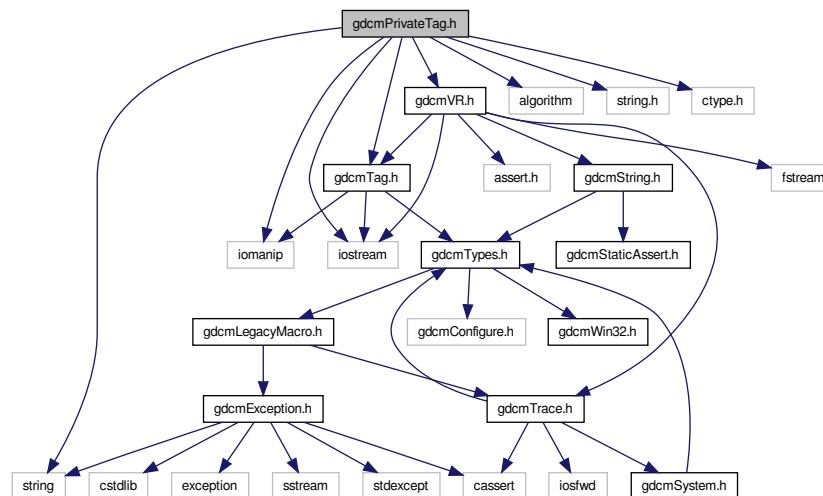
26.178 gdcmPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

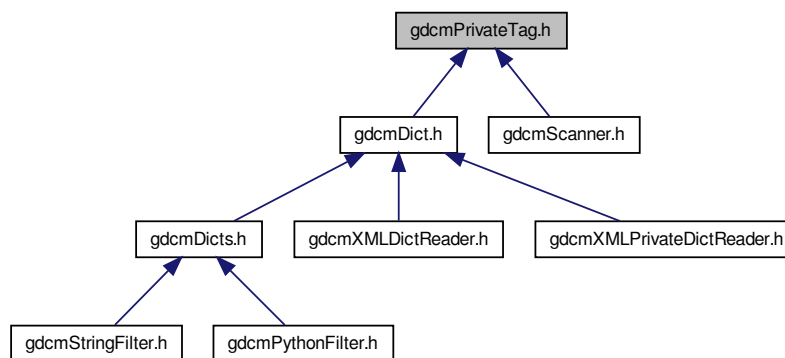


```
#include "gdcmVR.h"
#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>
#include <string.h>
#include <ctype.h>
```

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

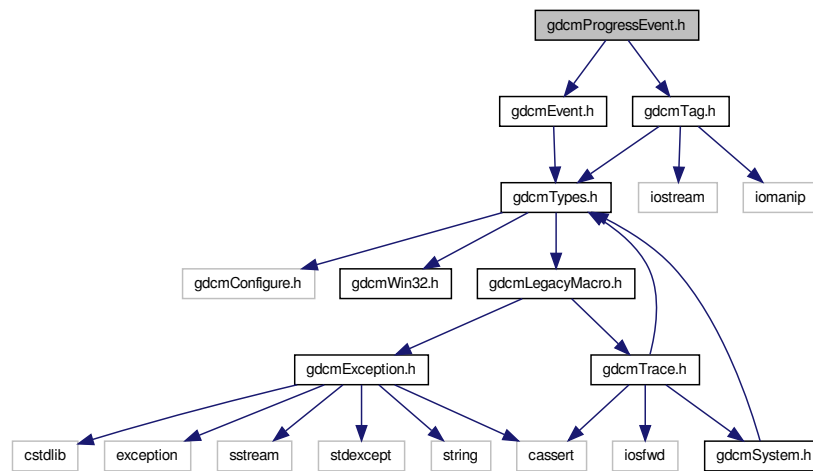
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)`

26.180 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.

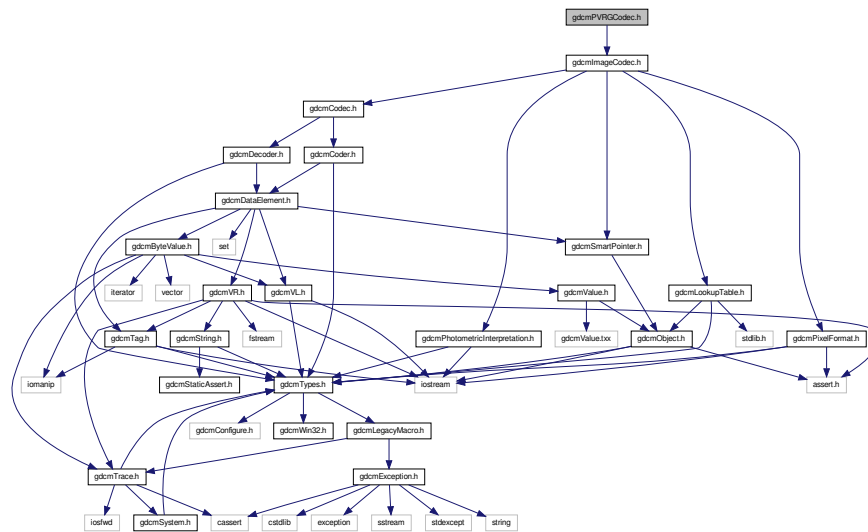
Namespaces

- [gdcm](#)

26.181 gdcmPVRGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```


Include dependency graph for gdcmPVRGCodec.h:



Classes

- class [gdcm::PVRGCodec](#)

PVRGCodec.

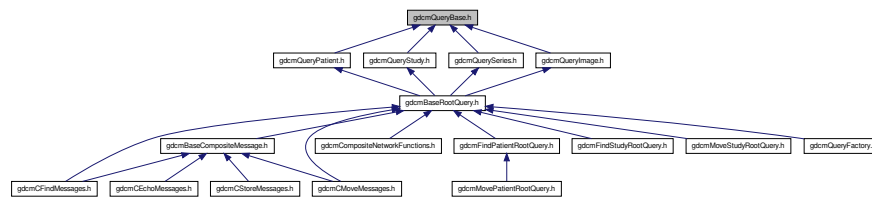
Namespaces

- [gdcm](#)

26.182 gdcmPythonFilter.h File Reference

```
#include <Python.h>
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.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](#)

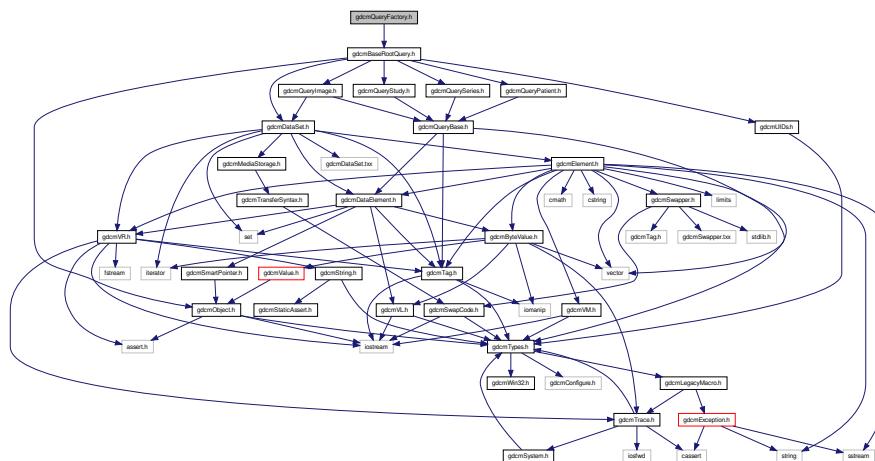
Enumerations

- enum [gdcm::ERootType](#) {
[gdcm::ePatientRootType](#),
[gdcm::eStudyRootType](#) }

26.184 gdcmQueryFactory.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcmQueryFactory.h:




```

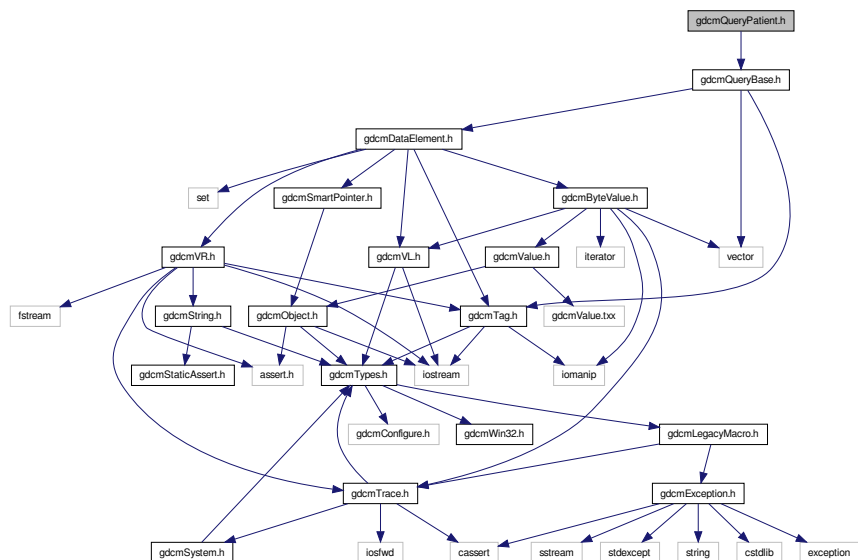
graph TD
    gdcmQueryImage["gdcmQueryImage.h"] --> gdcmBaseRootQuery["gdcmBaseRootQuery.h"]
    gdcmBaseRootQuery --> gdcmBaseCompositeMessage["gdcmBaseCompositeMessage.h"]
    gdcmBaseRootQuery --> gdcmComposeNetworkFunctions["gdcmComposeNetworkFunctions.h"]
    gdcmBaseRootQuery --> gdcmFindPatientRootQuery["gdcmFindPatientRootQuery.h"]
    gdcmBaseRootQuery --> gdcmFindStudyRootQuery["gdcmFindStudyRootQuery.h"]
    gdcmBaseRootQuery --> gdcmMoveStudyRootQuery["gdcmMoveStudyRootQuery.h"]
    gdcmBaseRootQuery --> gdcmQueryFactory["gdcmQueryFactory.h"]
    gdcmBaseCompositeMessage --> gdcmCFindMessages["gdcmCFindMessages.h"]
    gdcmBaseCompositeMessage --> gdcmCInfoMessages["gdcmCInfoMessages.h"]
    gdcmBaseCompositeMessage --> gdcmCStoreMessages["gdcmCStoreMessages.h"]
    gdcmComposeNetworkFunctions --> gdcmCMoveMessages["gdcmCMoveMessages.h"]
    gdcmFindPatientRootQuery --> gdcmMovePatientRootQuery["gdcmMovePatientRootQuery.h"]
  
```

- class `gdcm::QueryImage`

Namespaces

- **gdcm**

```
#include "gdcmQueryBase.h"
Include dependency graph for gdcmQueryPatient.h:
```



```

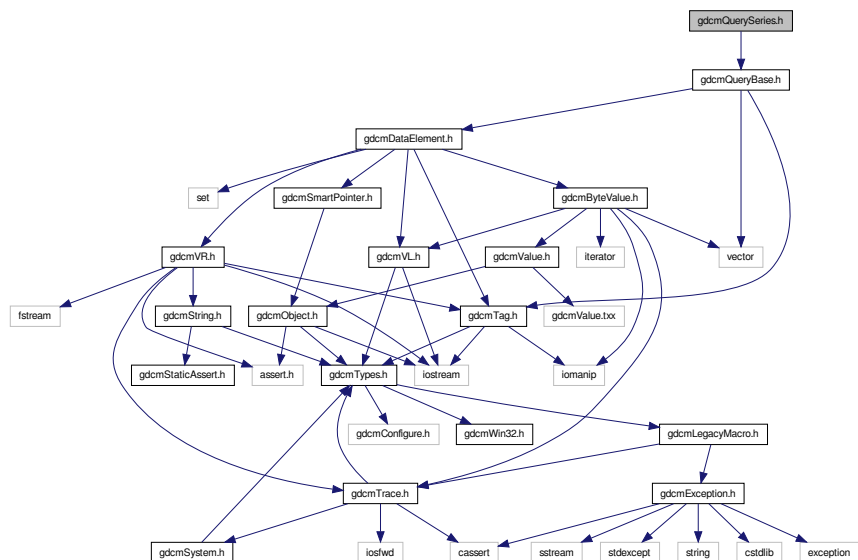
graph TD
    gdcmQueryPattern["gdcmQueryPattern.h"] --> gdcmBaseRootQuery["gdcmBaseRootQuery.h"]
    gdcmBaseRootQuery --> gdcmBaseCompositeMessage["gdcmBaseCompositeMessage.h"]
    gdcmBaseRootQuery --> gdcmComposeNetworkFunctions["gdcmComposeNetworkFunctions.h"]
    gdcmBaseRootQuery --> gdcmFindPatientRootQuery["gdcmFindPatientRootQuery.h"]
    gdcmBaseRootQuery --> gdcmFindStudyRootQuery["gdcmFindStudyRootQuery.h"]
    gdcmBaseRootQuery --> gdcmMoveStudyRootQuery["gdcmMoveStudyRootQuery.h"]
    gdcmBaseRootQuery --> gdcmQueryFactory["gdcmQueryFactory.h"]
    gdcmBaseCompositeMessage --> gdcmCFindMessages["gdcmCFindMessages.h"]
    gdcmBaseCompositeMessage --> gdcmCImageMessages["gdcmCImageMessages.h"]
    gdcmBaseCompositeMessage --> gdcmCMoveMessages["gdcmCMoveMessages.h"]
    gdcmComposeNetworkFunctions --> gdcmCImageMessages
    gdcmComposeNetworkFunctions --> gdcmCMoveMessages
    gdcmFindPatientRootQuery --> gdcmCImageMessages
    gdcmFindPatientRootQuery --> gdcmCMoveMessages
    gdcmFindStudyRootQuery --> gdcmCImageMessages
    gdcmFindStudyRootQuery --> gdcmCMoveMessages
    gdcmMoveStudyRootQuery --> gdcmCImageMessages
    gdcmMoveStudyRootQuery --> gdcmCMoveMessages
    gdcmQueryFactory --> gdcmCImageMessages
    gdcmQueryFactory --> gdcmCMoveMessages
  
```

- class `gdcm::QueryPatient`

Namespaces

- ## 26.187 gdcmQuerySeries.h File Reference

Include dependency graph for `gdcmQuerySeries.h`:



```

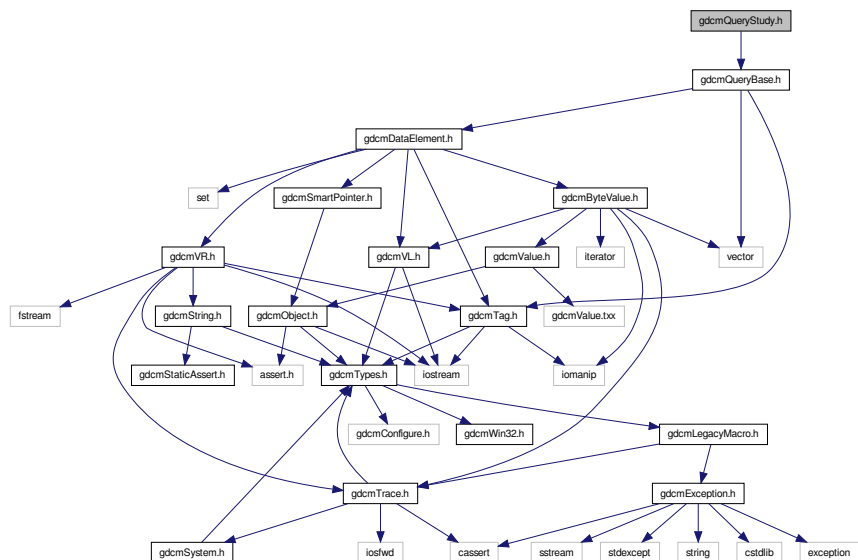
graph TD
    gtdcmQuerySeries.h --> gtdcmBaseRootQuery.h
    gtdcmBaseRootQuery.h --> gtdcmBaseCompositeMessage.h
    gtdcmBaseRootQuery.h --> gtdcmComposeNetworkFunctions.h
    gtdcmBaseRootQuery.h --> gtdcmFindPatientRootQuery.h
    gtdcmBaseRootQuery.h --> gtdcmFindStudyRootQuery.h
    gtdcmBaseRootQuery.h --> gtdcmMoveStudyRootQuery.h
    gtdcmBaseRootQuery.h --> gtdcmQueryFactory.h
    gtdcmBaseCompositeMessage.h --> gtdcmCFindMessages.h
    gtdcmBaseCompositeMessage.h --> gtdcmCInfoMessages.h
    gtdcmBaseCompositeMessage.h --> gtdcmCStoreMessages.h
    gtdcmComposeNetworkFunctions.h --> gtdcmCInfoMessages.h
    gtdcmComposeNetworkFunctions.h --> gtdcmCMoveMessages.h
    gtdcmFindPatientRootQuery.h --> gtdcmCMoveMessages.h
    gtdcmFindPatientRootQuery.h --> gtdcmMovePatientRootQuery.h
  
```

- class `gdcm::QuerySeries`

Namespaces

- **gdcm**

```
#include "gdcmQueryBase.h"
Include dependency graph for gdcmQueryStudy.h:
```



RAWCodec class.

Namespaces

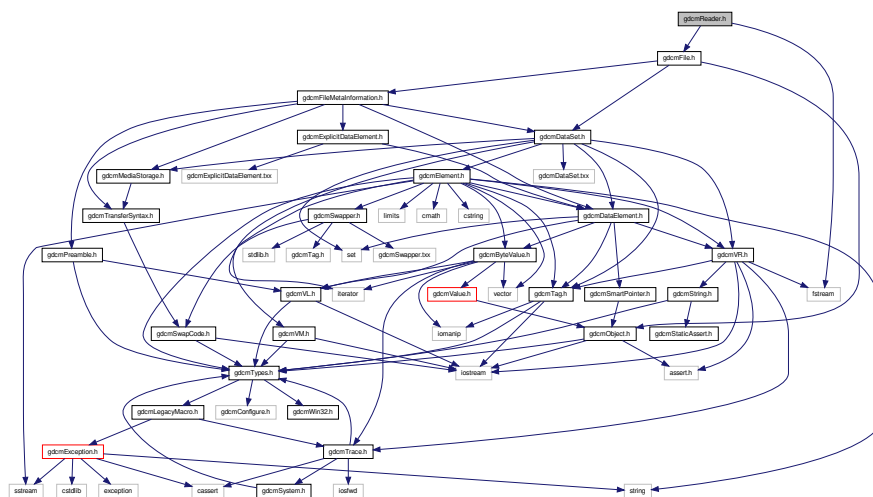
- [gdcm](#)

26.191 gdcmReader.h File Reference

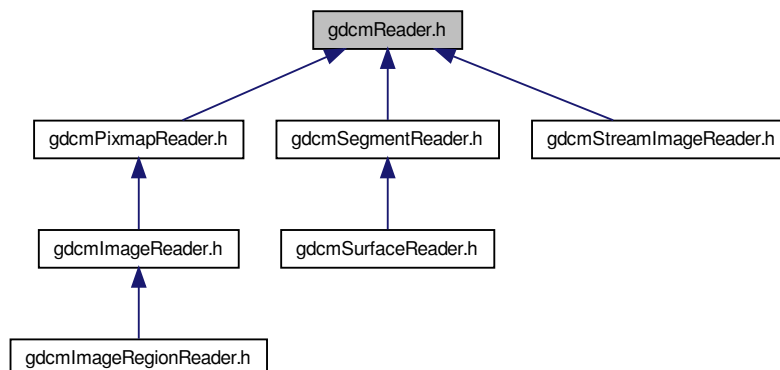
```
#include "gdcmFile.h"
```

```
#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](#)

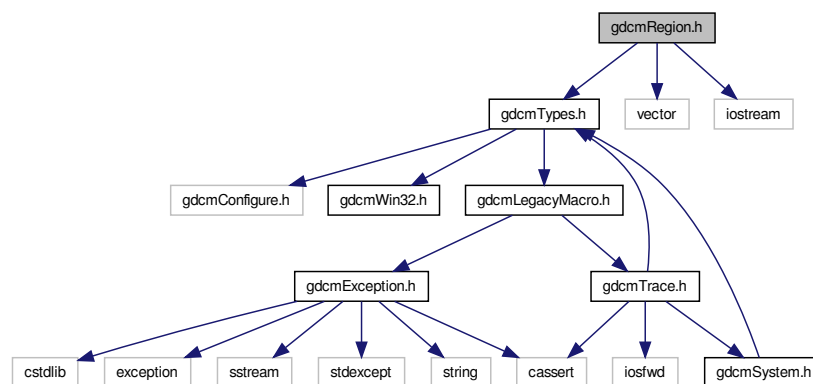
26.192 gdcmRegion.h File Reference

```
#include "gdcmTypes.h"
```

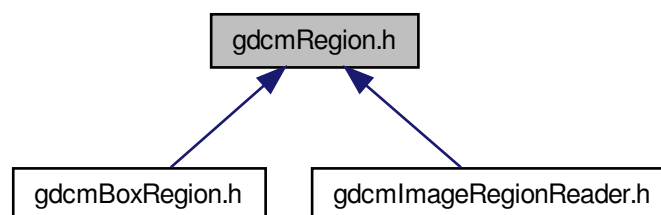
```
#include <vector>
```

```
#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

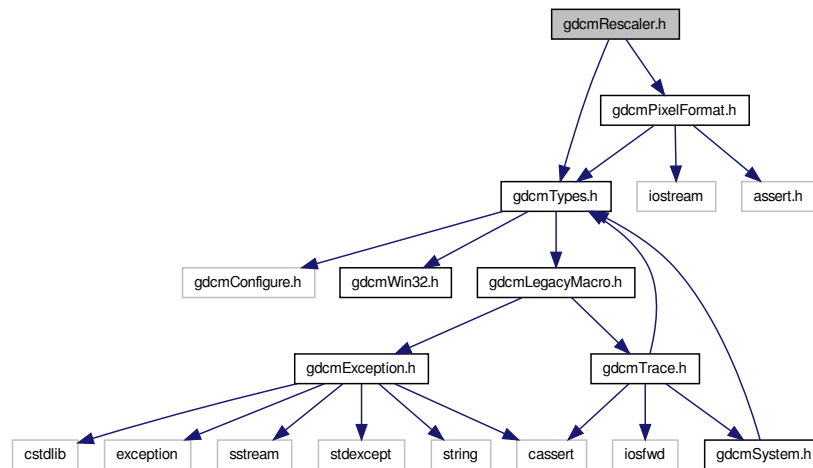
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const Region &r)

26.193 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmPixelFormat.h"
```

Include dependency graph for gdcmRescaler.h:



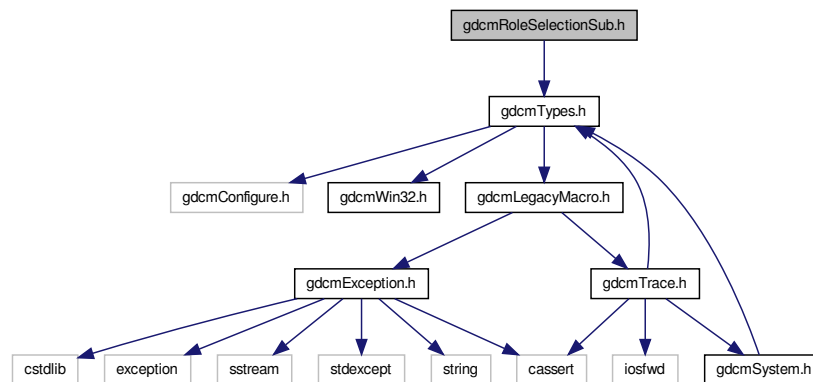
Classes

- 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.

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](#)

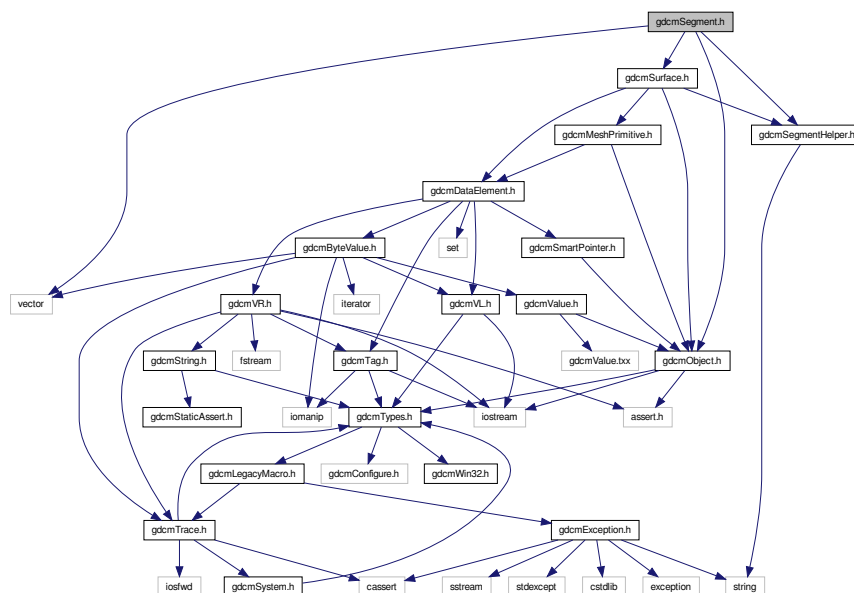
26.196 gdcmScanner.h File Reference

```

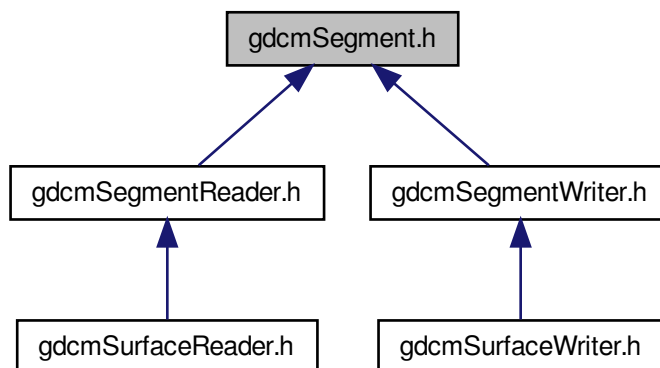
#include "gdcmDirectory.h"
#include "gdcmSubject.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmSmartPointer.h"
#include <map>
#include <set>
#include <string>
#include <string.h>

```


Include dependency graph for `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.

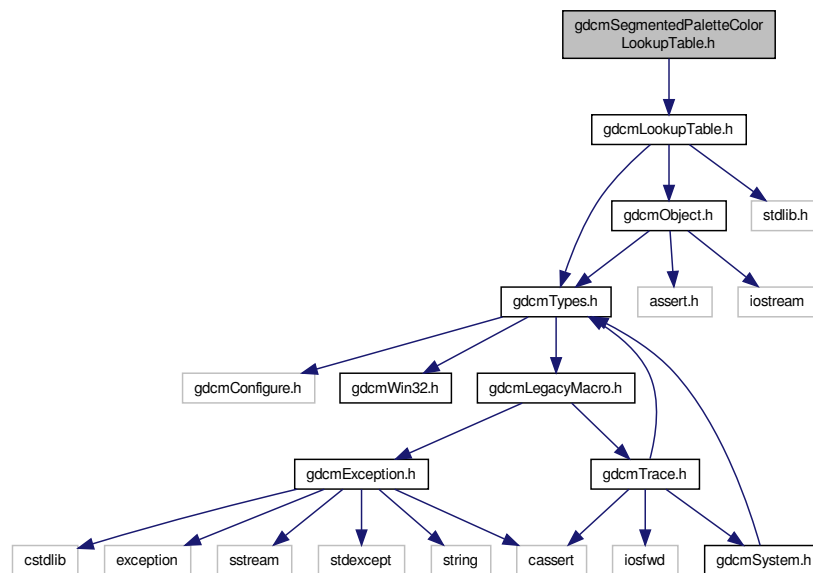
Namespaces

- [gdcm](#)

26.200 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



Classes

- class [gdcm::SegmentedPaletteColorLookupTable](#)
SegmentedPaletteColorLookupTable class.

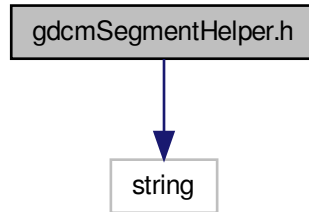
Namespaces

- [gdcm](#)

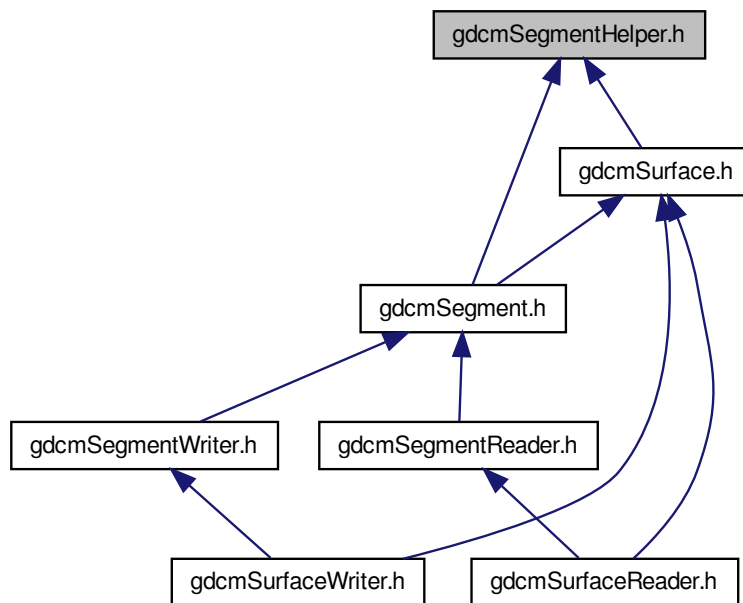
26.201 gdcmSegmentHelper.h File Reference

```
#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.

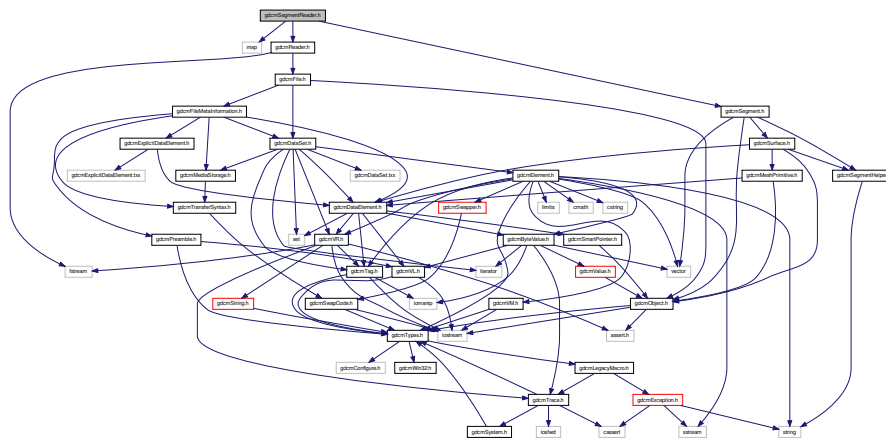
Namespaces

- `gdcm`
- `gdcm::SegmentHelper`

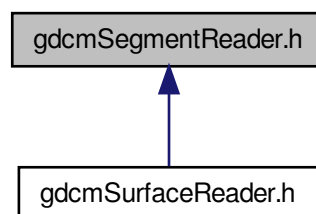
26.202 gdcmSegmentReader.h File Reference

```
#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`

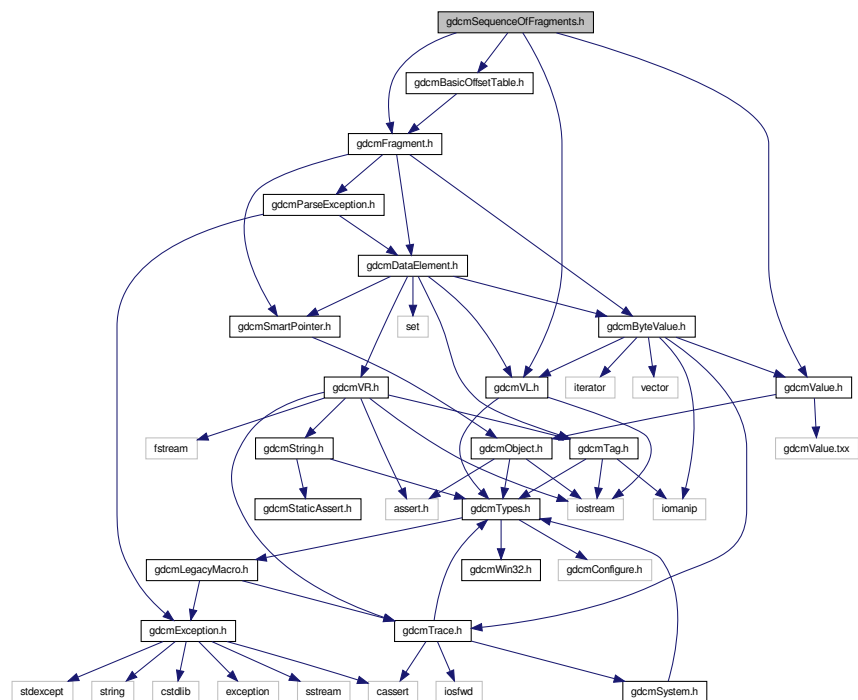
This class defines a segment reader. It reads attributes of group 0x0062.

Namespaces

- [gdcm](#)

26.204 gdcmSequenceOfFragments.h File Reference

```
#include "gdcmValue.h"
#include "gdcmVL.h"
#include "gdcmFragment.h"
#include "gdcmBasicOffsetTable.h"
Include dependency graph for gdcmSequenceOfFragments.h:
```



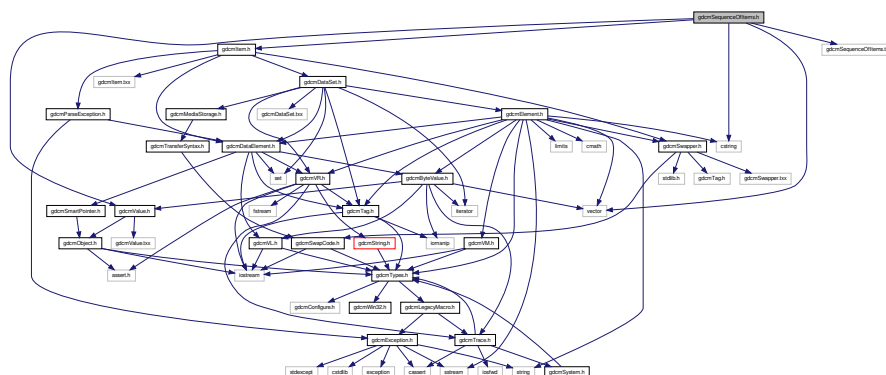
Classes

- class [gdcm::SequenceOfFragments](#)
Class to represent a Sequence Of Fragments.

Namespaces

- [gdcm](#)

```
#include "gdcValue.h"
#include "gdcItem.h"
#include <vector>
#include <cstring>
#include "gdcSequenceOfItems.txx"
Include dependency graph for gdcSequenceOfItems.h:
```



- class `gdcm::SequenceOfItems`

Class to represent a Sequence Of Items (value representation : SQ)

- **gdcm**

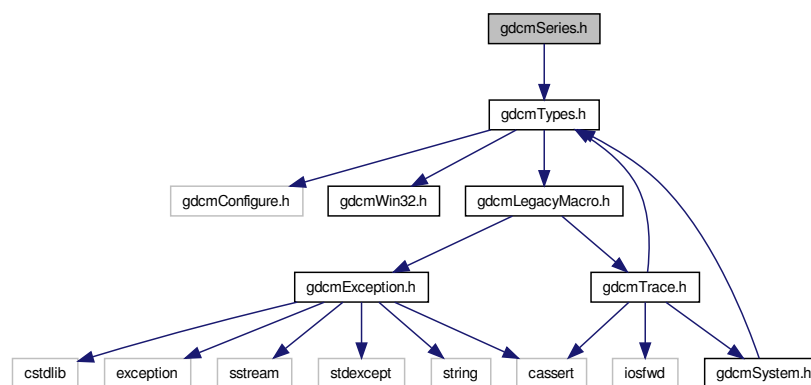
```
#include "gdcmTag.h"
#include "gdcmSmartPointer.h"
#include "gdcmFile.h"
#include <vector>
#include <string>
#include <map>
```


- enum `gdcm::LodModeType` {
`gdcm::LD_ALL = 0x00000000,`
`gdcm::LD_NOSEQ = 0x00000001,`
`gdcm::LD_NOSHADOW = 0x00000002,`
`gdcm::LD_NOSHADOWSEQ = 0x00000004 }`

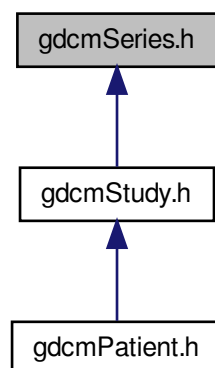
26.207 gdcmSeries.h File Reference

```
#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.

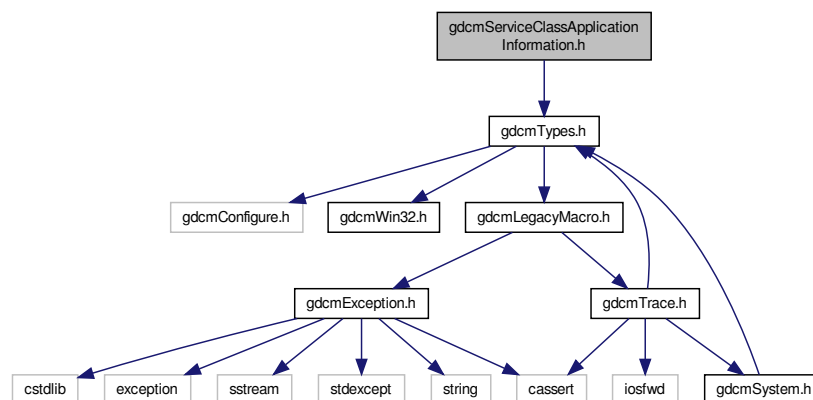
Namespaces

- [gdcm](#)

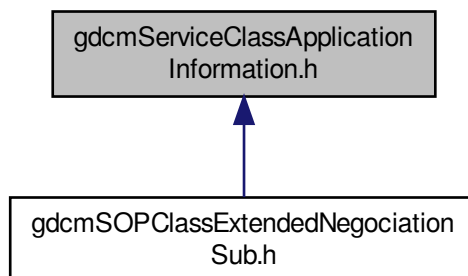
26.208 gdcmServiceClassApplicationInformation.h File Reference

```
#include "gdcmTypes.h"
```

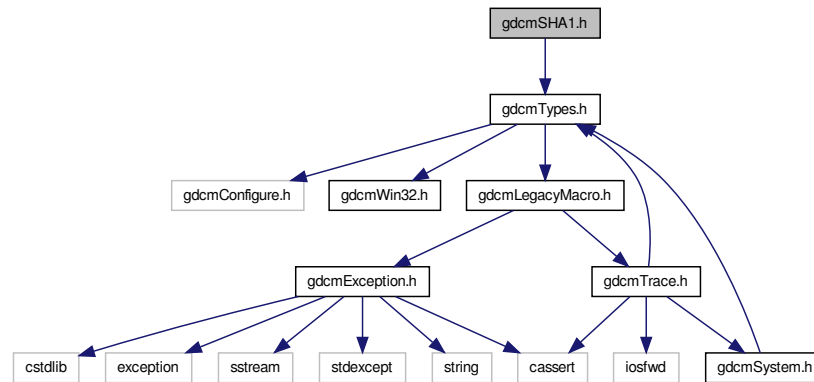
Include dependency graph for gdcmServiceClassApplicationInformation.h:



This graph shows which files directly or indirectly include this file:



Include dependency graph for `gdcmSHA1.h`:



Classes

- class `gdcm::SHA1`

Class for `SHA1`.

Namespaces

- `gdcm`

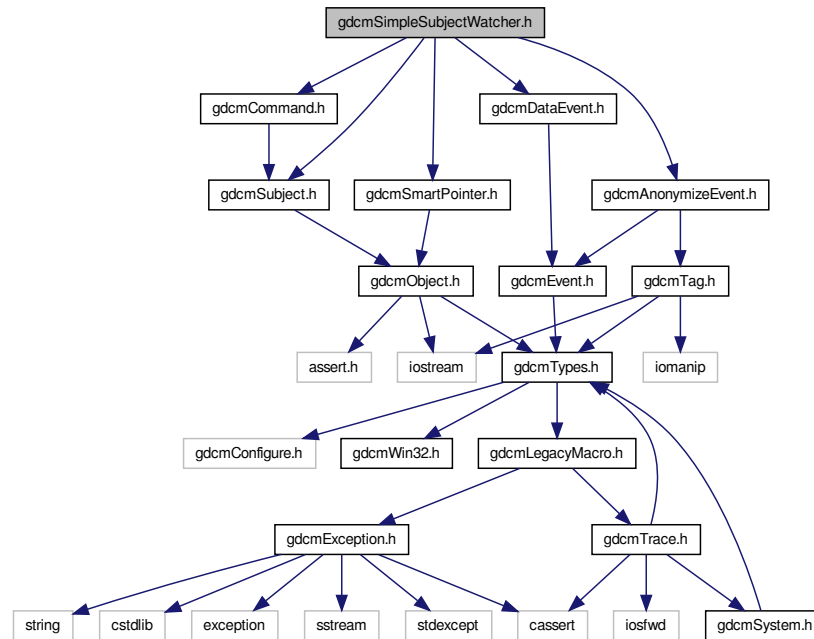
26.211 `gdcmSimpleSubjectWatcher.h` File Reference

```

#include "gdcmSubject.h"
#include "gdcmCommand.h"
#include "gdcmSmartPointer.h"
#include "gdcmAnonymizeEvent.h"
#include "gdcmDataEvent.h"

```

Include dependency graph for gdcSimpleSubjectWatcher.h:



Classes

- class [gdc::SimpleSubjectWatcher](#)

[SimpleSubjectWatcher](#) This is a typical [Subject](#) Watcher class. It will observe all events.

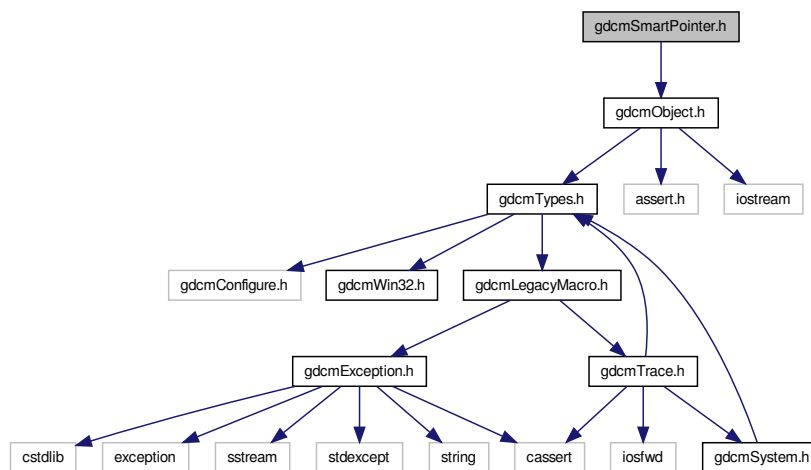
Namespaces

- [gdc](#)

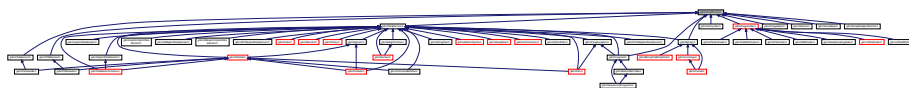
26.212 gdcSmartPointer.h File Reference

```
#include "gdcObject.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.

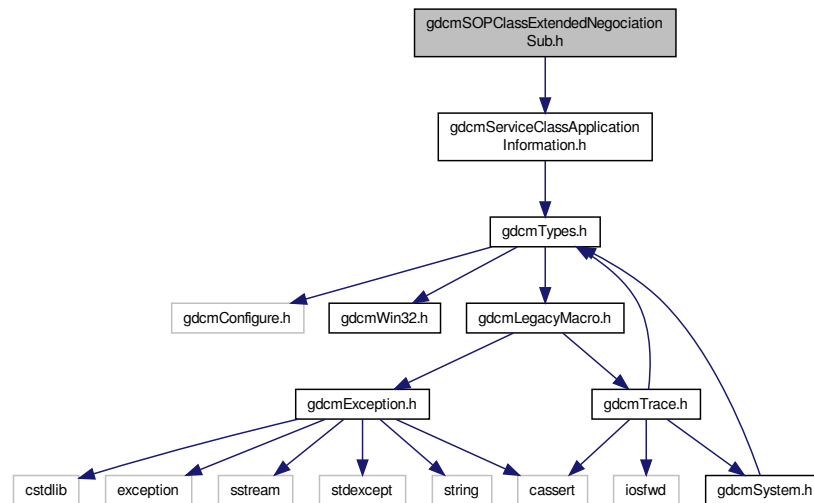
Namespaces

- `gdcm`

26.213 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)

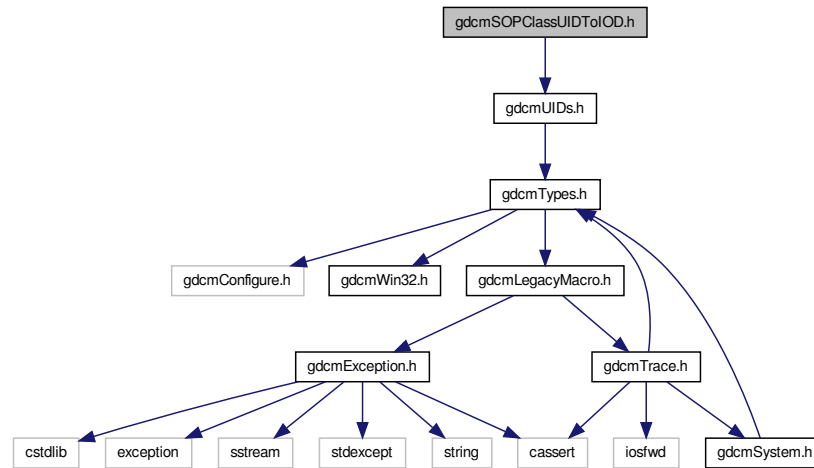
Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.214 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`

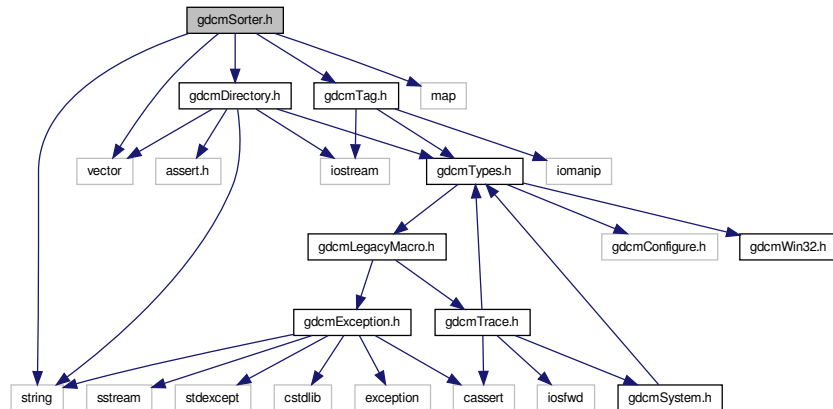
26.215 gdcmSorter.h File Reference

```

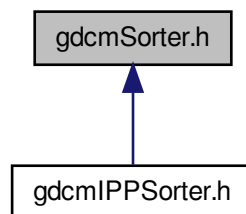
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
#include <map>

```

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](#).

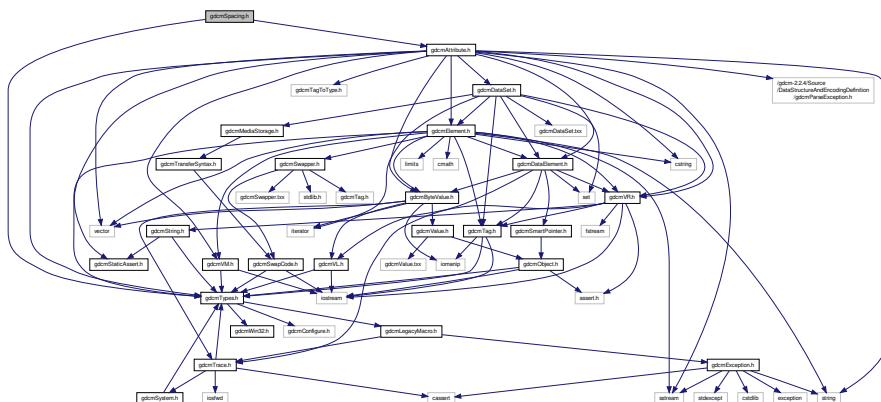
Namespaces

- [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)`

```
#include "gdcmTypes.h"
#include "gdcmAttribute.h"
Include dependency graph for gdcmSpacing.h:
```



- class `gdcm::Spacing`

Class for *Spacing*.

- **gdcm**

```
#include "gdcmFile.h"
```


- class `gdcm::Spectroscopy`
Spectroscopy class.

- **gdcm**

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```

[illegible]

Classes

- class [gdcm::SplitMosaicFilter](#)

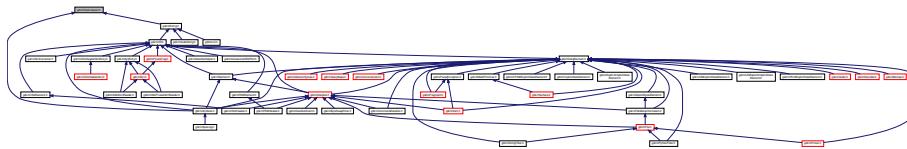
[SplitMosaicFilter](#) class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA [Image](#) Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.

Namespaces

- [gdcm](#)

26.219 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.

26.219.1 Macro Definition Documentation

26.219.1.1 `#define GDCM_DO_JOIN\(X, Y \) GDCM_DO_JOIN2\(X,Y\)`

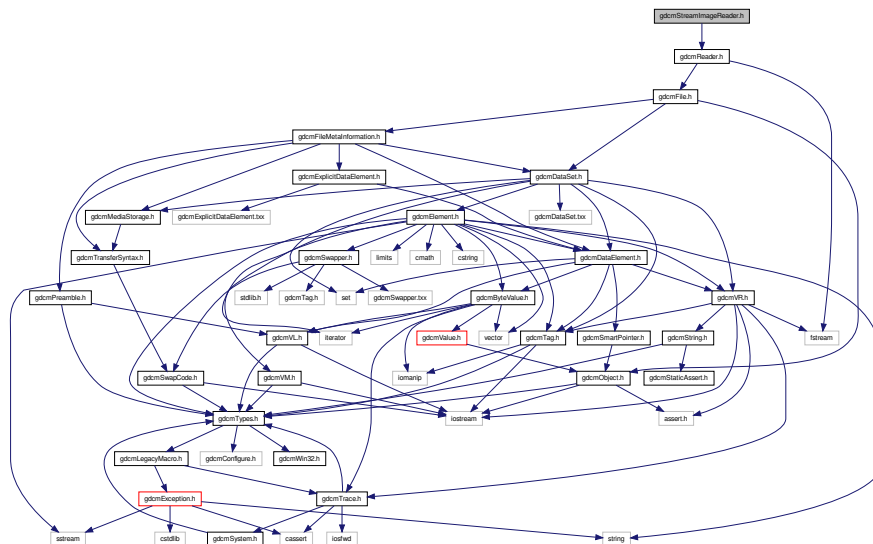
26.219.1.2 `#define GDCM_DO_JOIN2\(X, Y \) X##Y`

26.219.1.3 `#define GDCM_JOIN\(X, Y \) GDCM_DO_JOIN\(X, Y \)`

Value:

The GDCM_JOIN + **LINE** is needed to create a uniq identifier.

```
#include "gdcmReader.h"
Include dependency graph for gdcmStreamImageReader.h:
```



- class `gdcm::StreamImageReader`
StreamImageReader.

- **gdcm**

```
#include "gdcmWriter.h"
#include <iostream>
#include "gdcmDataSet.h"
```

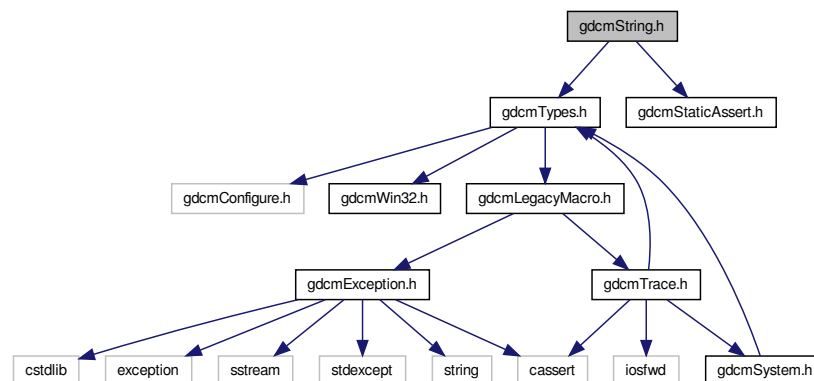


- StreamReader*.

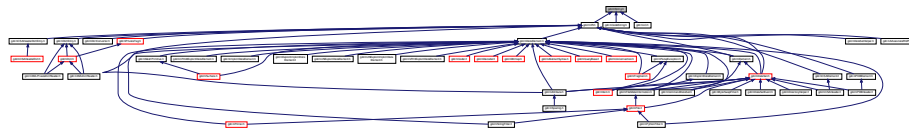
- **gdcm**

```
#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)

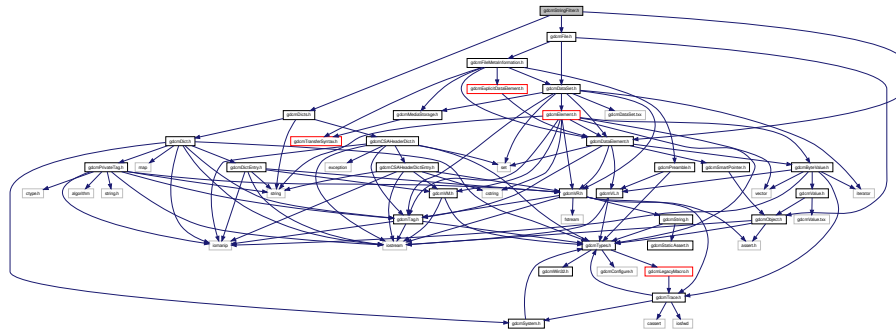
26.223 gdcmStringFilter.h File Reference

```

#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.

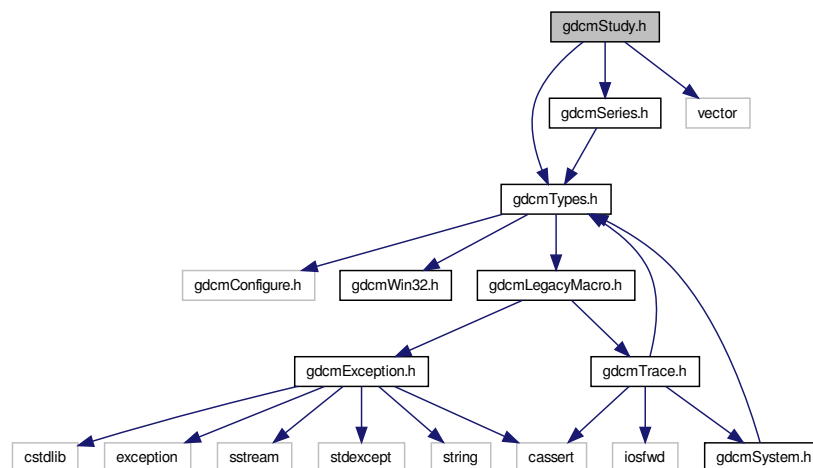
Namespaces

- `gdcm`

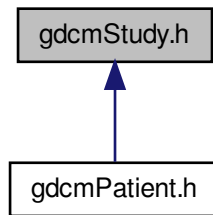
26.224 gdcmStudy.h File Reference

```
#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.

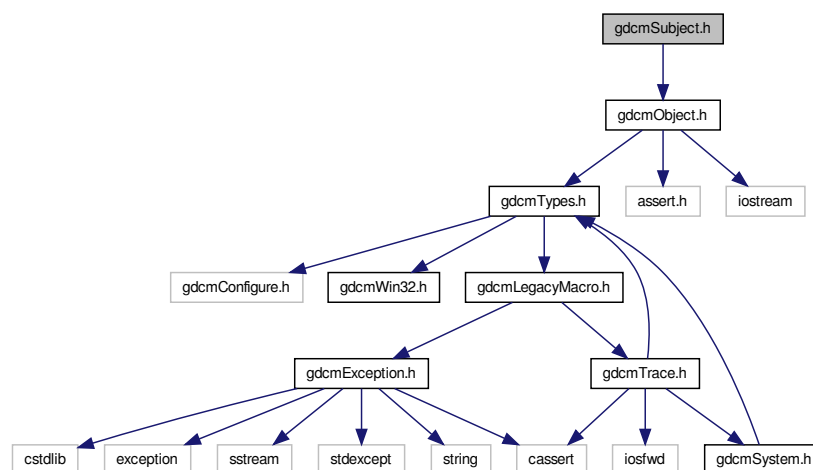
Namespaces

- `gdcm`

26.225 gdcmsubject.h File Reference

```
#include "gdcmObject.h"
```

Include dependency graph for `gdcmsubject.h`:



```

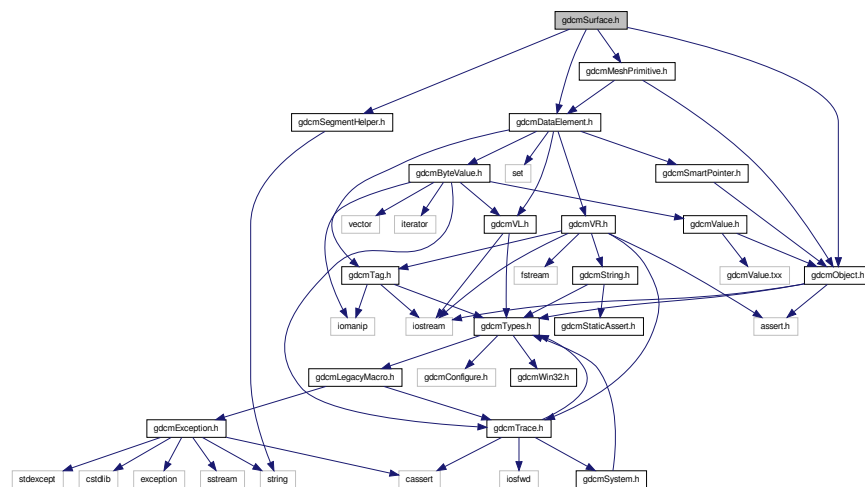
graph TD
    gdcmSubject_h[gdcmSubject.h]
    gdcmCommand_h[gdcmCommand.h]
    gdcmSimpleSubjectWatcher_h[gdcmSimpleSubjectWatcher.h]
    gdcmAnonymizer_h[gdcmAnonymizer.h]
    gdcmFileAnonymizer_h[gdcmFileAnonymizer.h]
    gdcmScanner_h[gdcmScanner.h]
    gdcmServiceClassUser_h[gdcmServiceClassUser.h]
    gdcmULConnectionManager_h[gdcmULConnectionManager.h]

    gdcmCommand_h --> gdcmSubject_h
    gdcmSimpleSubjectWatcher_h --> gdcmSubject_h
    gdcmAnonymizer_h --> gdcmSubject_h
    gdcmFileAnonymizer_h --> gdcmSubject_h
    gdcmScanner_h --> gdcmSubject_h
    gdcmServiceClassUser_h --> gdcmSubject_h
    gdcmULConnectionManager_h --> gdcmSubject_h
  
```

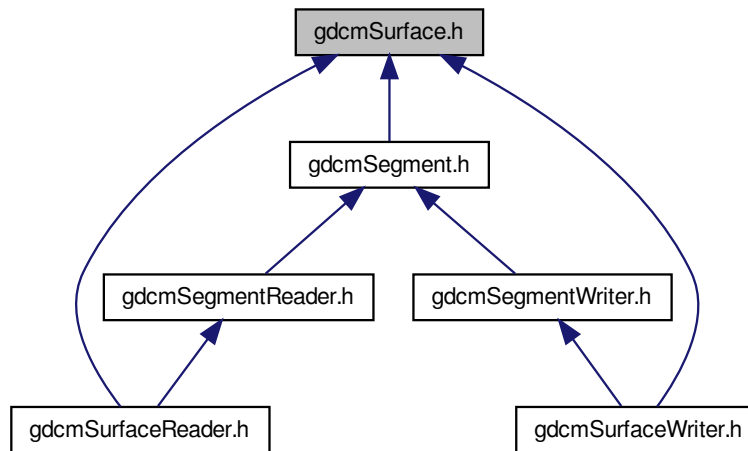
- class `gdcm::Subject`
Subject.

- **gdcm**

```
#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.

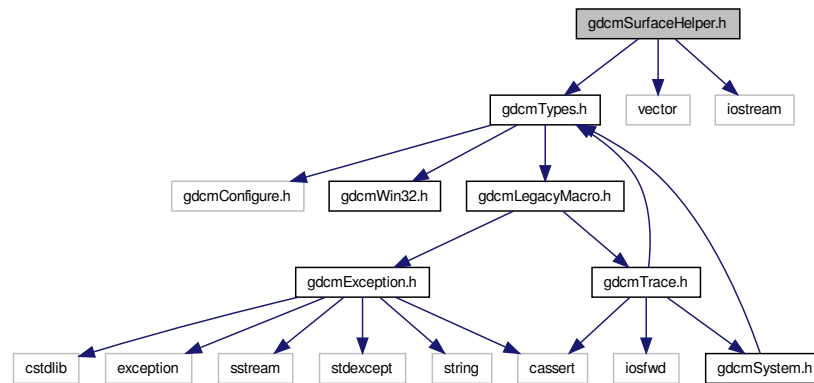
Namespaces

- [gdcm](#)

26.227 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](#)

26.228 gdcmSurfaceReader.h File Reference

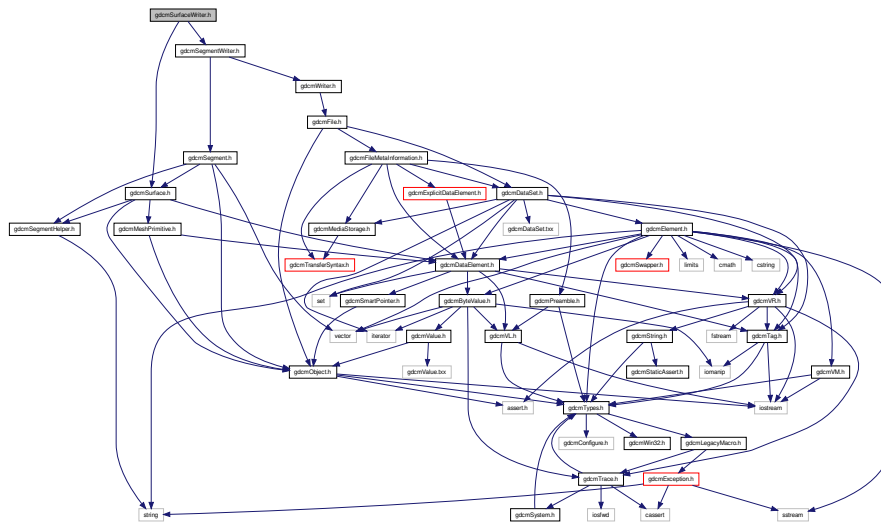
```
#include <gdcmSegmentReader.h>
#include <gdcmSurface.h>
```

- class `gdcm::SurfaceReader`

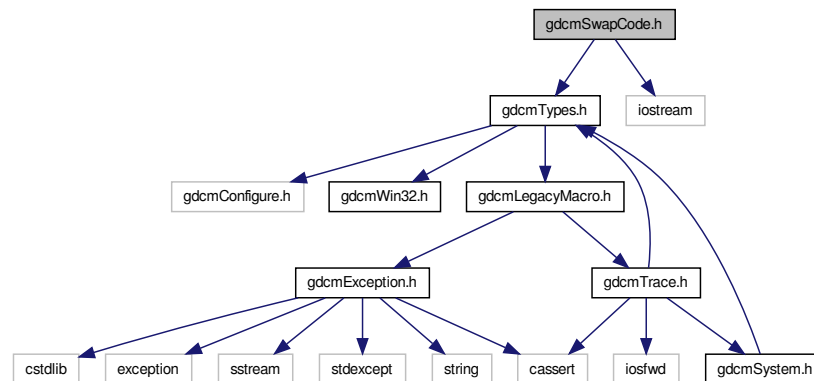
Namespaces

- **gdcm**

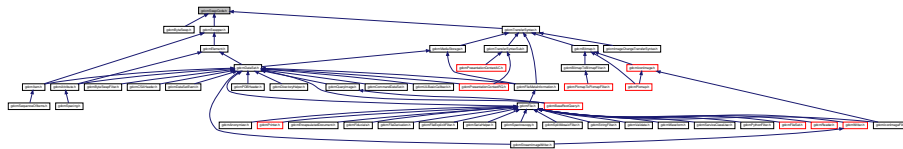
```
#include <gdcmSegmentWriter.h>
#include <gdcmSurface.h>
```



Include dependency graph for gdcmSwapCode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcm::SwapCode`
SwapCode representation.

Namespaces

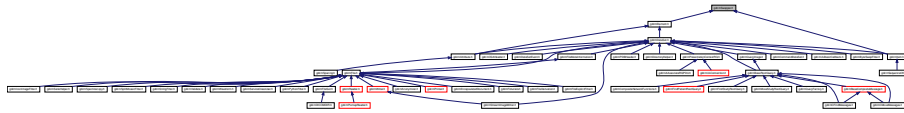
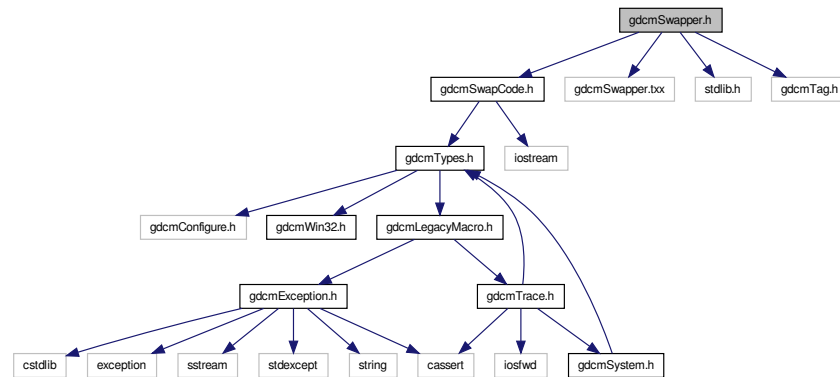
- `gdcm`

Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)`

26.231 gdcmSwapper.h File Reference

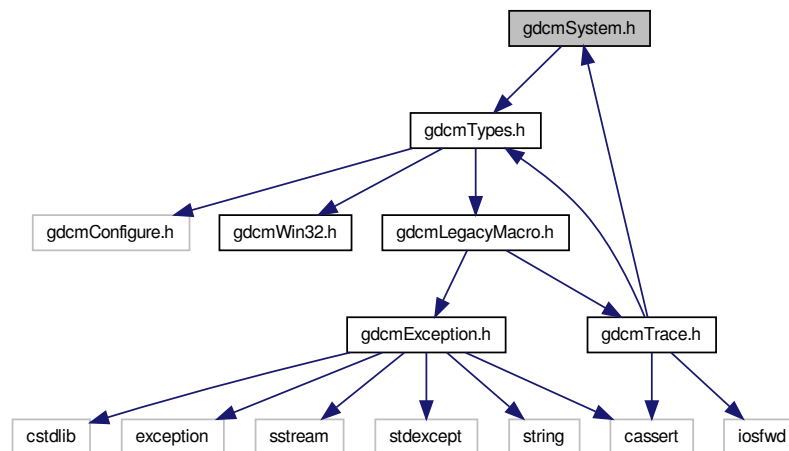
```
#include "gdcmSwapCode.h"
#include "gdcmSwapper.txx"
```



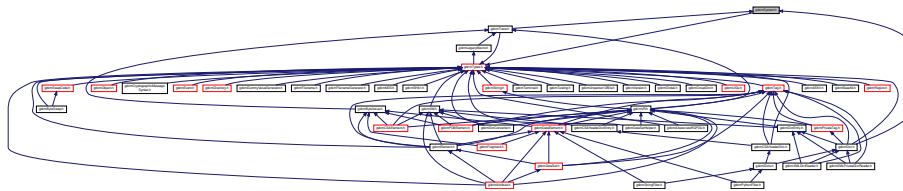
- class `gdc::SwapperDoOp`
- class `gdc::SwapperNoOp`

- **gdcm**

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`

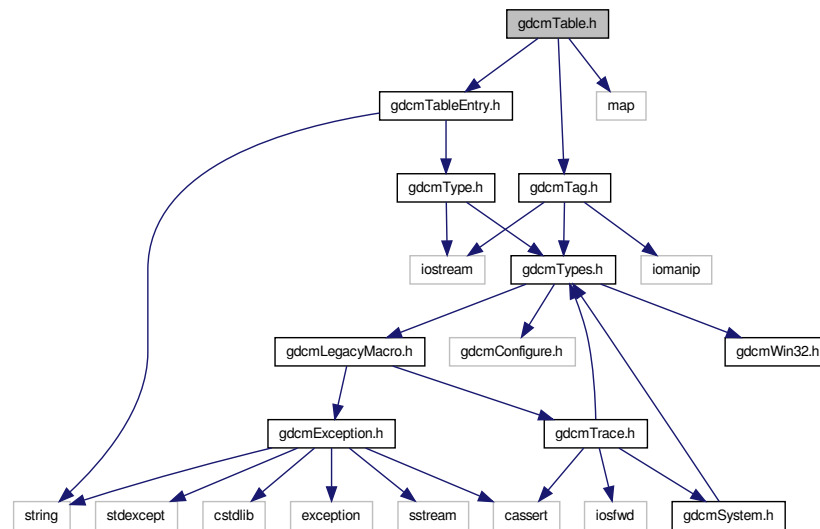
26.233 gdcmTable.h File Reference

```

#include "gdcmTableEntry.h"
#include "gdcmTag.h"
#include <map>

```

Include dependency graph for `gdcmTable.h`:



Classes

- class [gdcm::Table](#)

Table.

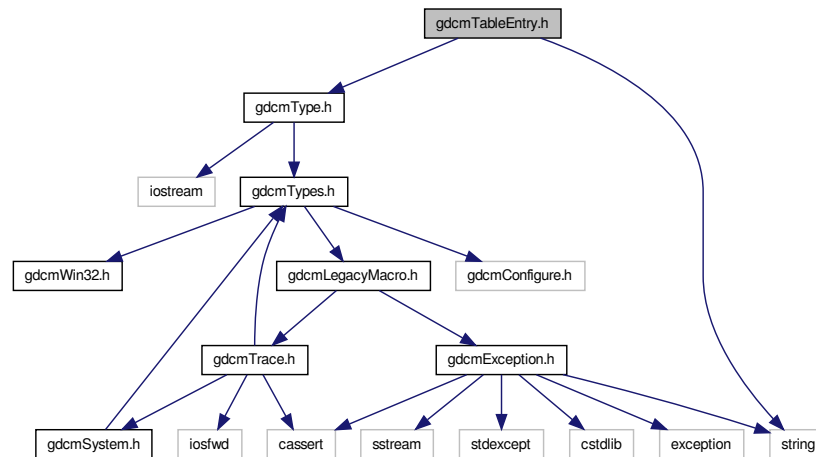
Namespaces

- [gdcm](#)

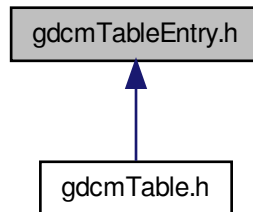
26.234 gdcmTableEntry.h File Reference

```
#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

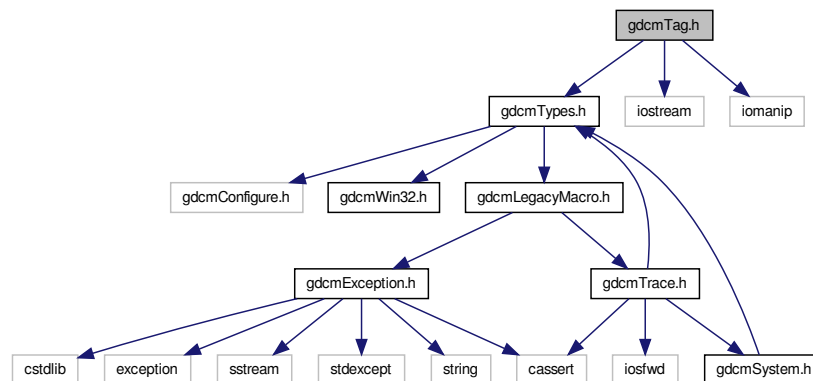
- [gdcm](#)

26.235 gdcmTableReader.h File Reference

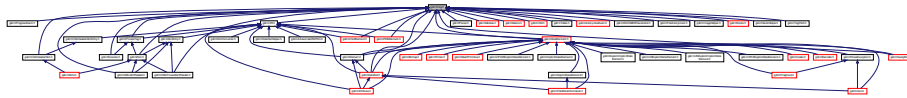
```
#include "gdcmTypes.h"
```


26.236 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

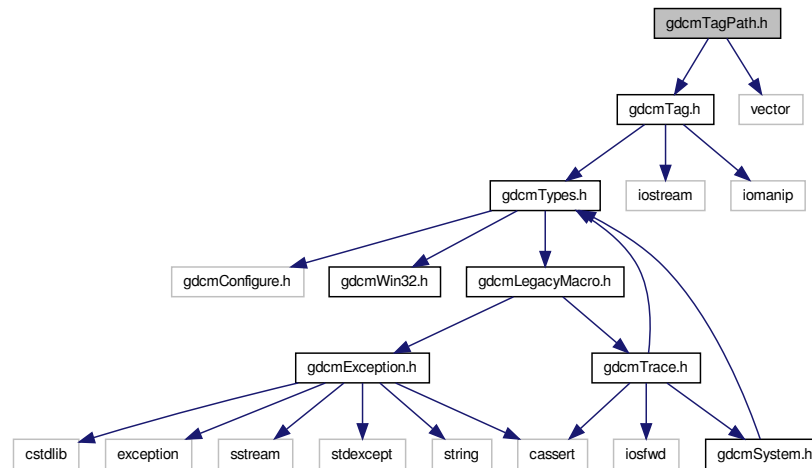
- `std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)`
- `std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)`

26.237 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](#)

26.238 gdcmTagToVR.h File Reference

Namespaces

- [gdcm](#)

Functions

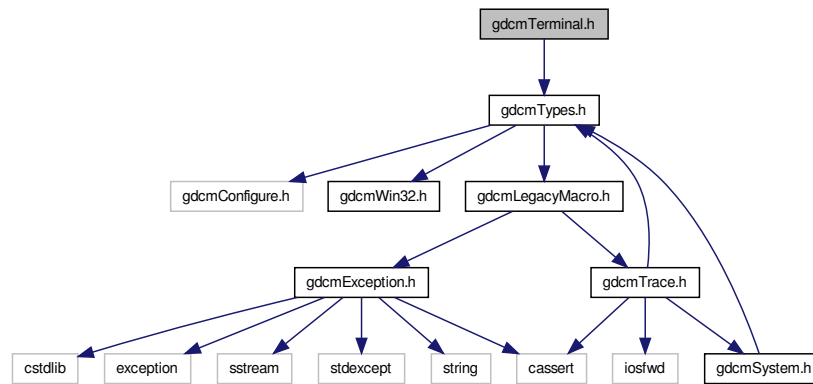
- VR::VRType [gdcm::GetVRFromTag](#) (Tag const &tag)

26.239 gdcmtar.man File Reference

26.240 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

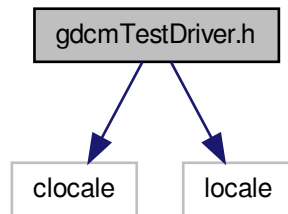
- `GDCM_EXPORT` `std::string gdc::terminal::setattribute` (Attribute att)
- `GDCM_EXPORT` `std::string gdc::terminal::setbgcolor` (Color c)
- `GDCM_EXPORT` `std::string gdc::terminal::setfgcolor` (Color c)
- `GDCM_EXPORT` `void gdc::terminal::setmode` (Mode m)

26.241 gdcmTestDriver.h File Reference

```
#include <clocale>
```

```
#include <locale>
```

Include dependency graph for gdcmTestDriver.h:

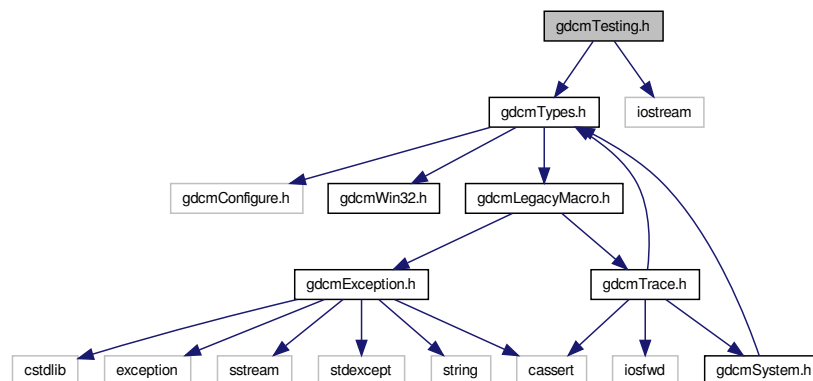


26.242 gdcmTesting.h File Reference

```
#include "gdcTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmTesting.h:



Classes

- class [gdcm::Testing](#)
class for testing

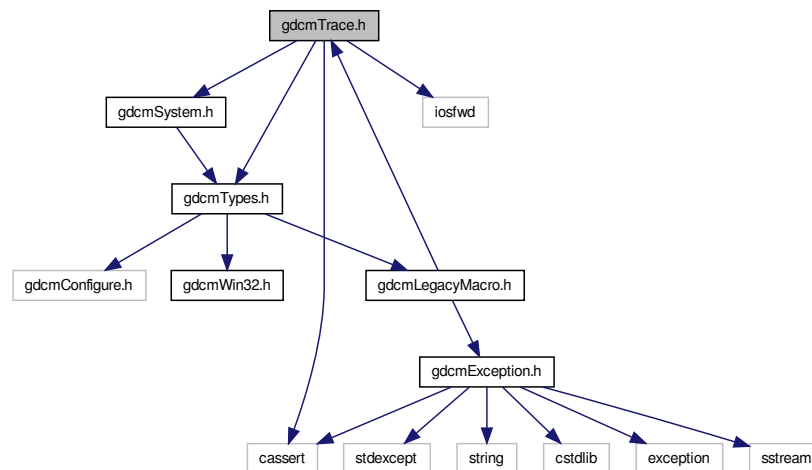
Namespaces

- [gdcm](#)

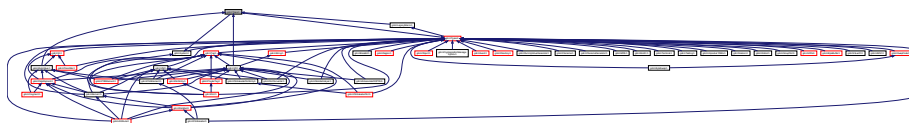
26.243 gdcmTrace.h File Reference

```
#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.

26.243.1 Macro Definition Documentation

26.243.1.1 #define [GDCM_FUNCTION](#) "<unknow>"

26.243.1.2 #define [gdcmAssertAlwaysMacro](#)(arg) [gdcmAssertMacro](#)(arg)

[AssertAlways.](#)

Parameters

<i>arg</i>	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\(\)](#).

26.243.1.3 #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

<i>arg</i>	argument to test An easy solution to pass also a message is to do: <code>gdcmAssertMacro("my message" && 2 < 3)</code>
------------	---

Referenced by `gdcm::PixelFormat::SetSamplesPerPixel()`.

26.243.1.4 `#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

<i>msg</i>	message part
------------	--------------

Referenced by `gdcm::ByteValue::ByteValue()`, `gdcm::SequenceOfItems::Read()`, `gdcm::Item::Read()`, `gdcm::VR::Read()`, `gdcm::SequenceOfFragments::ReadPreValue()`, `gdcm::SequenceOfFragments::ReadValue()`, and `gdcm::ByteValue::SetLength()`.

26.243.1.5 `#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

<i>msg</i>	second message part
------------	---------------------

Referenced by `gdcm::CommandDataSet::Insert()`, `gdcm::FileMetaInformation::Insert()`, `gdcm::DataSet::Insert()`, `gdcm::Item::Read()`, and `gdcm::Fragment::ReadBacktrack()`.

26.243.1.6 `#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

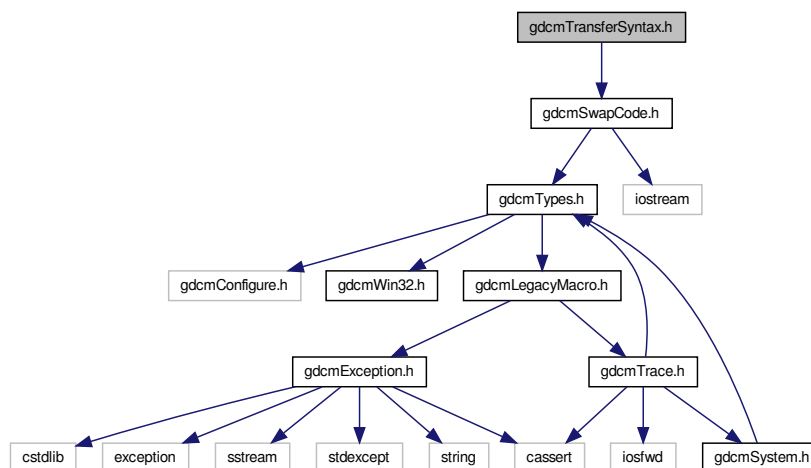
<i>msg</i>	message part
------------	--------------

Referenced by `gdcM::DataSet::InsertDataElement()`, `gdcM::SequenceOfItems::Read()`, `gdcM::Item::Read()`, `gdcM::Fragment::ReadBacktrack()`, `gdcM::Fragment::ReadValue()`, `gdcM::SequenceOfFragments::ReadValue()`, and `gdcM::Item::Write()`.

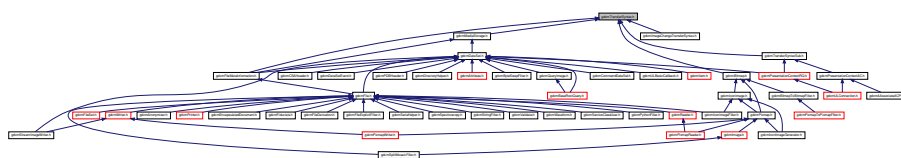
26.244 gdcMTransferSyntax.h File Reference

```
#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

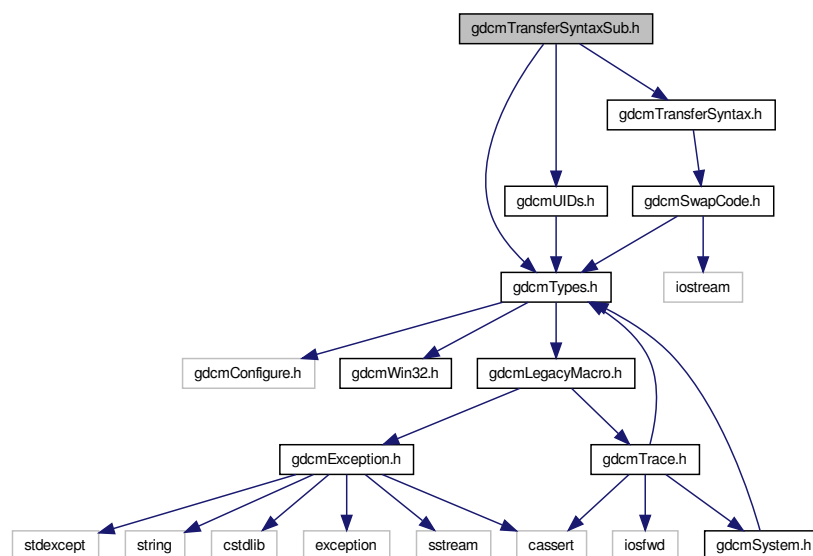
- [gdcm](#)

Functions

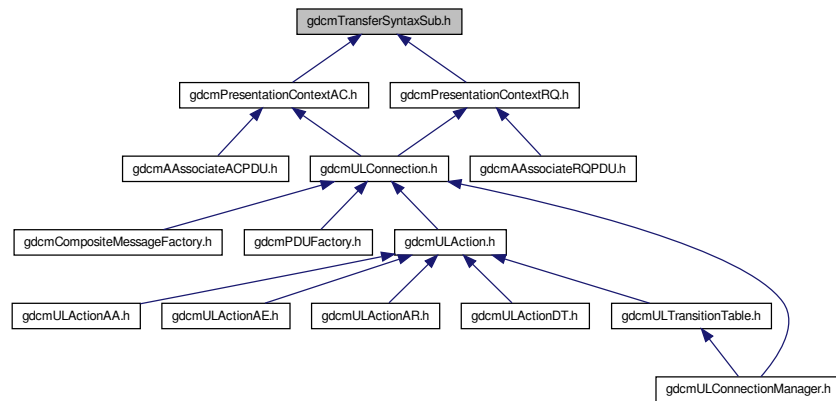
- `std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)`

26.245 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.

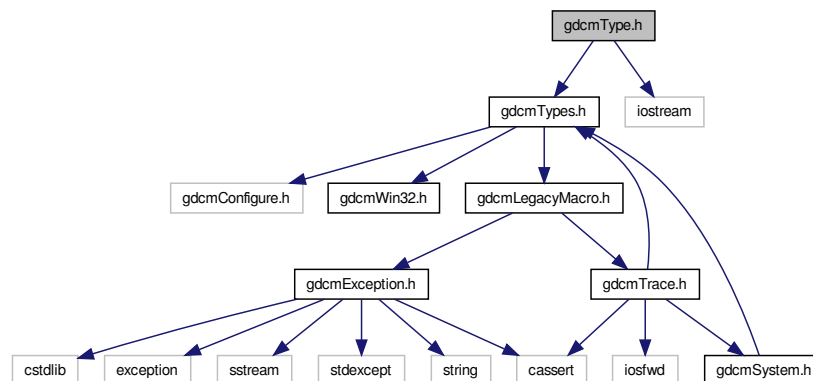
Namespaces

- [gdcm](#)
- [gdcm::network](#)

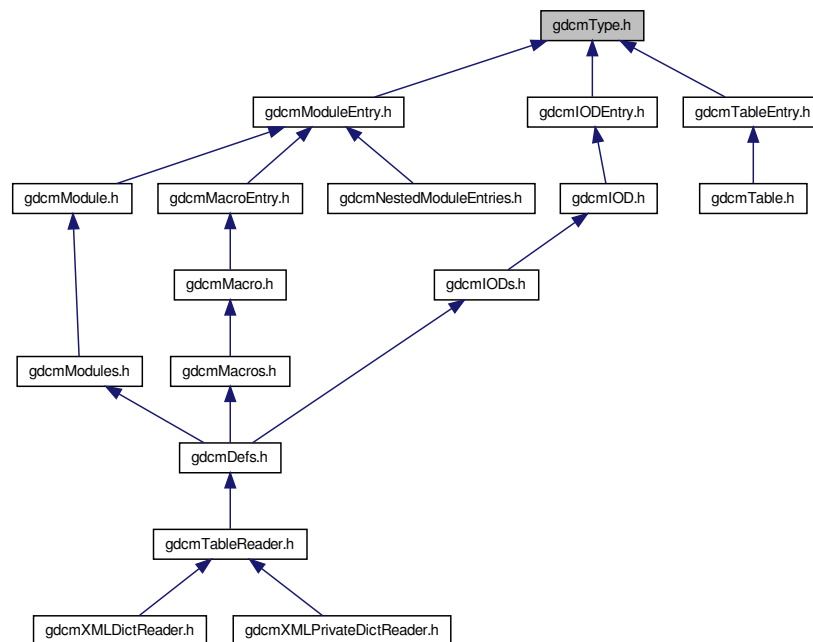
26.246 gdcmType.h File Reference

```
#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

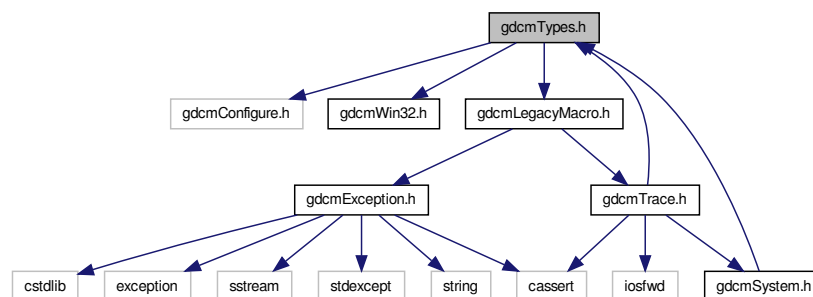
- [gdcm](#)

Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)`

26.247 gdcmTypes.h File Reference

```
#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
Include dependency graph for gdcmTypes.h:
```



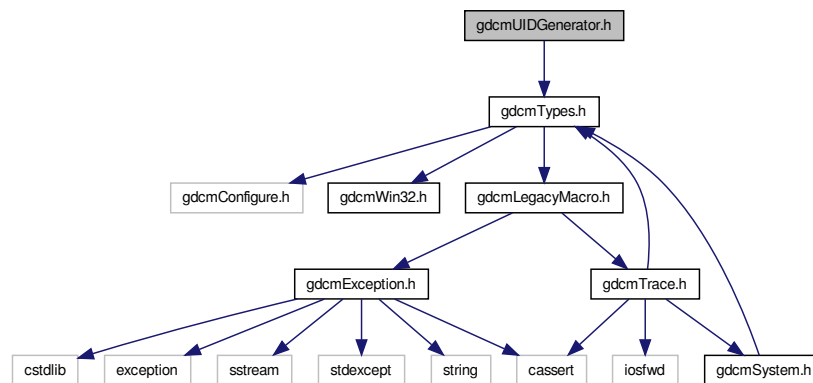
This graph shows which files directly or indirectly include this file:



26.248 gdcmUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUIDGenerator.h:



Classes

- class [gdcm::UIDGenerator](#)
Class for generating unique UID.

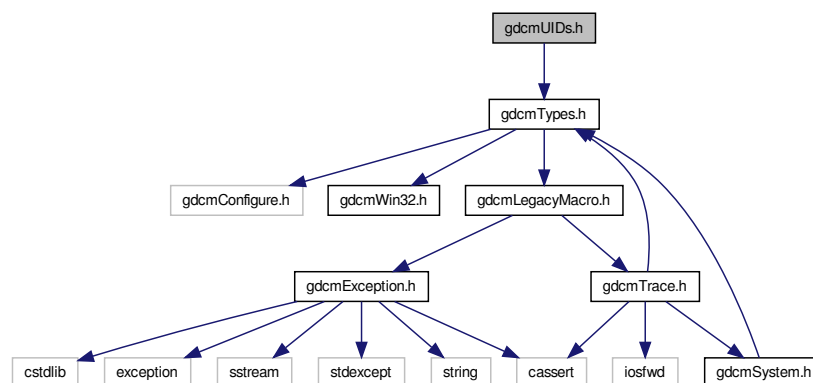
Namespaces

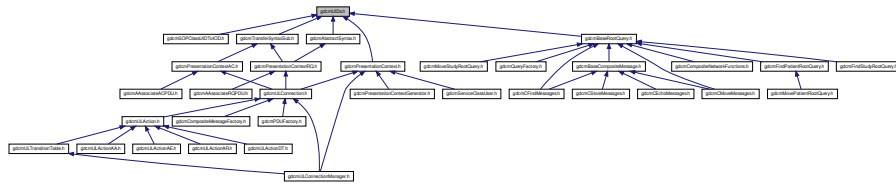
- [gdcm](#)

26.249 gdcmUIDs.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUIDs.h:



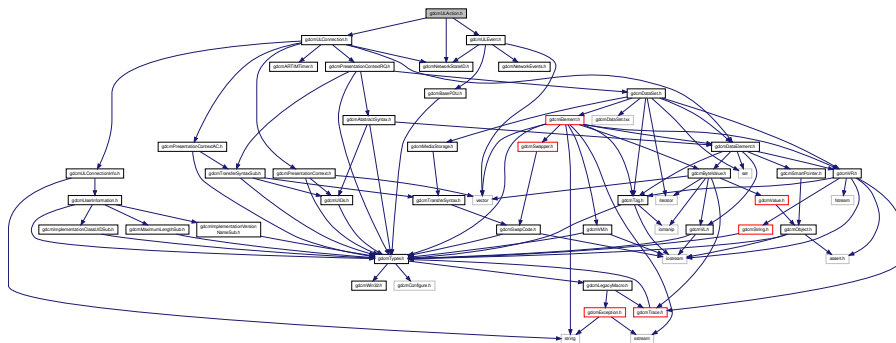


- class `gdcm::UIDs`
all known uids

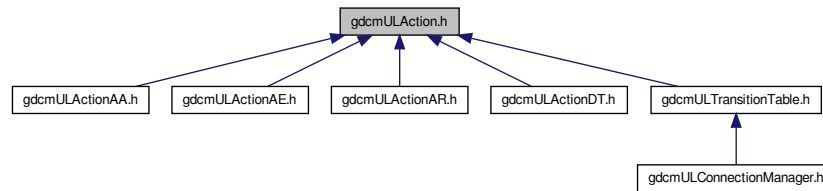
- **gdc**

- `std::ostream & gdcmm::operator<< (std::ostream &_os, const UIDs &uid)`

```
#include "gdcNetworkStateID.h"
#include "gdcULEvent.h"
#include "gdcULConnection.h"
Include dependency graph for gdcULAction.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*.

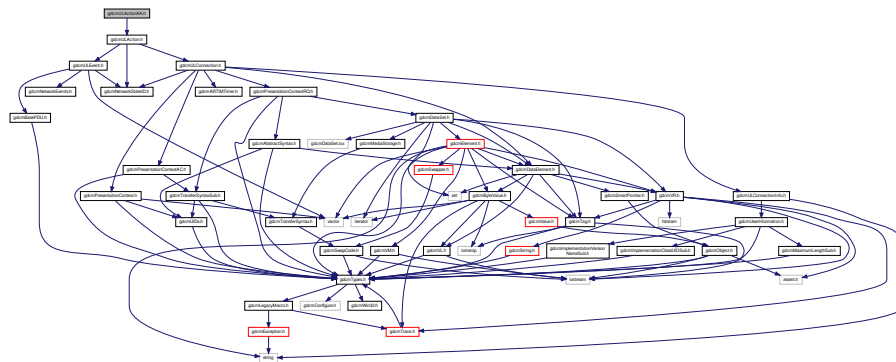
Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.251 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](#)

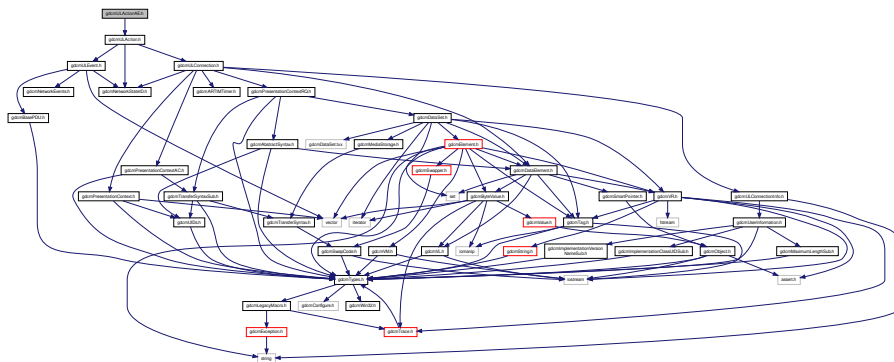
Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.252 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](#)

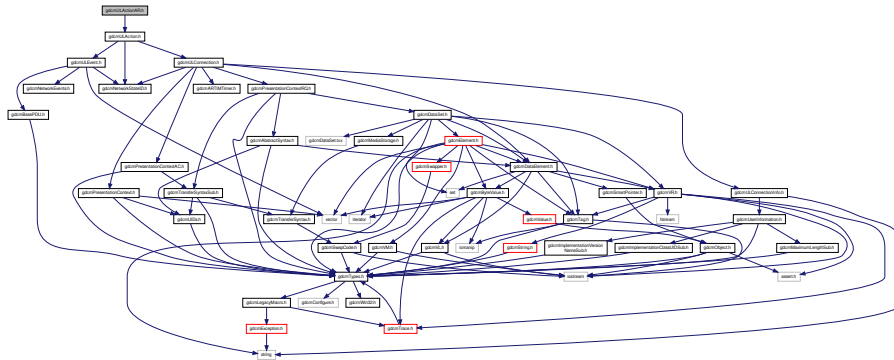
Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.253 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](#)

26.254 gdcmULActionDT.h File Reference

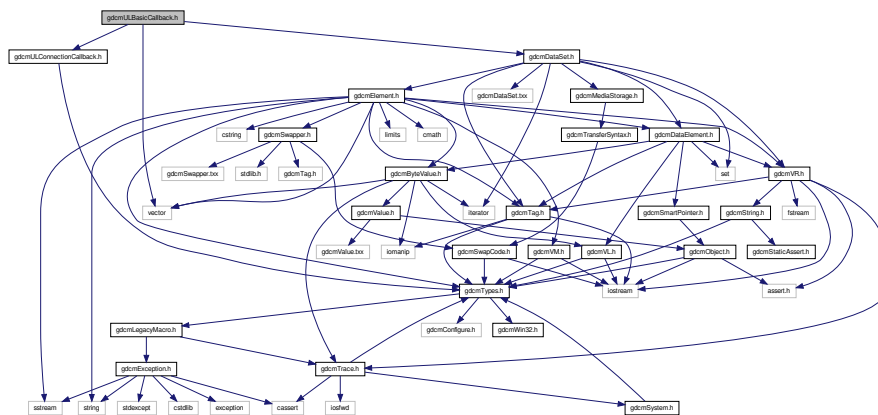
```
#include "gdcmULAction.h"
```

[illegible]

- class `gdcm::network::ULActionDT1`
- class `gdcm::network::ULActionDT2`

- `gdc`
- `gdc::network`

```
#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*.

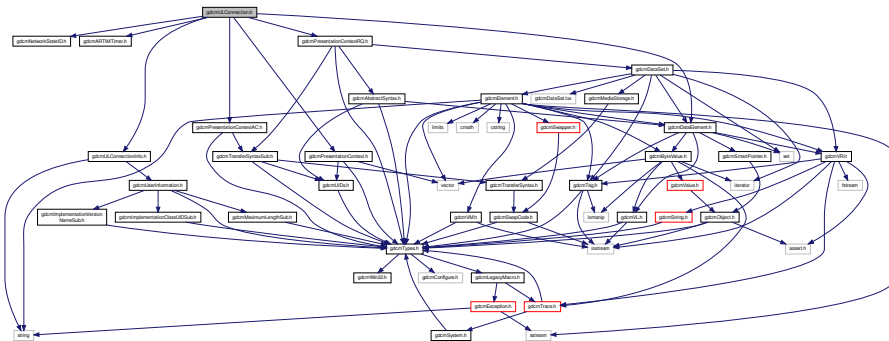
Namespaces

- [gdcm](#)
- [gdcm::network](#)

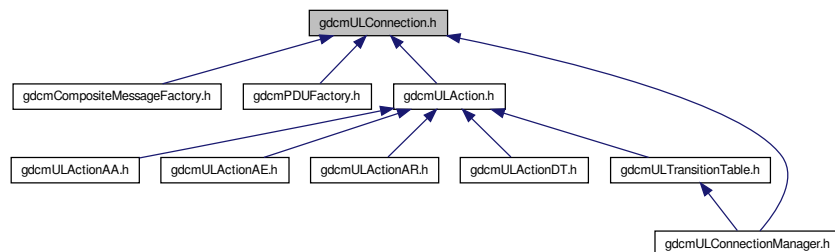
26.256 gdcmULConnection.h File Reference

```
#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.

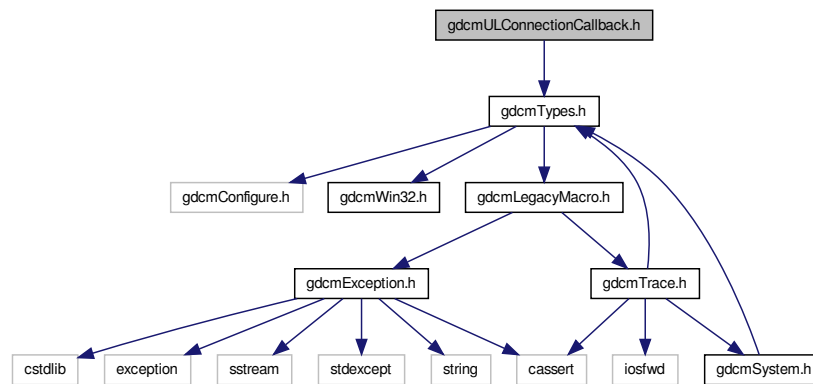
Namespaces

- [gdcm](#)
- [gdcm::network](#)

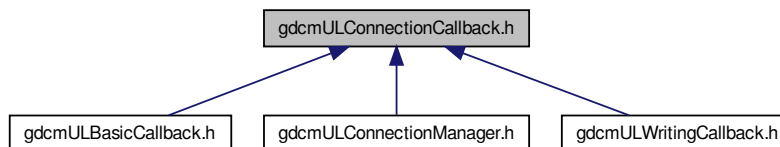
26.257 gdcmULConnectionCallback.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



Classes

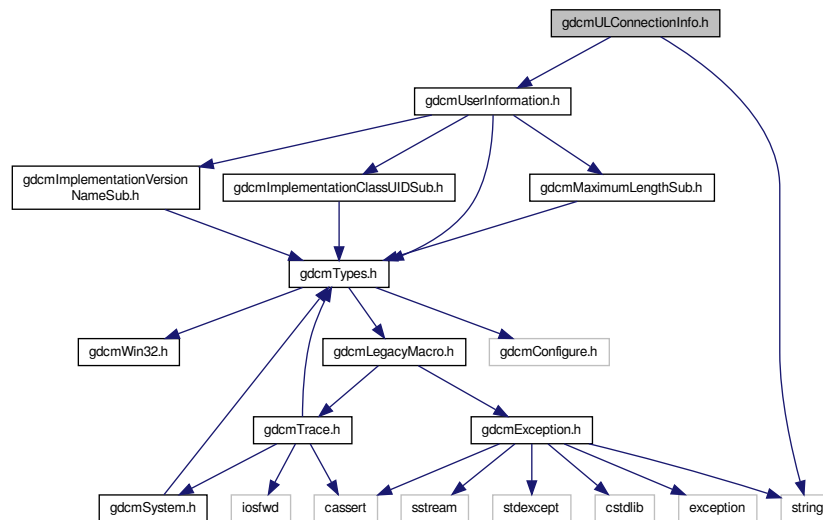
- class [gdcm::network::ULConnectionCallback](#)

Namespaces

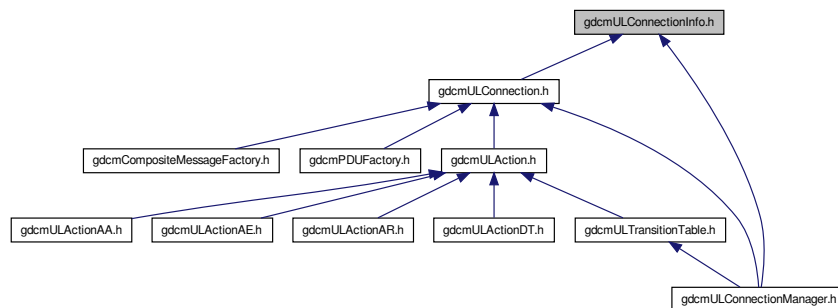
- [gdcm](#)
- [gdcm::network](#)

26.258 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:



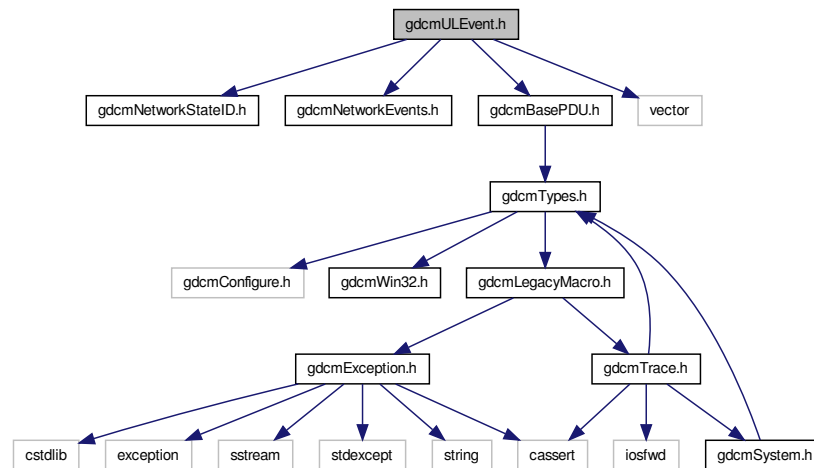
Classes

- class [gdcm::network::ULConnectionInfo](#)

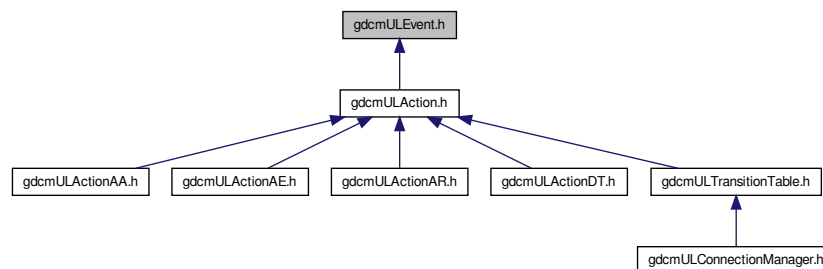
26.260 gdcmULEvent.h File Reference

```
#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

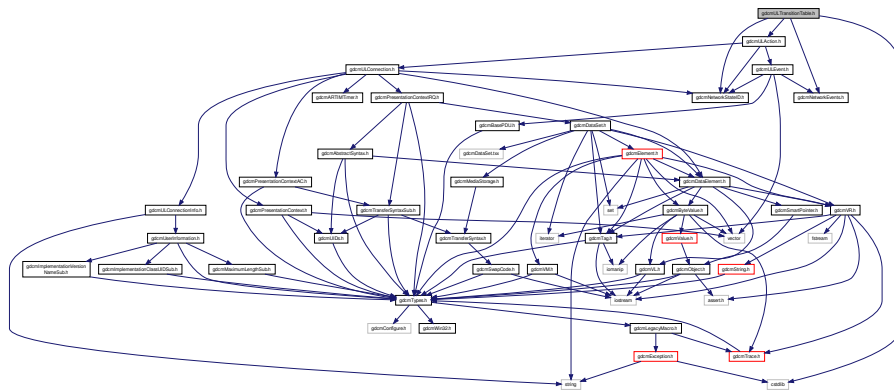
- [gdcm](#)

- [gdcm::network](#)

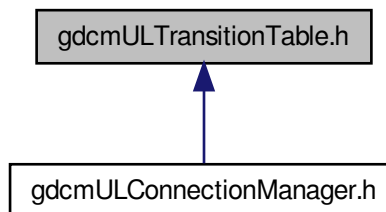
26.261 gdcmULTransitionTable.h File Reference

```
#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](#)

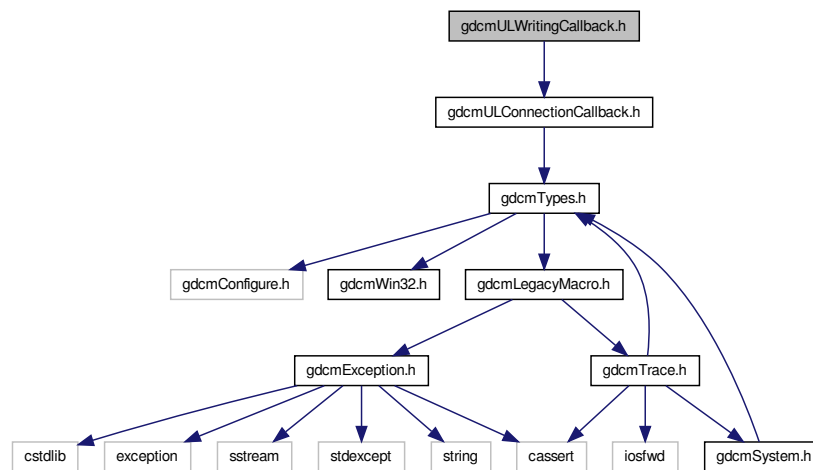
ULTransitionTable The transition table of all the ULEvents, new ULActions, and ULStates.

Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.262 gdcmULWritingCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
Include dependency graph for gdcmULWritingCallback.h:
```



Classes

- class [gdcm::network::ULWritingCallback](#)

Namespaces

- [gdcm](#)
- [gdcm::network](#)

26.263 gdcmUNExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmUNExplicitDataElement.txx"
```


Class to read/write a [DataElement](#) as Explicit/Implicit Data [Element](#) This class gather two known bugs:

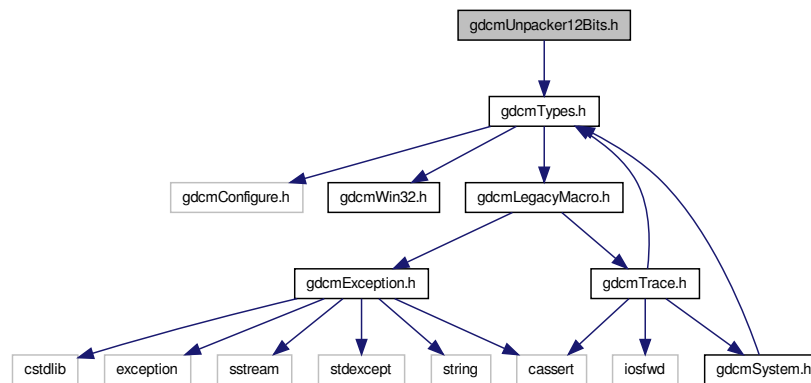
Namespaces

- [gdcm](#)

26.265 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

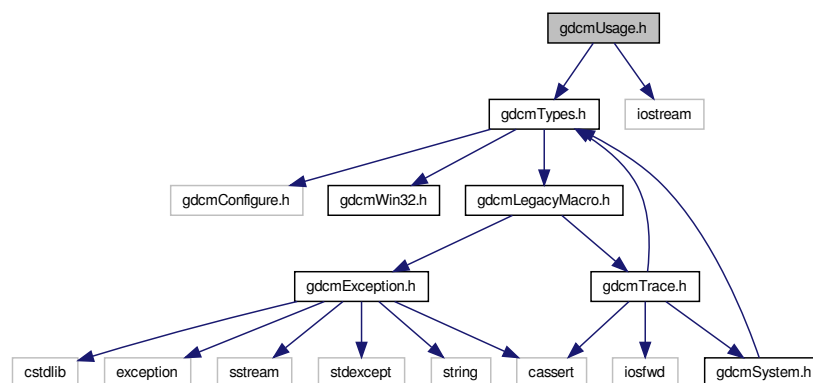
- [gdcm](#)

26.266 gdcmUsage.h File Reference

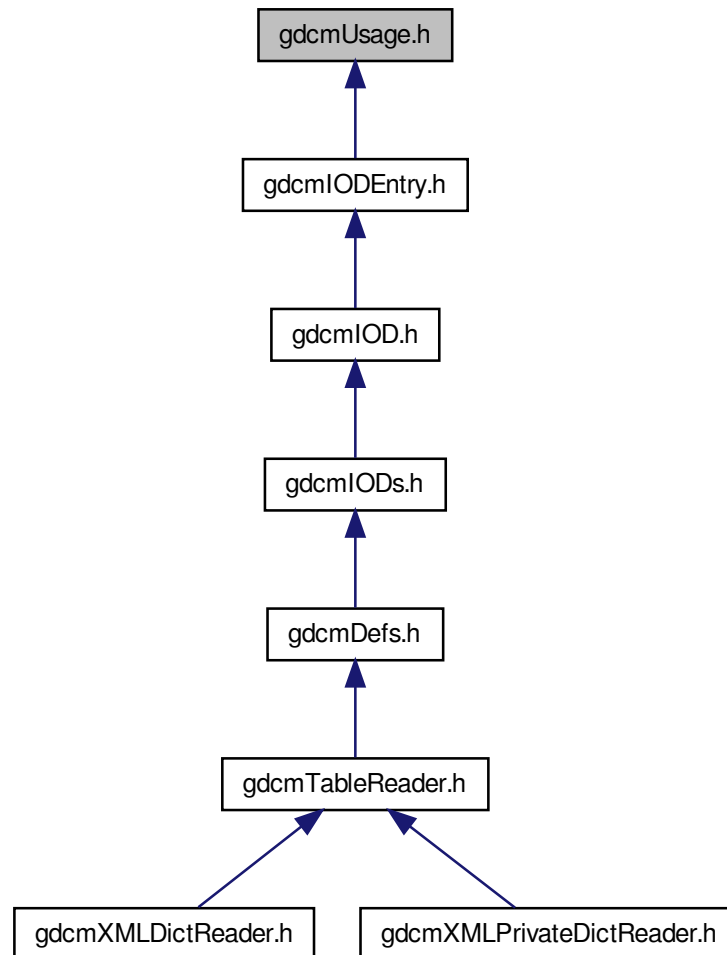
```
#include "gdcmTypes.h"
```

```
#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`

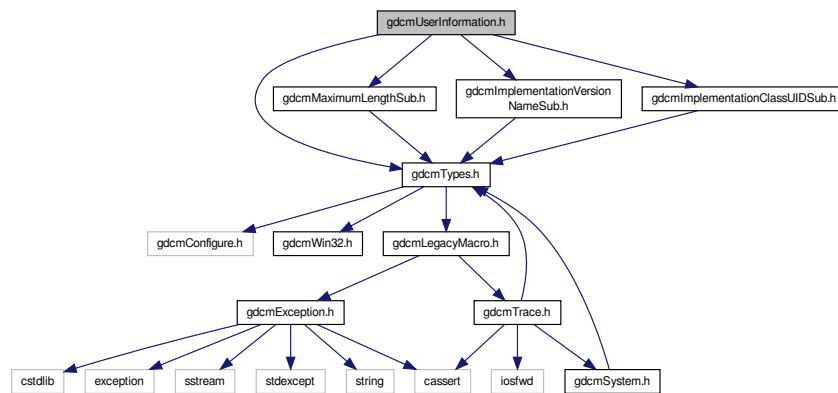
Functions

- `std::ostream & gdcmm::operator<< (std::ostream &_os, const Usage &val)`

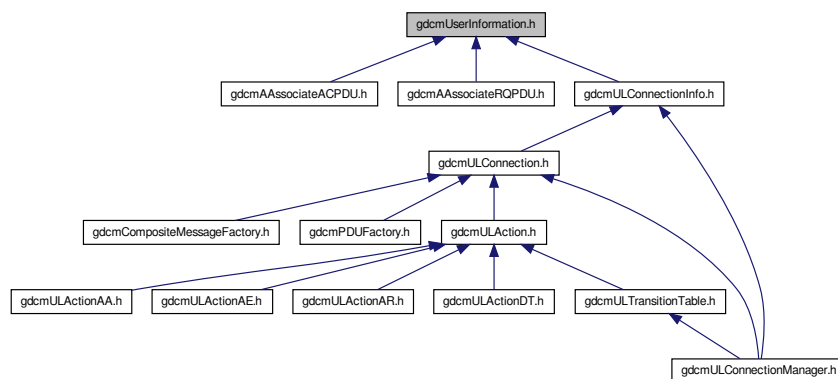
26.267 gdcmmUserInformation.h File Reference

```
#include "gdcmmTypes.h"
#include "gdcmmMaximumLengthSub.h"
#include "gdcmmImplementationVersionNameSub.h"
#include "gdcmmImplementationClassUIDSub.h"
```

Include dependency graph for `gdcmmUserInformation.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `gdcmm::network::UserInformation`

UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.

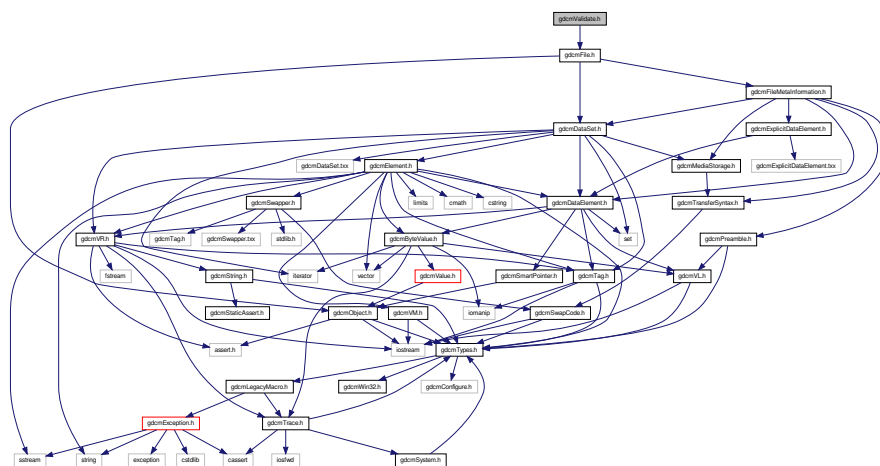
Namespaces

- `gdcm`
- `gdcm::network`

26.268 gdcmValidate.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for `gdcmValidate.h`:



Classes

- class `gdcm::Validate`
Validate class.

Namespaces

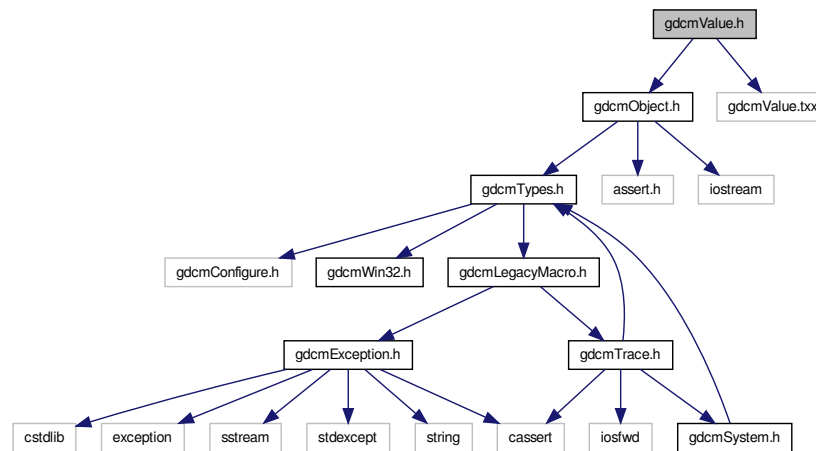
- **gdc**

26.269 gdcmValue.h File Reference

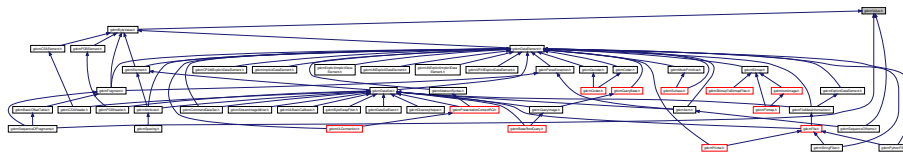
```
#include "gdcmObject.h"
```

```
#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](#).

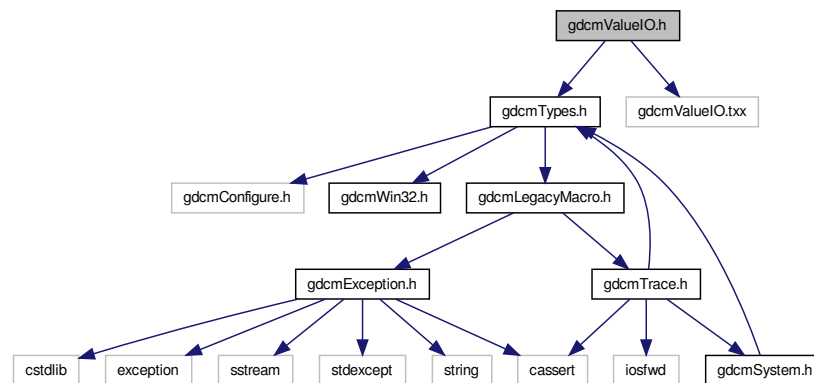
Namespaces

- `gdcm`

26.270 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

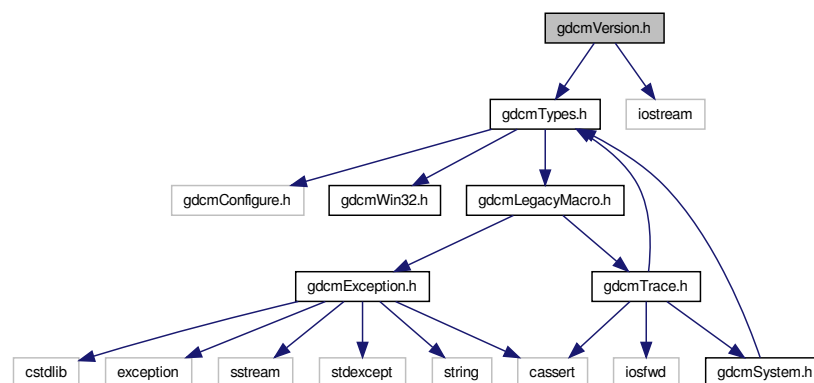
- [gdcm](#)

26.271 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)`

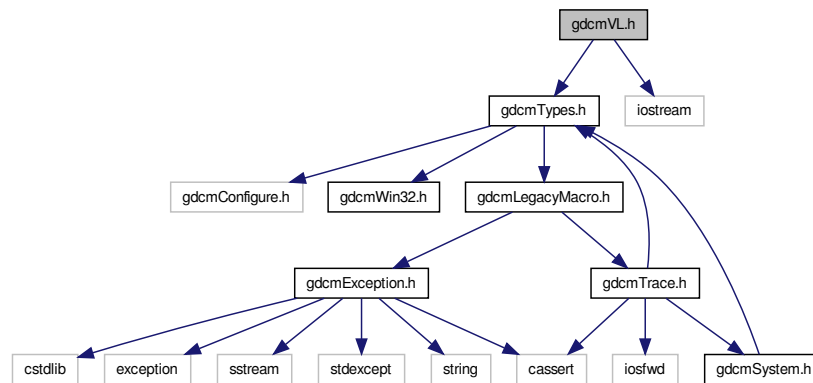
26.272 gdcmviewer.man File Reference

26.273 gdcmVL.h File Reference

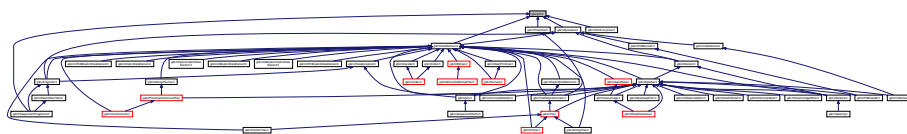
```
#include "gdcmTypes.h"
```

```
#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

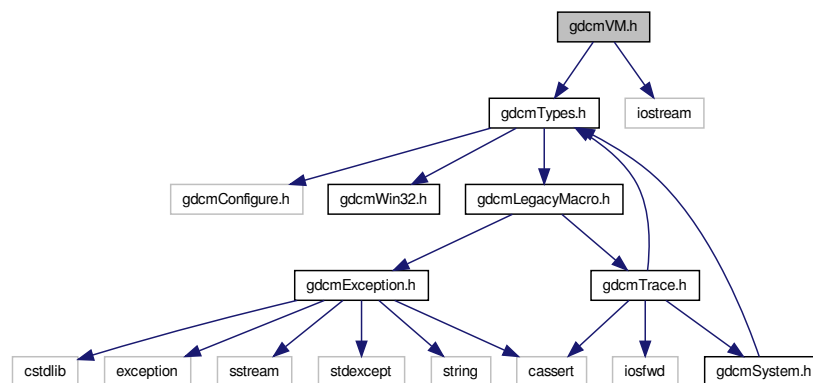
- `std::ostream & gdcM::operator<< (std::ostream &os, const VL &val)`

26.274 gdcVM.h File Reference

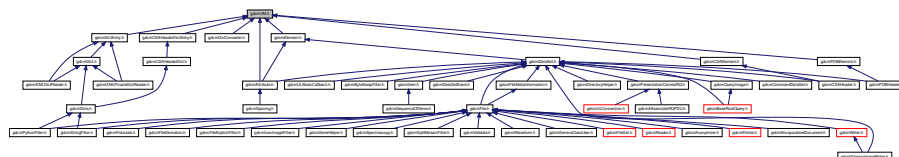
```
#include "gdcTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcVM.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)`

26.274.1 Macro Definition Documentation

26.274.1.1 `#define TYPETOLENGTH(type, length)`

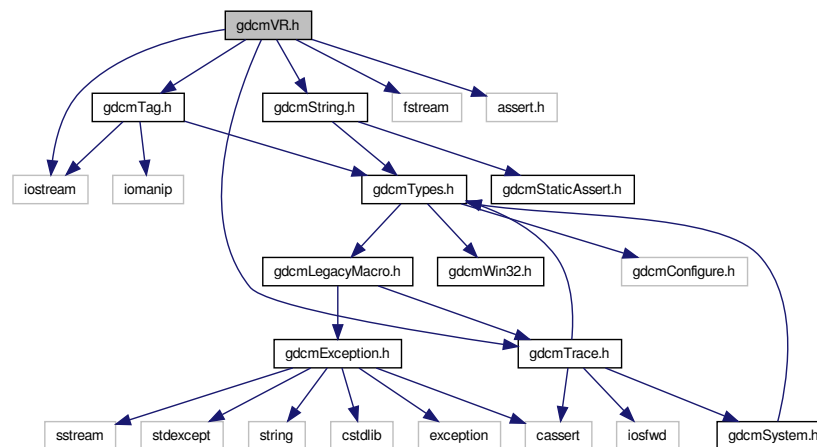
Value:

```
template<> struct VMToLength<VM::type> \
{ enum { Length = length }; };
```

26.275 gdcmVR.h File Reference

```
#include "gdcmTag.h"
#include "gdcmTrace.h"
#include "gdcmString.h"
#include <iostream>
#include <fstream>
#include <assert.h>
```

Include dependency graph for gdc VR.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [gdc::UI](#)
- class [gdc::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 [gdc::VRToEncoding< T >](#)
- struct [gdc::VRToType< T >](#)

Namespaces

- [gdc](#)

Macros

- #define [TYPETOENCODING](#)(type, rep, rtype)
- #define [VRTypeTemplateCase](#)(type)

Typedefs

- `typedef String<'\', 16 > gdcm::AECComp`
- `typedef String<'\', 64 > gdcm::ASComp`
- `typedef String<'\', 16 > gdcm::CSCComp`
- `typedef String<'\', 64 > gdcm::DACComp`
- `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`

26.275.1 Macro Definition Documentation

26.275.1.1 `#define TYPETOENCODING(type, rep, rtype)`

Value:

```
template<> struct VRToEncoding<VR::type> \
{ enum { Mode = VR::rep }; }; \
template<> struct VRToType<VR::type> \
{ typedef rtype Type; };
```

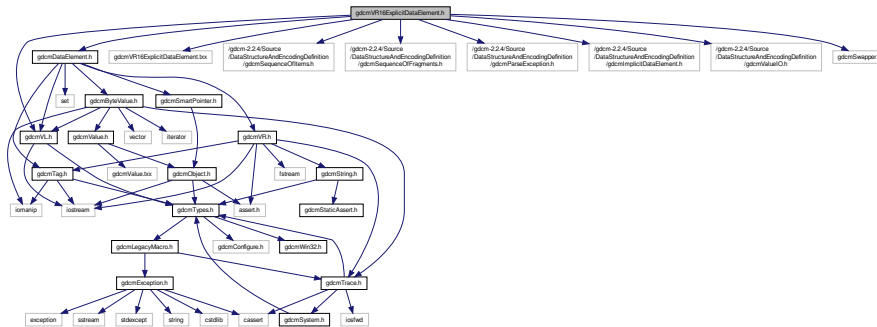
26.275.1.2 `#define VRTypeTemplateCase(type)`

Value:

```
case VR::type: \
return sizeof ( VRToType<VR::type>::Type );
```

Referenced by `gdcm::VR::GetSize()`.

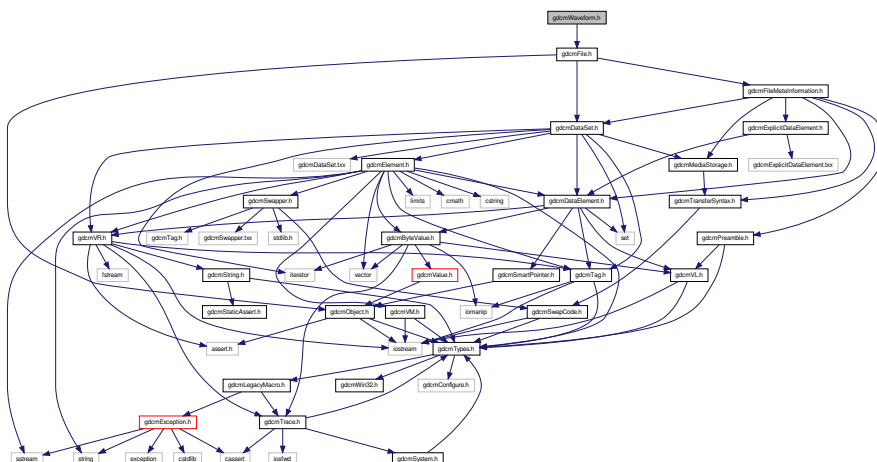

```
#include "gdcmDataElement.h"
#include "gdcmVR16ExplicitDataElement.hxx"
Include dependency graph for gdcmVR16ExplicitDataElement.h:
```



- class `gdcm::VR16ExplicitDataElement`
Class to read/write a `DataElement` as Explicit Data `Element`.

- **gdcm**

```
#include "gdcmFile.h"
Include dependency graph for gdcmWaveform.h:
```



Classes

- class [gdcm::Waveform](#)
Waveform class.

Namespaces

- [gdcm](#)

26.278 gdcmWin32.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define` [GDCM_EXPORT](#)

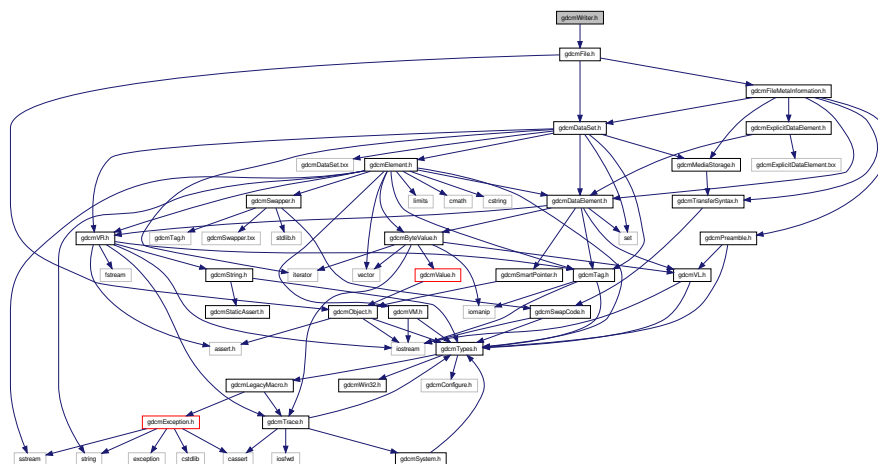
26.278.1 Macro Definition Documentation

26.278.1.1 `#define` GDCM_EXPORT

26.279 gdcmWriter.h File Reference

```
#include "gdcmFile.h"
```

Include dependency graph for gdcmWriter.h:



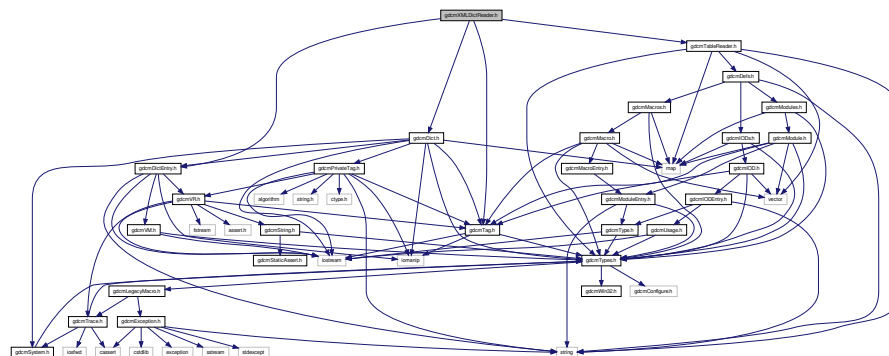
```
graph BT; gdcImage[gdcImageWriter.h] --> gdcPix[gdcPixmapWriter.h]; gdcSurf[gdcSurfaceWriter.h] --> gdcSeg[gdcSegmentWriter.h]; gdcPix --> gdcW[gdcWriter.h]; gdcSeg --> gdcW; gdcStream[gdcStreamImageWriter.h] --> gdcW; style gdcW fill:#d3d3d3
```

- class `gdcm::Writer`

Namespaces

- ## 26.280 gdcmlXMLDictReader.h File Reference

Include dependency graph for gdcmlXMLDictReader.h:

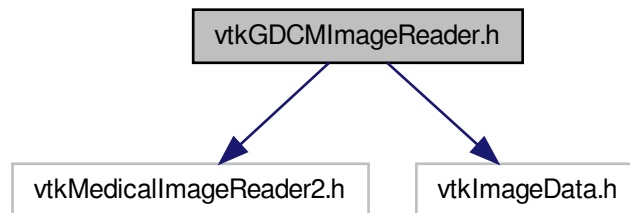


26.284 vtkGDCMImageReader.h File Reference

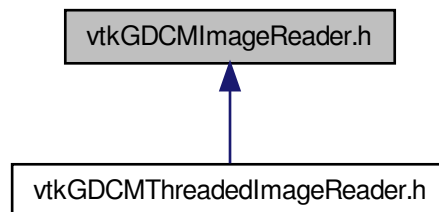
```
#include "vtkMedicalImageReader2.h"
```

```
#include "vtkImageData.h"
```

Include dependency graph for vtkGDCMImageReader.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [vtkGDCMImageReader](#)

Namespaces

- [gdc](#)

Macros

- `#define` [VTK_CMYK](#) 8
- `#define` [VTK_INVERSE_LUMINANCE](#) 5

- `#define VTK_LOOKUP_TABLE` 6
- `#define VTK_YBR` 7

26.284.1 Macro Definition Documentation

26.284.1.1 `#define VTK_CMYK` 8

26.284.1.2 `#define VTK_INVERSE_LUMINANCE` 5

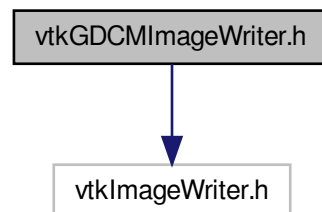
26.284.1.3 `#define VTK_LOOKUP_TABLE` 6

26.284.1.4 `#define VTK_YBR` 7

26.285 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



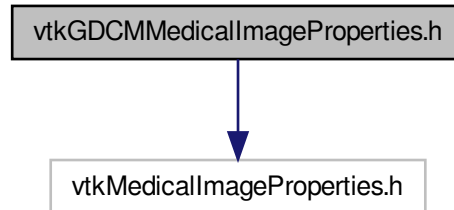
Classes

- class `vtkGDCMImageWriter`

26.286 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkMedicalImageProperties.h"
```

Include dependency graph for vtkGDCMMedicalImageProperties.h:



Classes

- class [vtkGDCMMedicalImageProperties](#)

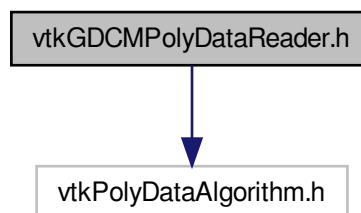
Namespaces

- [gdc](#)

26.287 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:



Classes

- class [vtkGDCMPolyDataReader](#)

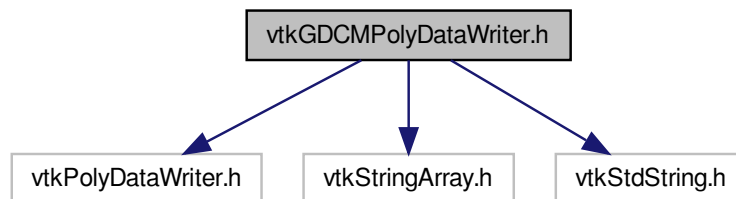
Namespaces

- [gdcm](#)

26.288 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkPolyDataWriter.h"  
#include "vtkStringArray.h"  
#include "vtkStdString.h"
```

Include dependency graph for vtkGDCMPolyDataWriter.h:



Classes

- class [vtkGDCMPolyDataWriter](#)

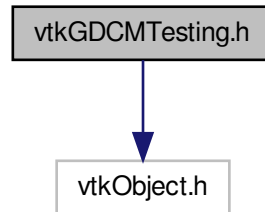
Namespaces

- [gdcm](#)

26.289 vtkGDCMTesting.h File Reference

```
#include "vtkObject.h"
```


Include dependency graph for vtkGDCMTesting.h:



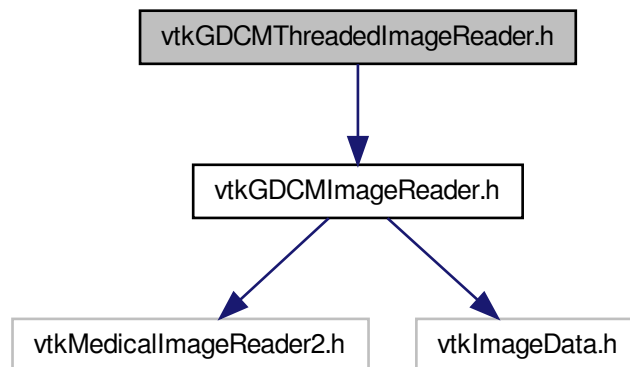
Classes

- class [vtkGDCMTesting](#)

26.290 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
```

Include dependency graph for vtkGDCMThreadedImageReader.h:



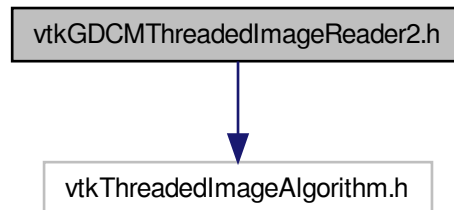
Classes

- class [vtkGDCMThreadedImageReader](#)

26.291 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkGDCMThreadedImageReader2.h:



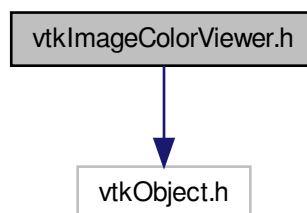
Classes

- class [vtkGDCMThreadedImageReader2](#)

26.292 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkImageColorViewer.h:



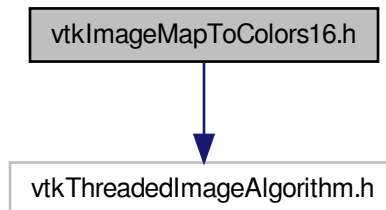
Classes

- class [vtkImageColorViewer](#)

26.293 vtkImageMapToColors16.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageMapToColors16.h:



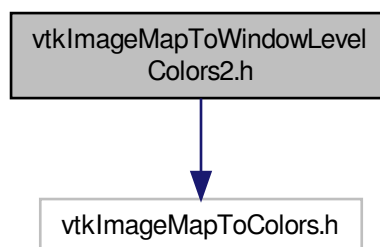
Classes

- class [vtkImageMapToColors16](#)

26.294 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkImageMapToColors.h"
```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



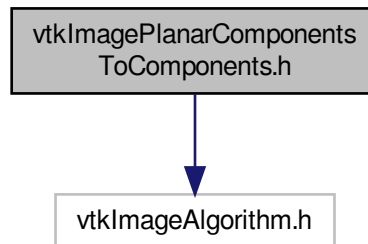
Classes

- class [vtkImageMapToWindowLevelColors2](#)

26.295 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



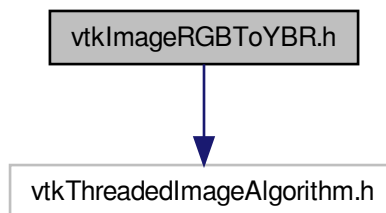
Classes

- class [vtkImagePlanarComponentsToComponents](#)

26.296 vtkImageRGBToYBR.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:



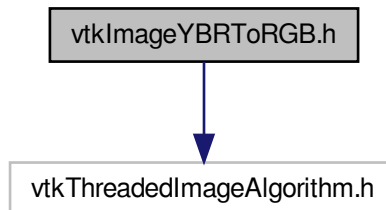
Classes

- class [vtkImageRGBToYBR](#)

26.297 vtkImageYBRToRGB.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageYBRToRGB.h:



Classes

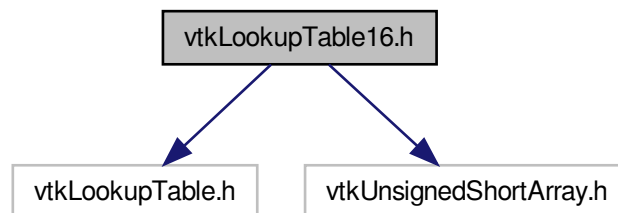
- class [vtkImageYBRToRGB](#)

26.298 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"
```

```
#include "vtkUnsignedShortArray.h"
```

Include dependency graph for vtkLookupTable16.h:



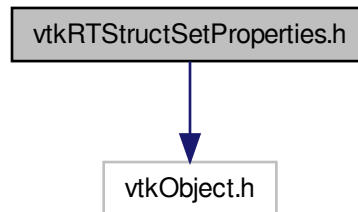
Classes

- class [vtkLookupTable16](#)

26.299 vtkRTStructSetProperties.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkRTStructSetProperties.h:



Classes

- class [vtkRTStructSetProperties](#)

Chapter 27

Example Documentation

27.1 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
// different coloration.

// Start by creating a black/white lookup table.
vtkLookupTable bwLut = new vtkLookupTable();
bwLut.SetTableRange(0, 2000);
bwLut.SetSaturationRange(0, 0);
bwLut.SetHueRange(0, 0);
bwLut.SetValueRange(0, 1);
bwLut.Build();

// Now create a lookup table that consists of the full hue circle (from
// HSV);.
vtkLookupTable hueLut = new vtkLookupTable();
hueLut.SetTableRange(0, 2000);
hueLut.SetHueRange(0, 1);
hueLut.SetSaturationRange(1, 1);
hueLut.SetValueRange(1, 1);
hueLut.Build();

// Finally, create a lookup table with a single hue but having a range
// in the saturation of the hue.
vtkLookupTable satLut = new vtkLookupTable();
satLut.SetTableRange(0, 2000);
satLut.SetHueRange(.6, .6);
satLut.SetSaturationRange(0, 1);
satLut.SetValueRange(1, 1);
satLut.Build();

// Create the first of the three planes. The filter vtkImageMapToColors
// maps the data through the corresponding lookup table created above.
// The vtkImageActor is a type of vtkProp and conveniently displays an
// image on a single quadrilateral plane. It does this using texture
// mapping and as a result is quite fast. (Note: the input image has to
// be unsigned char values, which the vtkImageMapToColors produces.);
// Note also that by specifying the DisplayExtent, the pipeline

```

```

// requests data of this extent and the vtkImageMapToColors only
// processes a slice of data.
vtkImageMapToColors 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);
    }
}

```

27.2 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.CryptographicMessageSyntax cms = new
        gdcm.CryptographicMessageSyntax();
        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;
    }
}

```

27.3 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];

        // instantiate the reader:
        gdcm.ImageReader reader = new gdcm.ImageReader();
        reader.SetFileName( filename );

        if (!reader.Read()) return 1;

        Image ir = reader.GetImage();

        uint[] dims = {0, 0, 0};
        dims[0] = ir.GetDimension(0);
        dims[1] = ir.GetDimension(1);
        dims[2] = ir.GetDimension(2);
        System.Console.WriteLine( "Dim:" + dims[0] );
        System.Console.WriteLine( "Dim:" + dims[1] );
        System.Console.WriteLine( "Dim:" + dims[2] );

        // buffer to get the pixels
        byte[] buffer = new byte[ ir.GetBufferLength()];
        System.Console.WriteLine( "Dim:" + ir.GetBufferLength() );
        ir.GetBuffer( buffer );

        for (uint z = 0; z < dims[2]; z++)
        {
            for (uint y = 0; y < dims[1] / 2; y++) // only half Y
            {
                for (uint x = 0; x < dims[0] / 2; x++) // only half X
                {
                    buffer[ (z * dims[1] + y) * dims[0] + x ] = 0; // works when pixel type == UINT8
                }
            }
        }

        DataElement pixeldata = new DataElement( new Tag(0x7fe0,0x0010) );
        pixeldata.SetByteValue( buffer, new VL( (uint)buffer.Length ) );
        ir.SetDataElement( pixeldata );
        ir.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.ExplicitVRLittleEndian ) );

        ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
        change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEGLSLossless ) );
        change.SetInput( ir );
        if ( !change.Change() )
        {
            System.Console.WriteLine( "Could not change: " + filename );
            return 1;
        }

        ImageWriter writer = new ImageWriter();
        writer.SetFileName( "out.dcm" );
        writer.SetFile( reader.GetFile() );
        writer.SetImage( change.GetOutput() );
        bool ret = writer.Write();
        if ( !ret )
        {
            return 1;
        }

        return 0;
    }
}

```

27.4 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 = gdcmm.ImageWriter()
80 writer.SetFileName( outfile )
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, outfile, tmpfile):
89     vtkreader = vtkgdcmm.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 = gdcmm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
99     tag2 = gdcmm.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 = gdcmm.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 = gdcmm.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 = vtkgdcmm.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 = gdcmm.ImageReader()
153     tmpreader.SetFileName( tmpfile )
154     if not tmpreader.Read():
155         sys.exit(1)
156
157     writer = gdcmm.ImageWriter()
158     writer.SetFileName( outfile )
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     gdcmm.Trace.DebugOff()
169     gdcmm.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 = gdcmm.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"

```

27.5 ChangeSequenceUltrasound.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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 "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmSmartPointer.h"
#include "gdcmmDataSetHelper.h"

/*
./ChangeSequenceUltrasound gdcmmData/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 < 0 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcmm::Reader reader;
    reader.SetFileName( filename );
    if (! reader.Read() )
    {
        return 1;
    }

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    gdcmm::Tag tsis(0x0008,0x2112); // SourceImageSequence
    if ( ds.FindDataElement( tsis ) )
    {
        const gdcmm::DataElement &sis = ds.GetDataElement( tsis );
        gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqsis = sis.
            GetValueAsSQ();

```



```

if ( sqsis && sqsis->GetNumberOfItems() )
{
    gdcmm::Item &item1 = sqsis->GetItem(1);
    gdcmm::DataSet &nestedds = item1.GetNestedDataSet();
    gdcmm::Tag tprcs(0x0040,0xa170); // PurposeOfReferenceCodeSequence
    if( nestedds.FindDataElement( tprcs ) )
    {
        const gdcmm::DataElement &prcs = nestedds.GetDataElement( tprcs );
        gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqprcs = prcs.
        GetValueAssSQ();
        if ( sqprcs && sqprcs->GetNumberOfItems() )
        {
            gdcmm::Item &item2 = sqprcs->GetItem(1);
            gdcmm::DataSet &nestedds2 = item2.GetNestedDataSet();
            // (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
            gdcmm::Tag tcm(0x0008,0x0104);
            if( nestedds2.FindDataElement( tcm ) )
            {
                gdcmm::DataElement cm = nestedds2.GetDataElement( tcm );
                std::string mystr = "GDCM was here";
                cm.SetByteValue( mystr.c_str(), (uint32_t)mystr.size() );
                nestedds2.Replace( cm );
            }
        }
    }
}

gdcmm::Writer writer;
writer.SetFile( file );
writer.SetFileName( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

27.6 CheckBigEndianBug.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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 gdcmmconv, 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-independant information to still
 * compare the files.
 */

#include "gdcmmImageReader.h"
#include "gdcmmImage.h"
#include "gdcmmWriter.h"
#include "gdcmmAttribute.h"
#include "gdcmmSystem.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 );
char *buffer1 = new char[s1];
is1.read(buffer1, s1);

std::ifstream is2( filename2 );
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;
}

```

27.7 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( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
}

```

```

    return 0;
}

```

27.8 ClinicalTrialIdentificationWorkflow.cs

This is a C# example on how to use [gdcm::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.FileMetaInformation fn = new gdcm.FileMetaInformation( 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: " + gdcms.UIDGenerator.GetRoot() );

gdcms.Global global = gdcms.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( !gdcms.PosixEmulation.FileIsDirectory(dir1) )
{
    System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
    return 1;
}
if( !gdcms.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 = gdcms.Filename.Join(gdcms.Testing.GetSourceDirectory(), "
/Testing/Source/Data/certificate.pem" );
gdcms.CryptographicMessageSyntax cms = new
gdcms.CryptographicMessageSyntax();
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.GetFilesNames();
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;
}
}

```

27.9 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( outfile );
    //image.GetBuffer2(out);
    //out.close();
    gdcm::ImageWriter writer;
    writer.SetImage( change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfile );
    if( !writer.Write() )
    {

```

```

    return 1;
}

return 0;
}

```

27.10 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;
}
}

```

27.11 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();
    cast->SetInput( reader->GetOutput() );
    cast->SetOutputScalarTypeToUnsignedChar();

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileName( "/tmp/cast.dcm" );
    writer->SetInput( cast->GetOutput() );
    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;
}

```

27.12 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()

```

27.13 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 );
    writer->SetInput( reader->GetOutput() );
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->Write();

    reader->Delete();
    writer->Delete();

    return 0;
}

```

27.14 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_tymap().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

```

27.15 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 #
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 save a DICOM image with PIL 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 PIL import Image, ImageOps
32
33
34 def get_gdcm_to_numpy_tymap():
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     result = numpy.frombuffer(gdcm_array, dtype=dtype)
63     maxV = float(result[result.argmax()])
64     ## linear gamma adjust
65     #result = result + .5*(maxV-result)
66     ## log gamma
67     result = numpy.log(result+50) ## 50 is appr. background level
68     maxV = float(result[result.argmax()])
69     result = result*(2.**8/maxV) ## histogram stretch
70     result.shape = d
71     return result
72
73 if __name__ == "__main__":
74     import sys
75     r = gdcm.ImageReader()
76     filename = sys.argv[1]
77     r.SetFileName( filename )
78     if not r.Read(): sys.exit(1)
79     numpy_array = gdcm_to_numpy( r.GetImage() )
80     ## L is 8 bit grey
81     ## http://www.pythonware.com/library/pil/handbook/concepts.htm
82     pilImage = Image.frombuffer('L',
83                                numpy_array.shape,
84                                numpy_array.astype(numpy.uint8),
85                                'raw','L',0,1)
86     ## cutoff removes background noise and spikes
87     pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
88     pilImage.save(sys.argv[1]+' .jpg')

```

27.16 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();
luminance->SetInput( reader->GetOutput() );

vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
writer->SetFileName( "/tmp/bla.dcm" );
writer->SetInput( luminance->GetOutput() );
//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;
}

```

27.17 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();
    copy->SetScalarType( VTK_UNSIGNED_CHAR );
    copy->SetExtent( reader->GetOutput()->GetExtent() );
    copy->AllocateScalars();

```

```

//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() );
writer->SetInput( copy );
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;
}

```

27.18 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() ==
    gdcmm::PhotometricInterpretation::MONOCHROME2 )
{
    if( gimage.GetPixelFormat() == gdcmm::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() == gdcmm::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 gdcmmData/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];

    gdcmm::ImageReader ir;
    ir.SetFileName( filename );
    if(!ir.Read())
    {
        //Read failed
        return 1;
    }

    std::cout<<"Getting image from ImageReader..."<<std::endl;

    const gdcmm::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;
}

```

27.19 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.rgb
 */

#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.rgb 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);

    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(

```

```

        gdcmm::TransferSyntax::ExplicitVRLittleEndian );

gdcmm::DataElement pixeldata( gdcmm::Tag(0x7fe0,0x0010) );
pixeldata.SetByteValue( buf, (uint32_t)len );
image.SetDataElement( pixeldata );

writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}
delete[] buf;

return 0;
}

```

27.20 CreateCMYKImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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 "gdcmmImageReader.h"
#include "gdcmmSequenceOfFragments.h"
#include "gdcmmSystem.h"
#include "gdcmmImageWriter.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 = gdcmm::System::FileSize(filename);
    std::ifstream is(filename);

    char * buf = new char[len];
    is.read(buf, len);

    gdcmm::ImageWriter writer;
    gdcmm::Image &image = writer.GetImage();
    image.SetNumberOfDimensions( 2 );
    unsigned int dims[3] = {};
    dims[0] = 380;
    dims[1] = 287;
    image.SetDimensions( dims );
    gdcmm::PixelFormat pf = gdcmm::PixelFormat::UINT8;
    pf.SetSamplesPerPixel( 4 );
    image.SetPixelFormat( pf );
    gdcmm::PhotometricInterpretation pi =
        gdcmm::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;
}

```

27.21 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 " );

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;
}

```

27.22 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=
17   "false"/>
18 """
19
20 import gdcM
21 import sys,os
22
23 if __name__ == "__main__":
24     r = gdcM.Reader()
25     # Will require Testing...
26     dataroot = gdcM.Testing.GetDataRoot()
27     filename = os.path.join( dataroot, '012345.002.050.dcm' )
28     r.SetFileName( filename )
29     r.Read()
30     f = r.GetFile()
31     ds = f.GetDataSet()
32
33     uid = "1.2.840.10008.5.1.4.1.1.66"
34     # f = gdcM.File()
35     ds = f.GetDataSet()
36     de = gdcM.DataElement( gdcM.Tag(0x0008,0x0016) )
37     de.SetByteValue( uid, gdcM.VL(len(uid)) )
38     vr = gdcM.VR( gdcM.VR.UI )
39     de.SetVR( vr )
40     ds.Replace( de )
41
42     ano = gdcM.Anonymizer()
43     ano.SetFile( r.GetFile() )
44     ano.RemovePrivateTags()
45     ano.RemoveGroupLength()
46     taglist = [
47         gdcM.Tag(0x0008,0x0008),
48         gdcM.Tag(0x0008,0x0022),
49         gdcM.Tag(0x0008,0x0032),
50         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()

```

27.23 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;
}

gdcmm::CSAHeader csa;
const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();

const gdcmm::PrivateTag &t1 = csa.GetCSAImageHeaderInfoTag();
//std::cout << t1 << std::endl;
//const gdcmm::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 gdcmm::CSAElement &cse1 = csa.GetCSAElementByName( "Columns" )
    ;
    std::cout << cse1 << std::endl;
    //const gdcmm::ByteValue *bv = cse1.GetByteValue();
    gdcmm::Element<gdcmm::VR::IS, gdcmm::VM::VM1> el;
    el.Set( cse1.GetValue() );
    dims[0] = el.GetValue();
    std::cout << "Columns:" << el.GetValue() << std::endl;
}

if( csa.FindCSAElementByName( "Rows" ) )
{
    const gdcmm::CSAElement &cse2 = csa.GetCSAElementByName( "Rows" );
    std::cout << cse2 << std::endl;
    gdcmm::Element<gdcmm::VR::IS, gdcmm::VM::VM1> el2;
    el2.Set( cse2.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 gdcmm::CSAElement &cse3 = csa.GetCSAElementByName( "PixelSpacing" );
    if( !cse3.IsEmpty() )
    {
        std::cout << cse3 << std::endl;
        gdcmm::Element<gdcmm::VR::DS, gdcmm::VM::VM2> el3;
        el3.Set( cse3.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;
}

gdcmm::ImageWriter writer;

gdcmm::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] );
gdcmm::PixelFormat pixeltype = gdcmm::PixelFormat::INT16; //

```



```

        bytewidth = spm_type('int16','bits')/8;

//unsigned long l = image.GetBufferLength();
//const int p = 1 / (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 csananimaget(0x7fe1,0x10,"SIEMENS CSA NON-IMAGE");
const gdcm::DataElement &pixeldata = ds.GetDataElement( csananimaget );
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;
}

```

27.24 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 refprogress (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;
}

gdcmm::Directory::FileNamesType filenames;
filenames.push_back( filename );

// setup the PC(s) based on the filenames:
gdcmm::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();
}

```

27.25 DecompressImage.cs

This is a C# example on how to use [gdcmm::Image](#)

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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/gdcmm/debug-gcc/bin
 * $ mono bin/DecompressImage.exe gdcmmData/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;
    }
}

```

27.26 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=gdc.jar javac ../../gdc/Examples/Java/DecompressImage.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdc.jar:. java DecompressImage gdcData/012345.002.050.dcm out.dcm
 */
import gdc.*;

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 );
        }
    }
}

```

27.27 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://gdc.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 gdcmlData/012345.002.050.dcm decompress.dcm
19 """
20
21 import gdcml
22 import sys
23
24 if __name__ == "__main__":
25
26     file1 = sys.argv[1]
27     file2 = sys.argv[2]
28
29     r = gdcml.ImageReader()
30     r.SetFileName( file1 )
31     if not r.Read():
32         sys.exit(1)
33
34     image = gdcml.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 = gdcml.Orientation.GetType(dircos)
46     l = gdcml.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 = gdcml.DataElement( gdcml.Tag(0x7fe0,0x0010) )
59     str1 = ir.GetBuffer()
60     #print ir.GetBufferLength()
61     pixeldata.SetByteValue( str1, gdcml.VL( len(str1) ) )
62     image.SetDataElement( pixeldata )
63
64     w = gdcml.ImageWriter()
65     w.SetFileName( file2 )
66     w.SetFile( r.GetFile() )
67     w.SetImage( image )
68     if not w.Write():
69         sys.exit(1)

```

27.28 DecompressImageMultiframe.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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):
 *
 * $ gdcmmraw -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
 * gdcmmraw always skip the first fragment (Basic Offset Table).
 *
 * Now from those individual jpeg stream, recreate a fake gdcmm.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.GetFilesNames();

        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 guarantee 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;
}
}

```

27.29 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;
}
}

```

27.30 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 );
        }
    }
}

```

27.31 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;
}

```

27.32 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 gdcmm {
    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::FilenameType > SortedFiles;
    std::vector< Directory::FilenameType > UnsortedFiles;

    Directory::FilenameType GetAllFileNamesFromTagToValue(
        Scanner const & s, Directory::FilenameType const & filesSubset, Tag const & t,
        const char *valueref)
    {
        Directory::FilenameType theReturn;
        if( valueref )
        {
            size_t len = strlen( valueref );
            Directory::FilenameType::const_iterator file = filesSubset.begin();
            for(; file != filesSubset.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::FilenameType 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 discriminat this
        // series. Eg. T1 / T2 intertwined. Multiple Echo (0018,0081)
        std::cerr << "Failed to sort: " << subset.begin()->c_str() << std::endl;
        for(
            Directory::FilenameType::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::FilenameType 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::FilenameType files = GetAllFileNamesFromTagToValue(
        s, subset, t3, frameuid);

    std::set< std::string > iopset;

    for(
        Directory::FilenameType::const_iterator file = files.begin();
        file != files.end(); ++file)
    {
        //std::cout << *file << std::endl;
        const char * value = s.GetValue(file->c_str(), gdcmm::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
        gdcmm::DirectionCosines ref;
        gdcmm::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(), gdcmm::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);

    gdcmm::Scanner::ValueType vt3 = s.GetValues(t3);
    for(
        gdcmm::Scanner::ValueType::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;
    gdcmm::Scanner::ValueType vt2 = s.GetValues(t2);
    for(
        gdcmm::Scanner::ValueType::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::FilenameType >::const_iterator it = SortedFiles.begin();
        it != SortedFiles.end(); ++it )
    {
        os << "Group: " << std::endl;
        for(
            Directory::FilenameType::const_iterator file = it->begin();
            file != it->end(); ++file)
        {
            os << *file << std::endl;
        }
    }
    os << "Unsorted Files: " << std::endl;
    for(
        std::vector< Directory::FilenameType >::const_iterator it = UnsortedFiles.begin();
        it != UnsortedFiles.end(); ++it )
    {
        os << "Group: " << std::endl;
        for(
            Directory::FilenameType::const_iterator file = it->begin();
            file != it->end(); ++file)
        {
            os << *file << std::endl;
        }
    }
}

std::vector< Directory::FilenameType > const & GetSortedFiles() const { return SortedFiles; }
std::vector< Directory::FilenameType > const & GetUnsortedFiles() const { return UnsortedFiles; }

void ProcessIntoVolume( Scanner const & s )
{
    gdcm::Scanner::ValueType vt1 = s.GetValues( gdcm::t1 );
    for(
        gdcm::Scanner::ValueType::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;
}

gdcmm::DiscriminateVolume dv;
dv.ProcessIntoVolume( s );
dv.Print( std::cout );

return 0;
}

```

27.33 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://gdcmm.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 gdcmm.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 gdcmmData/012345.002.050.dcm out.dcm
22
23 """
24
25 import gdcmm
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","GenerateMSOPIId"),
40     (0x0008,0x1155):("Method","GenerateMSOPIId"),
41     (0x0008,0x0018):("Method","GenerateMSOPIId"),
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
55     # Other:
56     (0x0012,0x0051):("Patient Field","Type Examen"),
57     (0x0018,0x1250):("Sequence Field","Receive Coil"),
58     (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
59     (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
60     (0x0018,0x0082):("Sequence Field","Inversion Time"),
61 }

```

```

62 class MyAnon:
63     def __init__(self):
64         self.studyuid = None
65         self.seriesuid = None
66         generator = gdcmm.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 GenerateMSOPIId(self):
78     def GenerateMSOPIId(self):
79         generator = gdcmm.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     gdcmm.FileMetaInformation.SetSourceApplicationEntityTitle
93     ( "DumbAnonymizer" )
94     gdcmm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
95
96     r = gdcmm.Reader()
97     filename = sys.argv[1]
98     r.SetFileName( filename )
99     if not r.Read(): sys.exit(1)
100
101     obj = MyAnon()
102
103     w = gdcmm.Writer()
104     ano = gdcmm.Anonymizer()
105     ano.SetFile( r.GetFile() )
106     ano.RemoveGroupLength()
107     for tag,rule in tag_rules.items():
108         if rule[0] == 'Value':
109             print tag,rule
110             ano.Replace( gdcmm.Tag( tag[0], tag[1] ), rule[1] )
111         elif rule[0] == 'Method':
112             print tag,rule
113             # result = locals()[rule[1]]()
114             methodname = rule[1]
115             if hasattr(obj, methodname):
116                 _member = getattr(obj, methodname)
117                 result = _member()
118                 ano.Replace( gdcmm.Tag( tag[0], tag[1] ), result )
119             else:
120                 print "Problem with: ", methodname
121
122     outfilename = sys.argv[2]
123     w.SetFileName( outfilename )
124     w.SetFile( ano.GetFile() )
125     if not w.Write(): sys.exit(1)

```

27.34 DumpADAC.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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 << "\n";
    }
    else
    {
        (void)len;
        os << " " << buffer << "\n";
    }
}

bool DumpADAC( std::istream & is )
{
    std::ostream &os = std::cout;

    char magic[6 + 1];
    magic[6] = 0;
    is.read( magic, 6);
    // std::cout << magic << "\n";
    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::stringstream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpADAC( is );
    if( !b ) return 1;

    return 0;
}

```

27.35 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 ) )
    {
        return false;
    }
    const DataElement & index = ds.GetDataElement( tindex );
    const DataElement & name = ds.GetDataElement( tname );
    if( index.IsEmpty() || name.IsEmpty() )
    {
        return false;
    }
    gdcmm::Element<VR::UL, VM::VM1> el1;
    el1.SetFromDataElement( index );

    gdcmm::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 ) )
    {
        return false;
    }
    const DataElement & index2 = ds.GetDataElement( tindex2 );
    if( index2.IsEmpty() )
    {
        return false;
    }
    gdcmm::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 tvalueu1(0x7fe1,0x53,"GEMS_Ultrasound_MovieGroup_001"); // UL
    PrivateTag tvalues1(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 tvalues13(0x7fe1,0x79,"GEMS_Ultrasound_MovieGroup_001"); // SL / 1-N
    PrivateTag tvalues12(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 );
        gdcmm::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 );
        gdcmm::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 );
        gdcmm::Element<VR::FD,VM::VM1> el2;
        el2.SetFromDataElement( value );
        std::cout << el2.GetValue() << std::endl;
    }
    else if( ds.FindDataElement( tvalues1 ) )
    {
        const DataElement & value = ds.GetDataElement( tvalues1 );
        gdcmm::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 );
        gdcmm::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 );
        gdcmm::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 );
        gdcmm::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 );
        gdcmm::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 );
        gdcmm::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 );
        gdcmm::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 );
        gdcmm::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 );
        gdcmm::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 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

```



```

    );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values83 =
        seq_values83.GetValueAssSQ();

    size_t ni3 = sqi_values83->GetNumberOfItems();
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcmm::Item &item_83 = sqi_values83->GetItem(i3);
        gdcmm::DataSet &ds83 = item_83.GetNestedDataSet();
        assert( ds83.Size() == 3 );

        const gdcmm::PrivateTag tseq_values84name(0x7fe1,0x84,"GEMS_Ultrasound_MovieGroup_001");
        const gdcmm::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( gdcmm::PrivateTag const &privtag0, const
    gdcmm::DataSet &subds, gdcmm::PrivateTag const &privtag1,
    gdcmm::PrivateTag const &privtag2,
    gdcmm::SequenceOfItems *sqi_dict, std::string const &indent )
{
    (void)indent;
    if( !subds.FindDataElement( privtag0 ) )
    {
        assert( 0 );
        return 1;
    }
    const gdcmm::DataElement& seq_values10 = subds.GetDataElement( privtag0 );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values10 =
        seq_values10.GetValueAssSQ();

    size_t nil = sqi_values10->GetNumberOfItems();
    // assert( nil == 1 );
    for( size_t i1 = 1; i1 <= nil; ++i1 )
    {
        gdcmm::Item &item_10 = sqi_values10->GetItem(i1);
        gdcmm::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 gdcmm::PrivateTag tseq_values20(0x7fe1,0x20,"GEMS_Ultrasound_MovieGroup_001");
        if( !ds10.FindDataElement( tseq_values20 ) )
        {
            assert( 0 );
            return 1;
        }
        const gdcmm::DataElement& seq_values20 = ds10.GetDataElement(
            tseq_values20 );
        gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values20 =
            seq_values20.GetValueAssSQ();

        size_t ni2 = sqi_values20->GetNumberOfItems();
        //assert( ni == 1 );
        for( size_t i2 = 1; i2 <= ni2; ++i2 )
        {
            gdcmm::Item &item_20 = sqi_values20->GetItem(i2);
            gdcmm::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)
            // (7fe1,1083) (*)

            const gdcmm::PrivateTag tseq_values20name(0x7fe1,0x24,"GEMS_Ultrasound_MovieGroup_001"
            );
            const gdcmm::PrivateTag tseq_values26(0x7fe1,0x26,"GEMS_Ultrasound_MovieGroup_001");
            PrintNameValueMapping3( tseq_values20name, tseq_values26, ds20, sqi_dict, " " );
            std::cout << std::endl;

```

```

        print83(ds20, sqi_dict, "    ");
    }

    print83(ds10, sqi_dict, "    ");
}
return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdc;
    const char *filename = argv[1];
    gdc::Reader reader;
    reader.SetFileName( filename );
    reader.Read();

    gdc::File &file = reader.GetFile();
    gdc::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, "    ");

#ifdef 0
    gdc::DataSet::ConstIterator it = subds.Begin();
    for( ; it != subds.End(); ++it )
    {
        const gdc::DataElement &de = *it;
        std::cout << de.GetTag() << std::endl;
    }
#endif

    return 0;
}

```

27.36 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( 1 - 16 );
    if( bytes.size() )
    {
        is.read( &bytes[0], 1 - 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);
    #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::stringstream 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 = 0.00416666668839752674; // 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;
}

```

27.37 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.GetFilesNames() );
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 gdc::Directory;
using gdc::Scanner;
const Directory::FileNamesType& files = d.GetFilesNames();
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;
}

```

27.38 DuplicatePCDE.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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
CSAHeaderType
(0029,1019) LO [20050723] # 8, 1
CSAHeaderVersion
(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 CSAHeaderInfo
(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 ...)

```



```

gdcmm::DataSet dup;
gdcmm::Tag new_private(0x0009,0x0);
while (start.GetGroup() == 0x9 )
{
    const gdcmm::DataElement& de = ds.FindNextDataElement(start);
    const gdcmm::Tag &t = de.GetTag();
    if( t.IsPrivateCreator() )
    {
        std::cout << t << std::endl;
        // Ok let's duplicate into the next available attribute:
        gdcmm::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;
        gdcmm::DataElement duplicate = de;
        duplicate.GetTag().SetPrivateCreator( new_private );
        if( const gdcmm::ByteValue *bv = duplicate.GetByteValue() )
        {
            // Warning: when doing : duplicate = de, only the pointer to the ByteValue is passed
            // (to avoid large memory duplicate). We need to explicitly duplicate the bytevalue ourselves:
            gdcmm::ByteValue *dupbv = new gdcmm::ByteValue( bv->GetPointer(),
                bv->GetLength() );
            // Let's recognize the duplicated ASCII-type elements:
            if( duplicate.GetVR() & gdcmm::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) );
}

gdcmm::DataSet::ConstIterator it = dup.Begin();
for( ; it != dup.End(); ++it )
{
    ds.Insert( *it );
}

gdcmm::Writer w;
w.SetFile( file );
w.SetFileName( outfilename );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

27.39 ELSCINT1WaveToText.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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 "gdcmmReader.h"
#include "gdcmmPrivateTag.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 (gdcmm-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' << buffer[i] << '\t' << buffer
                << '\t' << " " << '\t' << " " << '\t' << buffer[i]
            [i+1] << std::endl;
            if (buffer[i+75] == 16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " " << '\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' << " "
                << '\t' << buffer[i+74] << '\t' << " " << '\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' << " " << '\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' << " " << '\t' <<
                buffer[i+74] << '\t' << " " << '\t' << " " << '\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' << " "
                << '\t' << buffer[i+74] << '\t' << buffer[i+74] << '\t' << " " << '\t' << " " << '\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' << " " << '\t' << " "
            << std::endl;
            if (buffer[i+75] == 16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " " << '\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' << " "
                << '\t' << buffer[i+74] << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " "
            << std::endl;
            if (buffer[i+75] == -32768)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " " << '\t' << " "
                << '\t' << " " << '\t' << buffer[i+74] << '\t' << " " << '\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( outfile );
    // Dump that to a CSV file:
    wave2stream( os, bv->GetPointer(), bv->GetLength() );
    os.close();

    return 0;
}

```

27.40 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( outfile );

    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: " << outfile << std::endl;
        return 1;
    }

    return 0;
}

```

27.41 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 MIMETimeTypeOfEncapsulatedDocument
* ...
*
* 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;
    }
}

```

27.42 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 &nesteddds = item.GetNestedDataSet();

    if( ! nesteddds.FindDataElement( gdcm::Tag( 0x0400,0x0520 ) ) ) return 1;

    const gdcm::DataElement &EncryptedContent = nesteddds.
        GetDataElement( gdcm::Tag( 0x0400,0x0520 ) );

    const gdcm::ByteValue *bv = EncryptedContent.GetByteValue();

    std::ofstream of( outfile );
    of.write( bv->GetPointer(), bv->GetLength() );
    of.close();

    return 0;
}

```

27.43 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 ); (void)b;
return true;
}

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" );
            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;
}

```

27.44 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];

        // 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();

        // 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 ];

        // 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;
}
}

```

27.45 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/rle16l00.dcm
 * $ md5sum /tmp/frame_rgb.raw
 * 73bf61325fdb6e2830244a2b7b0c4ae2 /tmp/frame_rgb.raw
 * $ gdcming --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
 * $ gdcmvviewer 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;
}
}

```

27.46 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 );
    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 del( gdcm::Tag(0x8,0x16) );
    del.SetVR( gdcm::VR::UI );
    gdcm::MediaStorage ms( gdcm::MediaStorage::CTImageStorage
        );
    del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
    ds.Insert( del );

    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 );

```

```

gdcmm::Attribute<0x0028,0x0010> row = {image->comps[0].w};
//row.SetValue(512);
ds.Insert( row.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcmm::Attribute<0x0028,0x0011> col = {image->comps[0].h};
ds.Insert( col.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0002> at1 = {image->numcomps};
ds.Insert( at1.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcmm::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;
        gdcmm::Element<gdcmm::VR::IS,gdcmm::VM::VM1> el2;
        el2.SetValue(i+1);
        gdcmm::DataElement rfn = el2.GetAsDataElement(); //ulr --> upper
        left row
        rfn.SetTag( gdcmm::Tag(0x0008,0x1160) );

        gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> el;
        el.SetValue(1,0);
        el.SetValue(1,1);
        gdcmm::DataElement ulr = el.GetAsDataElement(); //ulr --> upper
        left col/row
        ulr.SetTag( gdcmm::Tag(0x0048,0x0201) );

        gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> el1;
        el1.SetValue(col,0);
        el1.SetValue(row,1);
        gdcmm::DataElement brr = el1.GetAsDataElement();
        brr.SetTag( gdcmm::Tag(0x0048,0x0202) ); //brr --> bottom right col/row
        gdcmm::Item it;
        gdcmm::DataSet &nds = it.GetNestedDataSet();
        nds.Insert( rfn );
        nds.Insert( ulr );
        nds.Insert( brr );

        sq->AddItem(it);
    }

    gdcmm::Writer w1;
    gdcmm::File &file1 = w1.GetFile();
    gdcmm::DataSet &ds1 = file1.GetDataSet();
    file1.GetHeader().SetDataSetTransferSyntax(
        gdcmm::TransferSyntax::ExplicitVRLittleEndian );

    gdcmm::UIDGenerator uid1;
    gdcmm::DataElement dea( gdcmm::Tag(0x8,0x18) ); // SOP Instance UID
    dea.SetVR( gdcmm::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<<"\nable to read";

// 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( outfile, std::ios::out | std::ios::binary );
theStreamWriter.SetStream(of);

if( !Different_Resolution( theStreamWriter, filename,res,of ) ) return 1;

uint16_t firstTag1 = 0xffff;
uint16_t secondTag1 = 0xe0dd;
uint32_t thirdTag1 = 0x00000000;
//uint16_t fourthTag1 = 0xffff;
const int theBufferSize = 2*sizeof(uint16_t)+sizeof(uint32_t);
char* tmpBuffer2 = new char[theBufferSize];
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, theBufferSize);
of.flush();
assert( of );

return 0;
}

```

27.47 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;
}
}

```

27.48 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 del( gdcm::Tag(0x8,0x16) );
del.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms(
    gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( del );

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<0x0028,0x0006> 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;

gdcmm::Element<gdcmm::VR::IS,gdcmm::VM::VM1> el2;
el2.SetValue(1);
gdcmm::DataElement rfn = el2.GetAsDataElement(); //rfn --->
    reference frame number
rfn.SetTag( gdcmm::Tag(0x0008,0x1160) );

gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> el;
el.SetValue(1,0);
el.SetValue(1,1);
gdcmm::DataElement ulr = el.GetAsDataElement(); //ulr --> upper
    left col/row
ulr.SetTag( gdcmm::Tag(0x0048,0x0201) );

gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> ell;
ell.SetValue(col1,0);
ell.SetValue(row1,1);
gdcmm::DataElement brr = ell.GetAsDataElement();
brr.SetTag( gdcmm::Tag(0x0048,0x0202) ); //brr --> bottom right col/row

gdcmm::Item it;
gdcmm::DataSet &nds = it.GetNestedDataSet();
nds.Insert( rfn );
nds.Insert( ulr );
nds.Insert( brr );

sq->AddItem(it);

gdcmm::DataElement des( gdcmm::Tag(0x0048,0x0200) );
des.SetVR(gdcmm::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<<"\nable to read";

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 =
    gdcmm::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;
}

```

27.49 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;
}
}

```

27.50 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" );
}

```

27.51 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]

```

27.52 FixBrokenJ2K.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

This software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the above copyright notice for more information.

```

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmFile.h"

// http://www.lost.in.ua/dicom/c.dcm
//
// -> BuggyJ2Kvvvua-fixed2-j2k.dcm

/*
 * This program attempts to fix a broken J2K/DICOM:
 * It contains 2 bugs:
 * 1. The first 8 bytes seems to be random bytes: remove them
 * 2. YCC is set to 1, while image is grayscale need to set it back to 0
 *
 * Ref:
 * It's a software from http://rentgenprom.ru/ , shipped with universal digital radiographic units
 * "ProScan-2000". The Ukrainian manufacturer developed own digital radiographic unit and it is
 * compatible with software from "ProScan-2000".
 */
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;

```

```

        gdcmm::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;
}

gdcmm::Writer writer;
writer.SetFile( reader.GetFile() );
writer.SetFileName( outfilename );
writer.CheckFileMetaInformationOff();
if( !writer.Write() )
{
    std::cerr << "Could not write" << std::endl;
}

// paranoid check:
gdcmm::ImageReader ireader;
ireader.SetFileName( outfilename );
if( !ireader.Read() )
{
    std::cerr << "file written is still not valid, please report" << std::endl;
    return 1;
}

return 0;
}

```

27.53 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://gdcmm.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 gdcmm 2.0.9
19 """
20
21 import gdcmm
22 import sys
23
24 filename = sys.argv[1]
25 outname = sys.argv[2]
26
27 # read
28 r = gdcmm.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.FindElement( 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 "Success!"
77 sys.exit(0) # success

```

27.54 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
        ( "FixJAIBugJPEGs" );

    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" );
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> rbyteOut;

```

```

//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;
}

```

27.55 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"

#ifdef 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;

```

```

gdcmm::Directory d;
d.Load(filename, recursive);
gdcmm::Directory::FileNamesType const &files = d.
GetFileNames();
for( gdcmm::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( gdcmm::System::FileExists( filename ) )
        {
            if( gdcmm::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 )
{
    //gdcmm::Trace::DebugOn();
    //gdcmm::Trace::WarningOn();
    gdcmm::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();
v16->SetInput( reader->GetOutput() );
v16->SetOutputSpacing( spacing[0], spacing[1], ippszspacing );
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();
    planeWidgetX->SetInput(v16->GetOutput());
    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();
    planeWidgetY->SetInput(v16->GetOutput());
    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();
planeWidgetZ->SetInput(vl6->GetOutput());
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();
colorMap->SetInput(planeWidgetZ->GetResliceOutput());
colorMap->SetLookupTable(planeWidgetX->GetLookupTable());

vtkImageActor* imageActor = vtkImageActor::New();
imageActor->PickableOff();
imageActor->SetInput(colorMap->GetOutput());

// 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( IOEventLog );

// 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;
}

```

27.56 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();
    flip->SetInput( reader->GetOutput() );
    flip->SetFilteredAxis(0);
    flip->Update();

    vtkImageReslice *reslice = vtkImageReslice::New();
    //reslice->SetInput( reader->GetOutput() );
    reslice->SetInput( flip->GetOutput() );
    //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();
texture->SetInput(ima);
texture->InterpolateOn();
texture->SetLookupTable(table);

// PlaneSource
vtkPlaneSource* plane = vtkPlaneSource::New();

// PolyDataMapper
vtkPolyDataMapper *planeMapper = vtkPolyDataMapper::New();
planeMapper->SetInput(plane->GetOutput());

// 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;
}

```

27.57 gdcmrtonplan.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/1,1 Ion Beam Sequence
(ffff,e000) na (Item with undefined length)         # 4,1 Institutional Department Name
(0008,1040) LO [Test]

```

```

(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/1,1 Ion Range Compensator Sequence
    (ffff,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
    (ffff,e00d)
*/
const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();
gdcmm::Tag tbeamsq(0x300a,0x03a2);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcmm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
//std::cout << beamsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi = beamsq.
    GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//     const gdcmm::Item &item = sqi->GetItem(1); // Item start at #1
//     const gdcmm::Item &item = sqi->GetItem(1); // Item start at #1
//     const gdcmm::DataSet& nestedds = item.GetNestedDataSet();
//     std::cout << nestedds << std::endl;
//     gdcmm::Tag tcompensatorsq(0x300a,0x02ea);
//     if( !nestedds.FindDataElement( tcompensatorsq ) )
//     {
//         return 1;
//     }
//     const gdcmm::DataElement &compensatorsq = nestedds.
//         GetDataElement( tcompensatorsq );
//     std::cout << compensatorsq << std::endl;
//     gdcmm::SmartPointer<gdcmm::SequenceOfItems> ssqi = compensatorsq
//         .GetValueAsSQ();
//     const gdcmm::Item &item2 = ssqi->GetItem(1); // Item start at #1
//     const gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
//     std::cout << nestedds2 << std::endl;
//     gdcmm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
//     if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
//     {
//         return 1;
//     }
//     const gdcmm::DataElement &compensatorthicknessdata = nestedds2.
//         GetDataElement( tcompensatorthicknessdata );
//     std::cout << compensatorthicknessdata << std::endl;
//     gdcmm::Attribute<0x300a,0x00ec> at;
//     at.SetFromDataElement( compensatorthicknessdata );
//     const double* pts = at.GetValues();
//     // (300a,00e7) IS [35] # 2,1 Compensator Rows
//     gdcmm::Attribute<0x300a,0x00e7> at1;
//     const gdcmm::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);
    img->SetScalarTypeToDouble();
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
    img->SetNumberOfScalarComponents(1);
    img->GetPointData()->SetScalars(d);

    img->Update();
    img->Print(std::cout);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
    writeb->SetInput( img );
    writeb->SetFileName( outfilename );
    writeb->Write( );
/*
    (300a,03a6) SQ # u/1,1 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;
gdcmm::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( blockdata );

vtkDoubleArray *scalars = vtkDoubleArray::New();
scalars->SetNumberOfComponents(3);

gdcmm::Attribute<0x300a,0x0104> bnpts; // IS [179 ]
           # 4,1 Block Number of Points
if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
{
    return 1;
}
const gdcmm::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();
output->Update();
output->Print( std::cout );

// }

vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

vtkImageColorViewer *viewer = vtkImageColorViewer::New();
viewer->SetInput(img);
viewer->SetupInteractor(iren);
viewer->SetSize(600, 600);
viewer->GetRenderer()->ResetCameraClippingRange();
viewer->Render();
viewer->GetRenderer()->ResetCameraClippingRange();

vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
//vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
cubeMapper->SetInput( output );
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();
writec->SetInput( output );
writec->SetFileName( outfilename2 );
writec->Write( );

iren->Initialize();

```

```

    iren->Start();

    return 0;
}

```

27.58 gdcmrtpplan.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/1,1 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/1,1 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\35.43\35.43\34.54\34.54\34.71\36.10\38.62\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);
    img->SetScalarTypeToDouble();
    img->SetSpacing( at3.GetValue(1), at3.GetValue(0), 1); // FIXME image is upside down
    img->SetOrigin( at4.GetValue(0), at4.GetValue(1), 1);
    img->SetNumberOfScalarComponents(1);
    img->GetPointData()->SetScalars(d);

    vtkXMLImageDataWriter *writeb= vtkXMLImageDataWriter::New();
    writeb->SetInput( img );
    writeb->SetFileName( outfilename );
    writeb->Write();
/*
(300a,00f4) SQ # u/1,1 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 &tblockdata = 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();
output->Update();
output->Print( std::cout );

// }

vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();

vtkImageColorViewer *viewer = vtkImageColorViewer::New();
viewer->SetInput(img);
viewer->SetupInteractor(iren);
viewer->SetSize(600, 600);
viewer->Render();

vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
//vtkPolyDataMapper2D* cubeMapper = vtkPolyDataMapper2D::New();
cubeMapper->SetInput( output );
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;
}

```

27.59 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)
    {
        append->AddInput( reader->GetOutput(i) );
    }

    vtkPolyDataWriter * writer = vtkPolyDataWriter::New();
    writer->SetInput( reader->GetOutput() );
    writer->SetFileName( "rtstruct.vtk" );
    // writer->Write();

    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    // vtkPolyDataMapper2D * cubeMapper = vtkPolyDataMapper2D::New();
    // cubeMapper->SetInput( reader->GetOutput() );
    cubeMapper->SetInput( append->GetOutput() );
    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;
}

```

27.60 gdcmttexture.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();
texture->SetInput(ima);
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();
planeMapper->SetInput(plane->GetOutput());

// 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;
}

```

27.61 gdcmvolume.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 gdcmlData/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;
}

```

27.62 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 gdcmm::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 gdcmm::Global &g = gdcmm::Global::GetInstance();
    static const gdcmm::Dicts &dicts = g.GetDicts();
    static const gdcmm::Dict &pubdict = dicts.GetPublicDict();
    using gdcmm::VR;
    using gdcmm::Tag;

    gdcmm::Writer w;

    gdcmm::File &f = w.GetFile();
    gdcmm::DataSet &ds = f.GetDataSet();

    gdcmm::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( true );
    fef.SetFile( w.GetFile() );
    if( !fef.Change() )
    {
        std::cerr << "Failed to change" << std::endl;
        return 1;
    }

    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new
        gdcmm::SequenceOfItems();
    sq->SetLengthToUndefined();

    // gdcmm::DummyValueGenerator dv;

    const std::size_t len = 10;
    char ss[len+1];
    ss[len] = '\0';

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdcmm::DataElement owner( gdcmm::Tag(0x4d4d, 0x10) );
    owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdcmm::VR::LO );

    // Create an item
    gdcmm::Item it;
    it.SetVLToUndefined();
    gdcmm::DataSet &nds = it.GetNestedDataSet();
    // nds.Insert(owner);
    // nds.Insert(de);

    // Insert sequence into data set
    gdcmm::DataElement des( gdcmm::Tag(0x4d4d, 0x1001) );
    des.SetVR(gdcmm::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;
}

```

27.63 GenerateDICOMDIR.cs

This is a C# example on how to use `gdcm::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.GetFilesNames();

gdcm.DICOMDIRGenerator gen = new DICOMDIRGenerator();
gen.SetFilenames( filenames );
gen.SetDescriptor( descriptor );
if( !gen.Generate() )
{
    return 1;
}

gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "GeneratedDICOMDIR" );
gdcm.Writer writer = new Writer();
writer.SetFile( gen.GetFile() );
writer.SetFileName( outfilename );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

27.64 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();
    cubeMapper->SetInput( inData );
    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 ... RTSTRUCT
 */
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";
gdcmm::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();
writer->SetInput(0, blank);
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)
{
    writer->SetInput(i, reader->GetOutput(i-1));
    append->AddInput(reader->GetOutput(i-1));
    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()->GetStructureSetROIInterpretedType(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;
}

```

27.65 GenerateStandardSOPClasses.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 "gdcmlDefs.h"
#include "gdcmlUIDs.h"
#include "gdcmlGlobal.h"
#include "gdcmlMediaStorage.h"
#include "gdcmlSOPClassUIDToIOD.h"

int main(int , char *[])
{
    using gdcml::MediaStorage;
    gdcml::Global& g = gdcml::Global::GetInstance();
    if( !g.LoadResourcesFiles() )
    {
        std::cerr << "Could not LoadResourcesFiles" << std::endl;
        return 1;
    }

    const gdcml::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;

    gdcml::MediaStorage::MSType mst;
    for ( mst = gdcml::MediaStorage::MediaStorageDirectoryStorage
        ; mst < gdcml::MediaStorage::MS_END;
        mst = (gdcml::MediaStorage::MSType)(mst + 1) )
    {
        const char *iod = defs.GetIODNameFromMediaStorage(mst);
        gdcml::UIDs uid;
        uid.SetFromUID( gdcml::MediaStorage::GetMSString(mst) /*
            mst.GetString()*/ );
        if( iod )
        {
            const char *iod_ref = gdcml::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;
}

```

```
}

```

27.66 GenFakelIdentifyFile.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 "gdcmlReader.h"
#include "gdcmlGlobal.h"
#include "gdcmlDummyValueGenerator.h"
#include "gdcmlMediaStorage.h"
#include "gdcmlWriter.h"
#include "gdcmlItem.h"
#include "gdcmlImageReader.h"
#include "gdcmlSequenceOfItems.h"
#include "gdcmlAttribute.h"
#include "gdcmlFile.h"
#include "gdcmlTag.h"
#include "gdcmlDict.h"
#include "gdcmlDictEntry.h"
#include "gdcmlDicts.h"
#include "gdcmlTransferSyntax.h"
#include "gdcmlUIDGenerator.h"
#include "gdcmlAnonymizer.h"

#include <cstdlib>
#include <cstring>

gdcml::DataElement CreateFakeElement(gdcml::Tag const &tag, bool toremove)
{
    static const gdcml::Global &g = gdcml::Global::GetInstance();
    static const gdcml::Dicts &dicts = g.GetDicts();
    static const gdcml::Dict &pubdict = dicts.GetPublicDict();
    static size_t countglobal = 0;
    static std::vector<gdcml::Tag> balcptags =
        gdcml::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
        ();
    size_t count = countglobal % balcptags.size();

    const gdcml::DictEntry &dictentry = pubdict.GetDictEntry(tag);

    gdcml::DataElement de;
    de.SetTag( tag );
    using gdcml::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
    gdc::Item it;
    it.SetVLToUndefined();
    gdc::DataSet &nds = it.GetNestedDataSet();
    // Insert sequence into data set
    assert(de.GetVR() == gdc::VR::SQ );
    gdc::SmartPointer<gdc::SequenceOfItems> sq = new
        gdc::SequenceOfItems();
    sq->SetLengthToUndefined();
    de.SetValue(*sq);
    de.SetVLToUndefined();
    //ds.Insert(de);

    if( !toremove )
    {
        nds.Insert( CreateFakeElement( balcptags[count], true ) );
        countglobal++;
    }
    else
    {
        gdc::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no
            reason to be 'anonymized'...
        nds.Insert( at1.GetAsDataElement() );
        gdc::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 gdc::Tag;
    using gdc::VR;
    const char *outfilename = argv[1];

    std::vector<gdc::Tag> balcptags =
        gdc::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
            ();

    gdc::Writer w;
    gdc::File &f = w.GetFile();
    gdc::DataSet &ds = f.GetDataSet();

    // Add attribute that need to be anonymized:
    std::vector<gdc::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 gdc::Global &g = gdc::Global::GetInstance();
    static const gdc::Dicts &dicts = g.GetDicts();
    static const gdc::Dict &pubdict = dicts.GetPublicDict();

    using gdc::Dict;
    Dict::ConstIterator dictit = pubdict.Begin();
    for(; dictit != pubdict.End(); ++dictit)
    {
        const gdc::Tag &dicttag = dictit->first;
        if( dicttag == Tag(0x6e65,0x6146) ) break;
        //const gdc::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;
}

```

27.67 GenFakelImage.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 exercise we will pretend that this image is referencing
// two source image (we need to generate fake UID for that).
const char ReferencedSOPClassUID[] = "1.2.840.10008.5.1.4.1.1.7"; // Secondary Capture
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );
fd.AddReference( ReferencedSOPClassUID, uid.Generate() );

// Again for the purpose of the exercise we will pretend that the image is a
// multiplanar reformat (MPR):
// CID 7202 Source Image Purposes of Reference
// {"DCM",121322,"Source image for image processing operation"},
fd.SetPurposeOfReferenceCodeSequenceCodeValue( 121322 );
// CID 7203 Image Derivation
// { "DCM",113072,"Multiplanar reformatting" },
fd.SetDerivationCodeSequenceCodeValue( 113072 );
fd.SetFile( *file );
// If all Code Value are ok the filter will execute properly
if( !fd.Derive() )
{
    std::cerr << "Sorry could not derive using input info" << std::endl;
    return 1;
}

// We pass both :
// 1. the fake generated image
// 2. the 'DERIVED' dataset object
// to the writer.
gdcm::ImageWriter w;
w.SetImage( *im );
w.SetFile( fd.GetFile() );

```

```

// Set the filename:
w.SetFileName( "ybr2.dcm" );
if( !w.Write() )
{
    return 1;
}

return 0;
}

```

27.68 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 where 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( outfile );
if (!w.Write() )
{
    return 1;
}

return 0;
}

```

27.69 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 where 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( outfile );
    if( !w.Write() )
    {
        return 1;
    }

    return 0;
}

```

27.70 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;
    }
}

```

```
}

```

27.71 GetJPEGSamplePrecision.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 != gdcmm::TransferSyntax::JPEGLosslessProcess14 && ts !=
        gdcmm::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 gdcmm::DataSet &ds = file.GetDataSet();

    const gdcmm::Tag rawTag(0x7fe0, 0x0010); // Default to Pixel Data
    const gdcmm::DataElement& pdde = ds.GetDataElement( rawTag );
    const gdcmm::SequenceOfFragments *sf = pdde.
        GetSequenceOfFragments();
    if( sf )
    {
        std::ofstream output(outfilename, std::ios::binary);
        sf->WriteBuffer(output);
    }
    else
    {
        std::cerr << "Error" << std::endl;
        return 1;
    }

    gdcmm::JPEGCodec jpeg;
    std::ifstream is(outfilename);
    gdcmm::PixelFormat pf ( gdcmm::PixelFormat::UINT8 ); // let's
        pretend it's a 8bits jpeg
    jpeg.SetPixelFormat( pf );
    gdcmm::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;
}

```

27.72 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://gdcmm.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       !
56       print diffgraddir
57
58       # repeat for t2 if you like it:
59       if ds.FindDataElement( t2 ):
60           csa_t2.LoadFromDataElement( ds.GetDataElement( t2 ) )
61           # print csa_t2
62
63       gdt = csa_t2.GetCSAElementByName( "GradientDelayTime" )
64       print gdt
65
66       bv = gdt.GetByteValue();
67       #print bv
68       str = bv.GetPointer()
69       print str.split("\\")

```

27.73 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 tsqr(0x0018,0x6011);
    if( !ds.FindDataElement( tsqr ) )
    {
        return false;
    }

    const gdcm::DataElement &sqr= ds.GetDataElement( tsqr );
    //std::cout << sqr << std::endl;
    const gdcm::SequenceOfItems *sqi = sqr.GetValueAssSQ();
    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;
}

```

27.74 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
 * (7fel,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(0x7fel,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(0x7fel,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(0x7fel,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(0x7fel,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(0x7fel,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 << "E1= " << el.GetValue() << std::endl;
        dimx = el.GetValue(0);
        dimy = el.GetValue(1);
    }
}

const PrivateTag tseq3(0x7fel,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(0x7fel,0x37,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq4(0x7fel,0x43,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq5(0x7fel,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" );
        out.write( bv4->GetPointer(), bv4->GetLength() );
        out.close();
    }
    #endif
    const ByteValue *bv5 = seq5.GetByteValue();
    #if 0
    {
        std::ofstream out( "/tmp/mo5" );
        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;
}

```

27.75 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()

```

27.76 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(vtkgdc.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.Handle, false, false);
        return imgout;
    }

    static vtkgdc.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        vtkgdc.vtkImageData imgout = new vtkgdc.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

        vtkgdc.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;
}

```

27.77 HelloActiviz2.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 gdcmlData/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() == vtkgdcml.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;
    }
}

```

27.78 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.WriteLine(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;
    }
}

```

27.79 HelloActiviz4.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed 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.WriteLine(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;
    }
}

```

27.80 HelloActiviz5.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed 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;
    }
}

```

27.81 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 );
    }
}

```

27.82 HelloVizWorld.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Basic example for dealing with a DICOM file that contains an Image
 * (read: Pixel Data element)
 */

#include "gdcmImageReader.h"
#include "gdcmImageWriter.h"
#include "gdcmImage.h"
#include "gdcmPhotometricInterpretation.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {

```

```

    std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
    return 1;
}
const char *filename = argv[1];
const char *outfilename = argv[2];

// Instantiate the image reader:
gdcm::ImageReader reader;
reader.SetFileName( filename );
if( !reader.Read() )
{
    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
}
// If we reach here, we know for sure 2 things:
// 1. It is a valid DICOM
// 2. And it contains an Image !

// The output of superclass gdcm::Reader is a gdcm::File
//gdcm::File &file = reader.GetFile();

// The other output of gdcm::ImageReader is a gdcm::Image
const gdcm::Image &image = reader.GetImage();

// Let's get some property from the image:
unsigned int ndim = image.GetNumberOfDimensions();
// Dimensions of the image:
const unsigned int *dims = image.GetDimensions();
// Origin
const double *origin = image.GetOrigin();
const gdcm::PhotometricInterpretation &pi = image.
    GetPhotometricInterpretation();
for(unsigned int i = 0; i < ndim; ++i)
{
    std::cout << "Dim(" << i << "): " << dims[i] << std::endl;
}
for(unsigned int i = 0; i < ndim; ++i)
{
    std::cout << "Origin(" << i << "): " << origin[i] << std::endl;
}
std::cout << "PhotometricInterpretation: " << pi << std::endl;

// Write the modified DataSet back to disk
gdcm::ImageWriter writer;
writer.SetImage( image );
writer.SetFileName( outfile );
//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: " << outfile << std::endl;
    return 1;
}

return 0;
}

```

27.83 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() == vtkgdcmt.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;
    }
}

```

27.84 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 preferred
// an import statement along the line of:
// import vtkgdcmt.*;
import vtk.*;

/*
 * Compilation:
 * CLASSPATH=vtkgdcmt.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:vtkgdcmt.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("vtkgdcmtJava");
        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() == vtkgdc.vtkgdc.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 );
}
}

```

27.85 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 vtkgdc;

```

```

/*
 * 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;
    }
}

```

27.86 HelloWorld.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example is ... guess what this is for :)
 */

#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmAttribute.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    // Instantiate the reader:
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }
}

```



```

// If we reach here, we know for sure only 1 thing:
// It is a valid DICOM file (potentially an old ACR-NEMA 1.0/2.0 file)
// (Maybe, it's NOT a Dicom image -could be a DICOMDIR, a RTSTRUCT, etc-)

// The output of gdcm::Reader is a gdcm::File
gdcm::File &file = reader.GetFile();

// the dataset is the the set of element we are interested in:
gdcm::DataSet &ds = file.GetDataSet();

// Construct a static(*) type for Image Comments :
gdcm::Attribute<0x0020,0x4000> imagecomments;
imagecomments.SetValue( "Hello, World !" );

// Now replace the Image Comments from the dataset with our:
ds.Replace( imagecomments.GetAsDataElement() );

// Write the modified DataSet back to disk
gdcm::Writer writer;
writer.CheckFileMetaInformationOff(); // Do not attempt to reconstruct the
    file meta to preserve the file           // as close to the original as possible.
writer.SetFileName( outfilename );
writer.SetFile( file );
if( !writer.Write() )
{
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

/*
 * (*) static type, means that extra DICOM information VR & VM are computed at compilation time.
 * The compiler is deducing those values from the template arguments of the class.
 */

```

27.87 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))

```

27.88 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;
}

```

27.89 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/1, 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

```

```

gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
//item2.SetVLTToUndefined();
//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
gdcmm::Tag tcontourdata(0x3006,0x0050);
const gdcmm::DataElement & contourdata = nestedds2.
    GetDataElement( tcontourdata );
//std::cout << contourdata << std::endl;

//const gdcmm::ByteValue *bv = contourdata.GetByteValue();
gdcmm::Attribute<0x3006,0x0046> ncontourpoints;
ncontourpoints.Set( nestedds2 );

gdcmm::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 );

gdcmm::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

gdcmm::TransferSyntax ts =
    gdcmm::TransferSyntax::ImplicitVRLittleEndian;
ts = gdcmm::TransferSyntax::ExplicitVRLittleEndian;

gdcmm::FileMetaInformation &fmi = file.GetHeader();
const char *tsuid = gdcmm::TransferSyntax::GetTSString( ts );
// const char * is ok since padding is \0 anyway...
gdcmm::DataElement de( gdcmm::Tag(0x0002,0x0010) );
de.SetByteValue( tsuid, (uint32_t)strlen(tsuid) );
de.SetVR( gdcmm::Attribute<0x0002, 0x0010>::GetVR() );
fmi.Replace( de );
fmi.Remove( gdcmm::Tag(0x0002,0x0012) ); // will be regenerated
fmi.Remove( gdcmm::Tag(0x0002,0x0013) ); // ' ' ' '
fmi.SetDataSetTransferSyntax(ts);

gdcmm::Writer w;
w.SetFile( file );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

27.90 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();
    cast->SetInput( reader->GetOutput() );
    cast->SetOutputScalarTypeToUnsignedShort();

    vtkImageMagnify *magnify = vtkImageMagnify::New();
    magnify->SetInput( cast->GetOutput() );
    magnify->SetInterpolate( 1 );
    magnify->SetInterpolate( 0 );
    int factor = 100;
    magnify->SetMagnificationFactors (factor, factor, 1);

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileName( "/tmp/bla.dcm" );
    writer->SetInput( magnify->GetOutput() );
    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;
}

```

27.91 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;
    }
}

```

27.92 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 Multiframe 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)

```

27.93 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 explicitly 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, gdcml.VL( len(mystr) ) )
68
69     w = gdcml.Writer()
70     w.SetFile( f )
71     w.SetFileName( file2 )
72     if not w.Write():
73         sys.exit(1)

```

27.94 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://gdcml.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 gdcml
28
29 if __name__ == "__main__":
30
31     file1 = sys.argv[1]
32     file2 = sys.argv[2]
33
34     r1 = gdcml.ImageReader()
35     r1.SetFileName( file1 )
36     if not r1.Read():
37         sys.exit(1)
38
39     r2 = gdcml.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 = gdcml.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)

```

27.95 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;
}
```

27.96 MetalmageMD5Activiz.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;
}
}

```

27.97 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("vtkgdcmlib");
    }

    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();

```

```

    }
}

```

27.98 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();
}

```

```
}
```

27.99 MPRViewer2.java

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 gdcml.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcml.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:vtkgdcml.jar:gdcml.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("vtkgdcmlJava");
    }

    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 = oz + (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 );
}
}

```

27.100 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."<TlmapFunctor">
    {
        <Class> "<TlmapFunctor@IceImagePostProcFunctors">

        <ParamBool."<EXECUTE"> { }
        <ParamDouble."<Flip_deg"> { <Precision> 16 14.7378520000000000 }
    }
...

*/
// Below is an attempt to play with the CSAHeader dict:
#if 0
const char gspec[] = "sGRADSPEC.flSensitivityX";
it = mymap.find( gspec );
if( it == mymap.end() ) return 1;
const std::string &dummy = it->second;
std::cout << dummy << std::endl;

const gdcm::CSAHeaderDictEntry &csaentry = csadict.
    GetCSAHeaderDictEntry( gspec );
std::cout << csaentry << std::endl;
#endif

/*
sSliceArray.ucMode -- should be in (1, 2, 4)
enum SeriesMode
{
    ASCENDING    = 0x01,
    DESCENDING   = 0x02,
    INTERLEAVED  = 0x04
};

*/
const char sliceorderstr[] = "sSliceArray.ucMode";
const gdcm::CSAHeaderDictEntry &sliceorder = csadict.
    GetCSAHeaderDictEntry( sliceorderstr );
std::cout << sliceorder << std::endl;

it = mymap.find( sliceorderstr );
if( it == mymap.end() ) return 1;
const std::string &slice_order = it->second;
if( slice_order == "0x1" )
{
    std::cout << "slice_order: ASCENDING" << std::endl;
}
else if( slice_order == "0x2" )
{
    std::cout << "slice_order: DESCENDING" << std::endl;
}
else if( slice_order == "0x4" )

```

```

        {
            std::cout << "slice_order: INTERLEAVED" << std::endl;
        }
    else
    {
        std::cerr << "Impossible: " << slice_order << std::endl;
        return 1;
    }

    return 0;
}

```

27.101 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;
}
}

```

27.102 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=gdcml.DataElement(gdcml.Tag(0x0400,0x0550))
60  des.SetVR(gdcml.VR(gdcml.VR.SQ))
61  des.SetValue(sq.__ref__())
62  des.SetVLToUndefined()
63
64  ds.Insert(des)
65
66  w = gdcml.Writer()
67  w.SetFile( f )
68  w.SetFileName( file2 )
69  if not w.Write():
70      sys.exit(1)

```

27.103 offscreenimage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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();
    windowlevel->SetInput( reader->GetOutput() );
    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();
    actor->SetInput( windowlevel->GetOutput() );

    renderer->AddActor( actor );

```

```

renWin->Render();

vtkWindowToImageFilter *w2if = vtkWindowToImageFilter::New();
w2if->SetInput ( renWin );

vtkPNGWriter *wr = vtkPNGWriter::New();
wr->SetInput( w2if->GetOutput() );
wr->SetFileName ( "offscreenimage.png" );
wr->Write();

reader->Delete();
renWin->Delete();
renderer->Delete();
windowlevel->Delete();
actor->Delete();
w2if->Delete();
wr->Delete();

return 0;
}

```

27.104 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() );
}
{
    gdcmm::Attribute<0x28,0x101> at;
    at.SetFromDataElement( ds.GetDataElement( at.
        GetTag() ) );
    if( at.GetValue() != 8 )
    {
        return 1;
    }
    at.SetValue( 32 );
    ds.Replace( at.GetAsDataElement() );
}
{
    gdcmm::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

{
    gdcmm::Attribute<0x28,0x8> at;
    at.SetFromDataElement( ds.GetDataElement( at.
        GetTag() ) );
    at.SetValue( at.GetValue() * 2 );
    ds.Replace( at.GetAsDataElement() );
}

gdcmm::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:
gdcmm::ImageReader ir;
ir.SetFileName( out );
if(!ir.Read())
{
    return 1;
}
gdcmm::Image &image = ir.GetImage();
unsigned long len = image.GetBufferLength();
const gdcmm::ByteValue *bv = ir.GetFile().GetDataSet().
    GetDataElement( gdcmm::Tag(0x7fe0,0x0010) ).GetByteValue();
if( !bv || len != bv->GetLength() )
{
    return 1;
}
std::cout << bv->GetLength() << " " << len << std::endl;

std::cout << "Success to rewrite image !" << std::endl;
image.Print( std::cout );
return 0;
}

```

27.105 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 e11 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
49 print e11
50 e12 = gdcm.DataElement( ds.GetDataElement( tag2 ) )
51 print e12
52
53 # (0028,1052) DS [-1000] # 6, 1 RescaleIntercept
54 # (0028,1053) DS [1] # 2, 1 RescaleSlope
55
56 e11.SetTag( gdcm.Tag(0x0028,0x1052) )
57 e12.SetTag( gdcm.Tag(0x0028,0x1053) )
58
59 ds.Insert( e11 )
60 ds.Insert( e12 )
61
62 w = gdcm.Writer()
63 w.SetCheckFileMetaInformation( False )
64 w.SetFileName( tmpfile )
65 w.SetFile( reader.GetFile() )
66 if not w.Write():
67     sys.exit(1)
68
69 print "success"

```

27.106 PlaySound.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 PlaySound.py input.dcm
19 """
20
21 import gdcmm
22 import sys
23
24 #filename = "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/audio_from_rafael_sanguinetti.dcm"
25 filename = sys.argv[1]
26 print filename
27
28 r = gdcmm.Reader()
29 r.SetFileName( filename )
30 if not r.Read():
31     sys.exit(1)
32
33 ds = r.GetFile().GetDataSet()
34
35 waveformatag = gdcmm.Tag(0x5400,0x0100)
36 waveformsq = ds.GetDataElement( waveformatag )
37 #print waveformsq
38
39 #print dir(waveformsq)
40
41 items = waveformsq.GetSequenceOfItems()
42
43 if not items.GetNumberOfItems():
44     sys.exit(1)
45
46 item = items.GetItem(1)
47 #print item
48
49 waveformsds = item.GetNestedDataSet()
50 #print waveformsds
51
52 waveformatdatatag = gdcmm.Tag(0x5400,0x01010)
53 waveformdata = waveformsds.GetDataElement( waveformatdatatag )
54
55 #print waveformdata.GetPointer()
56 bv = waveformdata.GetByteValue()
57 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()
```

27.107 pmsct_rgb1.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 ELSCT1/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 (gdcml-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 "gdcmlReader.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlAttribute.h"
#include "gdcmlImageWriter.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;
}

```

27.108 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 file 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") )

```

27.109 PublicDict.cxx

```

/*=====

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

```

```

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Dummy example to show GDCM Dict(s) API (Part 6) + Collected Private Attributes:
 */

#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmCSAHeader.h"
#include "gdcmPrivateTag.h"

int main(int , char *[])
{
    const gdcm::Global& g = gdcm::Global::GetInstance(); // sum of all
        knowledge !
    const gdcm::Dicts &dicts = g.GetDicts();
    const gdcm::Dict &pub = dicts.GetPublicDict(); // Part 6

    //std::cout << pub << std::endl;

    // 3 different ways to access the same information

    // 1. From the public dict only:
    gdcm::Tag patient_name(0x10,0x10);
    const gdcm::DictEntry &entry1 = pub.GetDictEntry(patient_name);
    std::cout << entry1 << std::endl;

    // 2. From all dicts:
    const gdcm::DictEntry &entry2 = dicts.GetDictEntry(patient_name);
    std::cout << entry2 << std::endl;

    // 3. This solution is the most flexible solution as you can request using the same
    // API either a public tag or a private tag
    const char *strowner = 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;
}

```

27.110 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::cout;
    }

    //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(
gdcmm::Tag (0x0010, 0x0010)))
        sqi->GetItem(itemused).GetDataElement(gdcmm::Tag (0x0010, 0x0010))
        .GetValue().Print(strm);
    std::cout << "PATIENT NAME : " << strm.str() << std::endl;

    //PATIENT ID
    strm.str("");
    if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0010, 0x0020)))
        sqi->GetItem(itemused).GetDataElement(gdcmm::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(
gdcmm::Tag (0x0004, 0x1430)))
        sqi->GetItem(itemused).GetDataElement(gdcmm::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(
gdcmm::Tag (0x0020, 0x000d)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::Tag (0x0020, 0x000d)).GetValue().Print(strm);
        std::cout << "          STUDY UID : " << strm.str() << std::endl;

        //STUDY DATE
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x0020)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::Tag (0x0008, 0x0020)).GetValue().Print(strm);
        std::cout << "          STUDY DATE : " << strm.str() << std::endl;

        //STUDY DESCRIPTION
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x1030)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::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(
gdcmm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement(
gdcmm::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(
gdcmm::Tag (0x0020, 0x000e)))
                sqi->GetItem(itemused).GetDataElement(
gdcmm::Tag (0x0020, 0x000e)).GetValue().Print(strm);
            std::cout << "          SERIE UID" << strm.str() << std::endl;

            //SERIE MODALITY
            strm.str("");
            if (sqi->GetItem(itemused).FindDataElement(
gdcmm::Tag (0x0008, 0x0060)))

```

```

        sqi->GetItem(itemused).GetDataElement (
gdcmm::Tag (0x0008, 0x0060)).GetValue().Print(strm);
        std::cout << "                SERIE MODALITY" << strm.str() << std::endl;

        //SERIE DESCRIPTION
        strm.str("");
        if (sqi->GetItem(itemused).FindDataElement (
gdcmm::Tag (0x0008, 0x103e)))
            sqi->GetItem(itemused).GetDataElement (
gdcmm::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 (
gdcmm::Tag (0x0004, 0x1430)))
            sqi->GetItem(itemused).GetDataElement (
gdcmm::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 (
gdcmm::Tag (0x0004, 0x1511)))
                    sqi->GetItem(itemused).GetDataElement (
gdcmm::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 (
gdcmm::Tag (0x0004, 0x1500)))
                    sqi->GetItem(itemused).GetDataElement (
gdcmm::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 (
gdcmm::Tag (0x0004, 0x1430)))
                    sqi->GetItem(itemused).GetDataElement (
gdcmm::Tag (0x0004, 0x1430)).GetValue().Print(strm);

            }
        }
    }
    itemused++;
}
}
return 0;
}

```

27.111 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)) !=
60         gdcm.MediaStorage.MediaStorageDirectoryStorage):
61         # File is not a DICOMDIR
62         print "This file is not a DICOMDIR (Media storage type: %s)" % (str(mediaStorage))
63         quit()
64
65     # Check Media Storage SOP Class
66     if (fileMetaInformation.FindDataElement(gdcm.Tag(0x0002, 0x0002))):
67         sopClassUid = str(fileMetaInformation.GetDataElement(gdcm.Tag(0x0002, 0x0002)).GetValue())
68         # Check SOP UID
69         if (sopClassUid != "1.2.840.10008.1.3.10"):
70             # File is not a DICOMDIR
71             print "This file is not a DICOMDIR"
72         else:
73             # Not present
74             print "Media Storage SOP Class not present"
75             quit()
76
77     # Iterate through the DICOMDIR data set
78     iterator = dataSet.GetDES().begin()
79     while (not iterator.equal(dataSet.GetDES().end())):
80         dataElement = iterator.next()
81
82         # Check the element tag
83         if (dataElement.GetTag() == gdcm.Tag(0x0004, 0x1220)):
84             # 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                                ))
123                            print value
124
125                        # Print study date
126                        if (item.FindDataElement(gdcm.Tag(0x0008, 0x0020))):
127                            value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0020)).GetValue(
128                                ))
129                            print value
130
131                        # Print study description
132                        if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
133                            value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue(
134                                ))
135                            print value
136
137                        # Next
138                        itemNr = itemNr + 1
139                        item = sequence.GetItem(itemNr)
140                        if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
141                            value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).
142                                GetValue())
143
144                        # SERIES
145                        while (value.strip() == "SERIES"):
146                            print value.strip()
147
148                            # Print series UID
149                            if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
150                                value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).
151                                    GetValue())
152                                print value
153
154                            # Print series modality
155                            if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
156                                value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060)).
157                                    GetValue())
158                                print "Modality"
159                                print value
160
161                            # Print series description
162                            if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
163                                value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e)).
164                                    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))).
166         GetValue())
167
168             # IMAGE
169             while (value.strip() == "IMAGE"):
170                 print value.strip()
171
172             # Print image UID
173             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
174                 value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1511))).
175         GetValue())
176
177             print value
178
179             # Next
180             if (itemNr < sequence.GetNumberOfItems()):
181                 itemNr = itemNr + 1
182             else:
183                 break
184
185             item = sequence.GetItem(itemNr)
186             if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
187                 value = str(item.GetDataElement(
gdcm.Tag(0x0004, 0x1430)).GetValue())
188
189             # Next
190             itemNr = itemNr + 1

```

27.112 ReadAndPrintAttributes.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This small example will show how one can read and print
 * a DICOM Attribute using different technique (by tag or by name)
 */

#include "gdcmReader.h"
#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmAttribute.h"
#include "gdcmStringFilter.h"

#include <iostream>

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " input.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];

    // Instantiate the reader:
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {

```

```

    std::cerr << "Could not read: " << filename << std::endl;
    return 1;
}

// The output of gdcm::Reader is a gdcm::File
gdcm::File &file = reader.GetFile();

// the dataset is the the set of element we are interested in:
gdcm::DataSet &ds = file.GetDataSet();

const gdcm::Global& g = gdcm::Global::GetInstance();
const gdcm::Dicts &dicts = g.GetDicts();
const gdcm::Dict &pubdict = dicts.GetPublicDict();

using namespace gdcm;

// In this example we will show why using name to lookup attribute can be
// dangerous.
Tag tPatientName(0x0,0x0);
//const DictEntry &de1 =
pubdict.GetDictEntryByName("Patient Name", tPatientName);

std::cout << "Found: " << tPatientName << std::endl;

// Indeed the attribute could not be found. Since DICOM 2003, Patient Name
// has become Patient's Name.

Tag tPatientsName;
//const DictEntry &de2 =
pubdict.GetDictEntryByName("Patient's Name", tPatientsName);

std::cout << "Found: " << tPatientsName << std::endl;

// Let's try to read an arbitrary DICOM Attribute:
Tag tDoseGridScaling;
//const DictEntry &de3 =
pubdict.GetDictEntryByName("Dose Grid Scaling", tDoseGridScaling);

std::cout << "Found: " << tDoseGridScaling << std::endl;

if( ds.FindDataElement( tDoseGridScaling ) )
{
    gdcm::StringFilter sf;
    sf.SetFile(file);
    std::cout << "Attribute Value as String: " << sf.ToString( tDoseGridScaling ) << std::endl;

    // Let's check the name again:
    std::pair<std::string, std::string> pss
        = sf.ToStringPair( tDoseGridScaling );
    std::cout << "Attribute Name Checked: " << pss.first << std::endl;
    std::cout << "Attribute Value (string): " << pss.second << std::endl;

    //const DataElement &dgs = ds.GetDataElement( tDoseGridScaling );

    // Let's assume for a moment we knew the tag number:
    Attribute<0x3004,0x000e> at;
    assert( at.GetTag() == tDoseGridScaling );
    at.SetFromDataSet( ds );
    // For the sake of long term maintenance, we will not write
    // that this particular attribute is stored as a double. What if
    // a user made a mistake. It is much safer to rely on GDCM internal
    // mechanism to deduce the VR::DS type (represented as a ieee double)
    Attribute<0x3004,0x000e>::ArrayType v = at.
        GetValue();
    std::cout << "DoseGridScaling=" << v << std::endl;
}

return 0;
}

```

27.113 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(0x01,0x42,"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;
}
```

27.114 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);
    }
}
}

```

27.115 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;
}

```

27.116 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;
}

```

27.117 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 preferred
// an import statement along the line of:
// import vtkgdcm.*;
import vtk.*;

/*
 * Usage:
 * export LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:.
 * java -classpath `pwd`/vtkgdcm.jar:/usr/share/java/vtk.jar:. ReadSeriesIntoVTK
 */
public class ReadSeriesIntoVTK
{
    static {
        System.loadLibrary("vtkCommonJava");
        System.loadLibrary("vtkFilteringJava");
        System.loadLibrary("vtkIOJava");
        System.loadLibrary("vtkImagingJava");
        System.loadLibrary("vtkGraphicsJava");
        System.loadLibrary("vtkgdcmJava");
        try {
            System.loadLibrary("vtkRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkHybridJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkHybrid, skipping...");
        }
        try {
            System.loadLibrary("vtkVolumeRenderingJava");
        } catch (Throwable e) {
            System.out.println("cannot load vtkVolumeRendering, skipping...");
        }
    }

    public static void main(String[] args)
    {
        vtkFileOutputWindow outWin = new vtkFileOutputWindow();
        outWin.SetInstance(outWin);
        outWin.SetFileName("MVSVTKViewer.log");

        // See: http://review.source.kitware.com/#change,888
        // vtkWrapJava does not handle static keyword
        // String directory = vtkGDCMTesting.GetGDCMDataRoot();
        vtkGDCMTesting t = new vtkGDCMTesting();
        String directory = t.GetGDCMDataRoot();
        String file0 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq0.dcm";
        String file1 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq1.dcm";
        String file2 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq2.dcm";
        String file3 = directory + "/SIEMENS_MAGNETOM-12-MONO2-FileSeq3.dcm";

        vtkStringArray s = new vtkStringArray();
        System.out.println("adding : " + file0 );
        s.InsertNextValue( file0 );
        s.InsertNextValue( file1 );
        s.InsertNextValue( file2 );
        s.InsertNextValue( file3 );

        vtkGDCMImageReader reader = new vtkGDCMImageReader();
        reader.SetFileNames( s );
        reader.Update();

        System.out.println("Success reading: " + file0 );

        vtkMetaImageWriter writer = new vtkMetaImageWriter();
        writer.DebugOn();
        writer.SetCompression( false );
        writer.SetInput( reader.GetOutput() );
        writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
        writer.Write();

        System.out.println("Success writing: " + writer.GetFileName() );
    }
}

```

```

    }
}

```

27.118 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 );
    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 explicetely 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:
    QApplication 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;
}

```

27.119 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;
    }
}

```

27.120 ReformatFile.cs

This is a C++ example on how to use [gdcm::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 excise 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;
}

```

27.121 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)

```

27.122 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;
}

```

27.123 reslicesphere.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 <sstream>
#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();
    }

```

```

        _imageView->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 );
        gdcmm::Directory d;
        d.Load(fileName);
        gdcmm::Directory::FileNamesType const &files = d.
        GetFileNames();

        gdcmm::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();
        v16->SetInput( _reader->GetOutput() );
        v16->SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
        v16->Update();

        _threshold=vtkImageThreshold::New();
        _threshold->ThresholdByUpper(-3024.0);
        _threshold->ReplaceOutOn();
        _threshold->SetOutValue(0.0);
        _threshold->SetInputConnection(v16->GetOutputPort());

        _shift=vtkImageShiftScale::New();
        _shift->SetShift(0);
        _shift->SetScale(1);

```

```

_shift->SetInputConnection(_threshold->GetOutputPort());

// Initialize the reslice with an axial orientation.
vtkSmartPointer<vtkMatrix4x4> matrix =
    vtkSmartPointer<vtkMatrix4x4>::New();
matrix->Identity();

_transform = vtkTransform::New();
_transform->SetMatrix(matrix);

_reslice = vtkImageReslice::New();
_reslice->SetOutputDimensionality(3);

// PROBLEM:
// The original intent was to connect the same transform
// to the vtkImageReslice and vtkTransformPolyDataFilter,
// but the resulting reslices appear different using the
// vtkTransform as opposed to explicitly setting the
// reslice axes via SetResliceAxes. Also, if the vtkTransform
// is connected and orientated other than axial, the extents
// don't seem to update resulting in VTK believing the slice
// is out of range.

_reslice->SetResliceTransform(_transform);
_reslice->SetResliceAxes(matrix);
_reslice->SetInputConnection(_reader->GetOutputPort());
_reslice->SetInputConnection(_shift->GetOutputPort());

// Create the sphere target shape.
_sphere=vtkSphereSource::New();
_sphere->SetRadius(7.0);
_sphere->SetThetaResolution(16);
_sphere->SetPhiResolution(16);
_sphere->SetCenter(sphereCenter[0], sphereCenter[1], sphereCenter[2]);

_sphereMapper=vtkPolyDataMapper::New();
_sphereMapper->SetInputConnection(_sphere->GetOutputPort());

_sphereActor=vtkActor::New();
_sphereActor->SetMapper(_sphereMapper);
_sphereActor->PickableOff();
_sphereActor->GetProperty()->SetColor(1.0, 0.0, 0.0);
_sphereActor->GetProperty()->SetEdgeColor(1.0, 0.0, 0.0);
_sphereActor->GetProperty()->SetDiffuseColor(1.0, 0.0, 0.0);
_sphereActor->SetVisibility(true);

// Create the cutting pipeline.
// This plane will be positioned in the original image coordinate system.
_plane = vtkPlane::New();
_plane->SetNormal(0.0, 0.0, 1.0);

_cutter = vtkCutter::New();
_cutter->SetInputConnection(_sphere->GetOutputPort());
_cutter->SetCutFunction(_plane);
_cutter->GenerateCutScalarsOn();
_cutter->SetValue(0, 0.5);

// The transform attached to _polyTransform should move the cut
// ROI into the resliced coordinate system, which should be the
// same as the coordinate system of the resliced images.
// PROBLEM: It doesn't.
_polyTransform = vtkTransformPolyDataFilter::New();
_polyTransform->SetTransform(_transform);
_polyTransform->SetInputConnection(_cutter->GetOutputPort());

_ROIMapper = vtkPolyDataMapper2D::New();
_ROIMapper->SetInputConnection(_polyTransform->GetOutputPort());

vtkCoordinate* coordinate = vtkCoordinate::New();
coordinate->SetCoordinateSystemToWorld();
_ROIMapper->SetTransformCoordinate(coordinate);

_ROIActor = vtkActor2D::New();
_ROIActor->SetMapper(_ROIMapper);

// Make sure the cut can be seen, especially the edges.
props=_ROIActor->GetProperty();
props->SetLineWidth(2);
props->SetOpacity(1.0);
props->EdgeVisibilityOn();
props->SetDiffuse(0.8);

```

```

//      props->SetSpecular(0.3);
//      props->SetSpecularPower(20);
//      props->SetRepresentationToSurface();
//      props->SetDiffuseColor(1.0, 0.0, 0.0);
//      props->SetEdgeColor(1.0, 0.0, 0.0);
props->SetColor(1.0, 0.0, 0.0);

_interactor = vtkRenderWindowInteractor::New();

// Create the image viewer and add the actor with the cut ROI.
_imageViewer = vtkImageViewer2::New();
_imageViewer->SetupInteractor(_interactor);
_imageViewer->SetSize(400, 400);
_imageViewer->SetColorWindow(1024);
_imageViewer->SetColorLevel(800);
_imageViewer->SetInputConnection(_reslice->GetOutputPort());
_imageViewer->GetImageActor()->SetOpacity(0.5);

_annotation = vtkTextActor::New();
_annotation->SetTextScaleModeToViewport();
_imageViewer->GetRenderer()->AddActor(_annotation);

// Add the cut shape actor to the renderer.
_imageViewer->GetRenderer()->AddActor(_ROIActor);

// Set up the key handler.
vtkSmartPointer<KeyCallback> callback = vtkSmartPointer<KeyCallback>::New();
callback->SetCallbackData(this);
_interactor->AddObserver(vtkCommand::KeyPressEvent, callback);

_interactor->Initialize();
}

void Start()
{
    _interactor->Start();
}

void ResetOrientation()
{
    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();
    matrix->Identity();

    SetOrientation(matrix);
}

// Make sure the orientation of the vtkImageReslice and
// vtkTransform are in sync.
void SetOrientation(vtkMatrix4x4* matrix)
{
    _reslice->SetResliceAxes(matrix);
    _reslice->Update();

    vtkMatrix4x4* inverse = vtkMatrix4x4::New();
    vtkMatrix4x4::Invert(matrix, inverse);

    _transform->SetMatrix(inverse);
    _transform->Update();
}

// Set the current slice of the current view.
void SetSlice(int slice)
{
    std::stringstream posString;

    double    center[3];
    double    spacing[3];
    double    origin[3];
    double    point[4];
    double    newPoint[4];

    vtkImageData* imageData;
    int newSlice;

    // Try to make sure the extents of the reslice are updated.
    // PROBLEM: It doesn't seem to work when changing the orientation.
    imageData=vtkImageData::SafeDownCast(_reslice->GetOutput());
    imageData->UpdateInformation();

    // 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());
    imageData->UpdateInformation();
    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.
_imageViewer->SetInput(NULL);
_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;
}

```

27.124 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
17 Slope/Intercept
18 and saving the Pixel Spacing in (0028,0030)
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
45                 correct
46                 if( sqmsuidname == 'MR Image Storage' and msuidname == 'Secondary Capture Image Storage' ):
47                     return True
48             # in all other case simply return the currentspacing:
49             return False
50
51 if __name__ == "__main__":
52     r = gdcm.ImageReader()
53     filename = sys.argv[1]
54     r.SetFileName( filename )
55     if not r.Read():
56         sys.exit(1)
57     f = r.GetFile()
58
59     if( CheckSecondaryCaptureObjectIsMRImageStorage(r) ):
60         # Special handling of the spacing:
61         # GDCM 1.2.0 would not rewrite correctly DICOM Object and would always set them as 'Secondary Capture
62         Image Storage'
63         # while we would rather have 'MR Image Storage'
64         gdcm.ImageHelper.SetForcePixelSpacing( True )
65         mrspacing = gdcm.ImageHelper.GetSpacingValue( r.GetFile() )
66         # TODO: I cannot do simply the following:
67         #image.SetSpacing( mrspacing )
68         image.SetSpacing(0, mrspacing[0] )
69         image.SetSpacing(1, mrspacing[1] )
70         image.SetSpacing(2, mrspacing[2] )
71         gdcm.ImageHelper.SetForceRescaleInterceptSlope( True )
72         ris = gdcm.ImageHelper.GetRescaleInterceptSlopeValue(
73             r.GetFile() )
74         image.SetIntercept( ris[0] )
75         image.SetSlope( ris[1] )
76
77     outfilename = sys.argv[2]

```

```

75  w = gdcmm.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)

```

27.125 rle2img.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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 (gdcmm-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 "gdcmmReader.h"
#include "gdcmmPrivateTag.h"
#include "gdcmmAttribute.h"
#include "gdcmmImageWriter.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 'out.rle.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( strcmp( bv->GetPointer(), comprle.c_str(), comprle.size() ) == 0 )
    {
        isrle = true;
    }
    if( strcmp( 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 = "out.rle.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;
}

```

27.126 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 RTSTRUCT 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 )
        writer->SetInput( num, reader->GetOutput(num) );
    //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)
    {
        append->AddInput( reader->GetOutput(i) );
    }

    // Now we'll look at it.
    vtkPolyDataMapper *cubeMapper = vtkPolyDataMapper::New();
    cubeMapper->SetInput( append->GetOutput() );
    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;
}

```

27.127 ScanDirectory.cs

This is a C# example on how to use [gdcm::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;
    }
}

```


27.128 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 r1 = lut.GetLUTLength( LookupTable.LookupTableType.RED );
            byte[] rbuf = new byte[ (int)r1 ];
            long r12 = lut.GetLUT( LookupTable.LookupTableType.RED, rbuf );
            assert r1 == r12;
            long g1 = lut.GetLUTLength( LookupTable.LookupTableType.GREEN );
            byte[] gbuf = new byte[ (int)g1 ];
            long g12 = lut.GetLUT( LookupTable.LookupTableType.GREEN, gbuf );
            assert g1 == g12;
            long b1 = lut.GetLUTLength( LookupTable.LookupTableType.BLUE );
            byte[] bbuf = new byte[ (int)b1 ];
            long b12 = lut.GetLUT( LookupTable.LookupTableType.BLUE, bbuf );
            assert b1 == b12;
            colorModel = new IndexColorModel(8, (int)r1, 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" );
}
}

```

27.129 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)

```

27.130 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;
}
}

```

27.131 SimplePrint.cs

This is a C# example on how to use `gdcmm::SWIGDataSet`

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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/gdcmm/debug-gcc/bin
 * $ mono bin/SimplePrint.exe gdcmmData/012345.002.050.dcm
 */
using System;
using gdcmm;

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;
}

```

27.132 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;
    }
}

```

27.133 SimpleScanner.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed 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 gdcml::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
 * gdcml::System::FileExists to decide whether or not the file actually exist on the disk.
 *
 * It was tested on this particular image:
 * ./SimpleScanner gdcmlData/012345.002.050.dcm
 */

#include "gdcmlScanner.h"

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";

    gdcml::Scanner s;

    const gdcml::Tag tag_array[] = {
        gdcml::Tag(0x8,0x50),
        gdcml::Tag(0x8,0x51),
        gdcml::Tag(0x8,0x60),
    };
    s.AddTag( tag_array[0] );
    s.AddTag( tag_array[1] );
    s.AddTag( tag_array[2] );

    gdcml::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;
}

```

27.134 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.GetFilesNames() );

    std::cout << "Sorter:" << std::endl;
    sorter.Print( std::cout );

    gdcm::Sorter sorter2;
    sorter2.SetSortFunction( mysort_part1 );
    sorter2.StableSort( dir.GetFilesNames() );
    sorter2.SetSortFunction( mysort_part2 );
    sorter2.StableSort( sorter2.GetFilesNames() ); // IMPORTANT
    sorter2.SetSortFunction( mysort_dummy );
    sorter2.StableSort( sorter2.GetFilesNames() ); // 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.GetFilesNames() );

    //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;
}

```

27.135 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

```

27.136 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;
    }
}

```

27.137 StandardizeFiles.cs

This is a C++ example on how to use [gdcm::ImageChangeTransferSyntax](#)

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

    This software is distributed WITHOUT ANY WARRANTY; without even
    the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
    PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * Simple C# example to show how one would 'Standardize' a DICOM File-Set
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/StandardizeFiles.exe input_path output_path
 */
using System;
using gdcm;

public class StandardizeFiles
{
    public static bool ProcessOneFile( string filename, string outfilename )
    {
        PixmapReader reader = new PixmapReader();
        reader.SetFileName( filename );
        if( !reader.Read() )
        {
            System.Console.WriteLine( "Could not read: " + filename );
            return false;
        }

        ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
        change.SetForce( false ); // do we really want to recompress when input is already compressed in same
        //    alg ?
        change.SetCompressIconImage( false ); // Keep it simple
        change.SetTransferSyntax( new TransferSyntax( TransferSyntax.TSType.JPEG2000Lossless ) );
        change.SetInput( reader.GetPixmap() );
        if( !change.Change() )
        {
            System.Console.WriteLine( "Could not change: " + filename );
            return false;
        }

        gdcm.FileMetaInformation fmi = reader.GetFile().GetHeader();
        // The following three lines make sure to regenerate any value:
        fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
        fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
        fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

        PixmapWriter writer = new PixmapWriter();
        writer.SetFileName( outfilename );
        writer.SetFile( reader.GetFile() );
        gdcm.Pixmap pixout = ((PixmapToPixmapFilter)change).GetOutput();
    }
}

```

```

writer.SetPixmap( pixout );
if( !writer.Write() )
{
    System.Console.WriteLine( "Could not write: " + outfilename );
    return false;
}

return true;
}

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My Standardize App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
    System.Console.WriteLine( "Root dir is now: " + gdcm.UIDGenerator.GetRoot() );

    string dir1 = args[0];
    string dir2 = args[1];

    // Check input is valid:
    if( !gdcm.PosixEmulation.FileIsDirectory(dir1) )
    {
        System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
        return 1;
    }
    if( !gdcm.PosixEmulation.FileIsDirectory(dir2) )
    {
        System.Console.WriteLine( "Output directory: " + dir2 + " does not exist. Sorry" );
        return 1;
    }

    Directory d = new Directory();
    uint nfiles = d.Load( dir1, true );
    if(nfiles == 0) return 1;

    // Process all filenames:
    FilenamesType filenames = d.GetFilenames();
    for( uint i = 0; i < nfiles; ++i )
    {
        string filename = filenames[ (int)i ];
        string outfilename = filename.Replace( dir1, dir2 );
        System.Console.WriteLine( "Filename: " + filename );
        System.Console.WriteLine( "Out Filename: " + outfilename );
        if( !ProcessOneFile( filename, outfilename ) )
        {
            System.Console.WriteLine( "Could not process filename: " + filename );
            //return 1;
        }
    }

    return 0;
}

```

27.138 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 del( gdcm::Tag(0x8,0x16) );
del.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms(
    gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( del );

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( outfile, 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;
}

```

27.139 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 << "v1: " << v1 << std::endl;
if( v1 != 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; } ul6;
memcpy(ul6.n, &t, 2 );
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
    (&ul6.tn, sc, 1);
uint16_t tn = ul6.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
    (&ul6.tn, gdcm::SwapCode::BigEndian, 1);
tn = ul6.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;
}

```

27.140 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;
}

```

27.141 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     success = r.Read()
22     #if verbose: print r.GetFile()
23     if verbose: print (r.GetFile().GetDataSet())
24     return success
25
26 if __name__ == "__main__":
27     success = 0

```

```

28     try:
29         filename = os.sys.argv[1]
30         success += TestRead( filename, True )
31     except:
32         # loop over all files:
33         gdcmm.Trace.DebugOff()
34         gdcmm.Trace.WarningOff()
35         t = gdcmm.Testing()
36         nfiles = t.GetNumberOfFileNames()
37         for i in range(0,nfiles):
38             filename = t.GetFileName(i)
39             success += TestRead( filename )
40
41
42     # Test succeed ?
43     sys.exit(success == 0)

```

27.142 threadgdcmm.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed 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 "gdcmmReader.h"
#include "gdcmmImageReader.h"
#include "gdcmmDirectory.h"
#include "gdcmmSystem.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;

        gdcmm::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;

#ifdef 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);

    switch( pixeltype )
    {
        case gdcm::PixelFormat::INT8:
#ifdef (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 gdcmm::PixelFormat::UINT32:
    output->SetScalarType ( VTK_UNSIGNED_INT );
    break;
default:
    assert(0);
}

output->SetNumberOfScalarComponents ( pixeltype.GetSamplesPerPixel() );

output->AllocateScalars();
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();
writer->SetInput( output );
writer->SetFileName( "/tmp/threadgdcmm.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;
}

```

27.143 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;
}

```

27.144 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 );
    }
    return 0;
}

```

27.145 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;
    }
}

```

```

gdcmm::Sorter sorter;
sorter.SetSortFunction( mysort1 );
sorter.StableSort( 12 );

sorter.SetSortFunction( mysort2 );
sorter.StableSort( sorter.GetFilesNames() );

sorter.SetSortFunction( mysort3 );
sorter.StableSort( sorter.GetFilesNames() );

sorter.SetSortFunction( mysort4 );
sorter.StableSort( sorter.GetFilesNames() );

//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;
{
    gdcmm::Scanner s;
    s.AddTag( gdcmm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcmm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( d.GetFilesNames() );

    //s.Print( std::cout );

    const gdcmm::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;
}

gdcmm::Directory::FileNamesType sorted_files = sorter.
    GetFilesNames();

// Which means we can take nvalues files at a time and execute gdcmm::IPPSorter on it:
gdcmm::IPPSorter ippsorter;
gdcmm::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;
}

```

27.146 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://gdcmm.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

- ~ASN1
 - gdcmm::ASN1, [162](#)
- ~AnonymizeEvent
 - gdcmm::AnonymizeEvent, [147](#)
- ~Anonymizer
 - gdcmm::Anonymizer, [150](#)
- ~Attribute
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [179](#)
- ~AudioCodec
 - gdcmm::AudioCodec, [189](#)
- ~Base64
 - gdcmm::Base64, [190](#)
- ~BasePDU
 - gdcmm::network::BasePDU, [196](#)
- ~BaseRootQuery
 - gdcmm::BaseRootQuery, [198](#)
- ~Bitmap
 - gdcmm::Bitmap, [208](#)
- ~BitmapToBitmapFilter
 - gdcmm::BitmapToBitmapFilter, [214](#)
- ~BoxRegion
 - gdcmm::BoxRegion, [217](#)
- ~ByteSwapFilter
 - gdcmm::ByteSwapFilter, [221](#)
- ~ByteValue
 - gdcmm::ByteValue, [223](#)
- ~CSAHeader
 - gdcmm::CSAHeader, [260](#)
- ~Coder
 - gdcmm::Coder, [238](#)
- ~Command
 - gdcmm::Command, [243](#)
- ~CommandDataSet
 - gdcmm::CommandDataSet, [245](#)
- ~CryptographicMessageSyntax
 - gdcmm::CryptographicMessageSyntax, [253](#)
- ~Curve
 - gdcmm::Curve, [271](#)
- ~DICOMDIRGenerator
 - gdcmm::DICOMDIRGenerator, [302](#)
- ~DataEvent
 - gdcmm::DataEvent, [284](#)
- ~DataSetEvent
 - gdcmm::DataSetEvent, [293](#)
- ~Decoder
 - gdcmm::Decoder, [294](#)
- ~Defs
 - gdcmm::Defs, [297](#)
- ~DeltaEncodingCodec
 - gdcmm::DeltaEncodingCodec, [299](#)
- ~DictConverter
 - gdcmm::DictConverter, [306](#)
- ~DictPrinter
 - gdcmm::DictPrinter, [311](#)
- ~Dicts
 - gdcmm::Dicts, [313](#)
- ~DirectionCosines
 - gdcmm::DirectionCosines, [317](#)
- ~Directory
 - gdcmm::Directory, [319](#)
- ~Dumper
 - gdcmm::Dumper, [324](#)
- ~Element
 - gdcmm::Element< TVR, VM::VM1_n >, [331](#)
- ~Event
 - gdcmm::Event, [349](#)
- ~Exception
 - gdcmm::Exception, [351](#)
- ~File
 - gdcmm::File, [359](#)
- ~FileAnonymizer
 - gdcmm::FileAnonymizer, [362](#)
- ~FileDerivation
 - gdcmm::FileDerivation, [364](#)
- ~FileExplicitFilter
 - gdcmm::FileExplicitFilter, [366](#)
- ~FileMetaInformation
 - gdcmm::FileMetaInformation, [370](#)
- ~FilenameGenerator
 - gdcmm::FilenameGenerator, [376](#)
- ~Global
 - gdcmm::Global, [388](#)
- ~GroupDict
 - gdcmm::GroupDict, [390](#)
- ~IPPSorter
 - gdcmm::IPPSorter, [448](#)
- ~IconImageFilter
 - gdcmm::IconImageFilter, [392](#)
- ~IconImageGenerator

- gdcm::IconImageGenerator, [394](#)
- ~Image
 - gdcm::Image, [398](#)
- ~ImageApplyLookupTable
 - gdcm::ImageApplyLookupTable, [402](#)
- ~ImageChangePhotometricInterpretation
 - gdcm::ImageChangePhotometricInterpretation, [405](#)
- ~ImageChangePlanarConfiguration
 - gdcm::ImageChangePlanarConfiguration, [408](#)
- ~ImageChangeTransferSyntax
 - gdcm::ImageChangeTransferSyntax, [411](#)
- ~ImageCodec
 - gdcm::ImageCodec, [415](#)
- ~ImageConverter
 - gdcm::ImageConverter, [419](#)
- ~ImageFragmentSplitter
 - gdcm::ImageFragmentSplitter, [421](#)
- ~ImageReader
 - gdcm::ImageReader, [427](#)
- ~ImageRegionReader
 - gdcm::ImageRegionReader, [430](#)
- ~ImageToImageFilter
 - gdcm::ImageToImageFilter, [433](#)
- ~ImageWriter
 - gdcm::ImageWriter, [435](#)
- ~JPEG12Codec
 - gdcm::JPEG12Codec, [456](#)
- ~JPEG16Codec
 - gdcm::JPEG16Codec, [458](#)
- ~JPEG2000Codec
 - gdcm::JPEG2000Codec, [460](#)
- ~JPEG8Codec
 - gdcm::JPEG8Codec, [463](#)
- ~JPEGCodec
 - gdcm::JPEGCodec, [466](#)
- ~JPEGLSCodec
 - gdcm::JPEGLSCodec, [470](#)
- ~KAKADUCodec
 - gdcm::KAKADUCodec, [473](#)
- ~LookupTable
 - gdcm::LookupTable, [478](#)
- ~MD5
 - gdcm::MD5, [485](#)
- ~MemberCommand
 - gdcm::MemberCommand, [495](#)
- ~MeshPrimitive
 - gdcm::MeshPrimitive, [499](#)
- ~ModuleEntry
 - gdcm::ModuleEntry, [504](#)
- ~Object
 - gdcm::Object, [516](#)
- ~Orientation
 - gdcm::Orientation, [518](#)
- ~Overlay
 - gdcm::Overlay, [522](#)
- ~PDBHeader
 - gdcm::PDBHeader, [535](#)
- ~PDFCodec
 - gdcm::PDFCodec, [537](#)
- ~PGXCodec
 - gdcm::PGXCodec, [541](#)
- ~PNMCodec
 - gdcm::PNMCodec, [563](#)
- ~PVRGCodec
 - gdcm::PVRGCodec, [584](#)
- ~ParseException
 - gdcm::ParseException, [526](#)
- ~Parser
 - gdcm::Parser, [528](#)
- ~PixelFormat
 - gdcm::PixelFormat, [546](#)
- ~Pixmap
 - gdcm::Pixmap, [551](#)
- ~PixmapReader
 - gdcm::PixmapReader, [555](#)
- ~PixmapToPixmapFilter
 - gdcm::PixmapToPixmapFilter, [557](#)
- ~PixmapWriter
 - gdcm::PixmapWriter, [560](#)
- ~Preamble
 - gdcm::Preamble, [565](#)
- ~Printer
 - gdcm::Printer, [576](#)
- ~PrivateDict
 - gdcm::PrivateDict, [578](#)
- ~ProgressEvent
 - gdcm::ProgressEvent, [582](#)
- ~PythonFilter
 - gdcm::PythonFilter, [585](#)
- ~QueryBase
 - gdcm::QueryBase, [587](#)
- ~RAWCodec
 - gdcm::RAWCodec, [598](#)
- ~RLECodec
 - gdcm::RLECodec, [611](#)
- ~Reader
 - gdcm::Reader, [602](#)
- ~Region
 - gdcm::Region, [605](#)
- ~Rescaler
 - gdcm::Rescaler, [608](#)
- ~SHA1
 - gdcm::SHA1, [651](#)
- ~Scanner
 - gdcm::Scanner, [618](#)
- ~Segment
 - gdcm::Segment, [623](#)
- ~SegmentReader

- gdcmm::SegmentReader, 628
- ~SegmentWriter
 - gdcmm::SegmentWriter, 631
- ~SegmentedPaletteColorLookupTable
 - gdcmm::SegmentedPaletteColorLookupTable, 626
- ~SerieHelper
 - gdcmm::SerieHelper, 643
- ~ServiceClassUser
 - gdcmm::ServiceClassUser, 648
- ~SimpleMemberCommand
 - gdcmm::SimpleMemberCommand, 654
- ~SimpleSubjectWatcher
 - gdcmm::SimpleSubjectWatcher, 656
- ~SmartPointer
 - gdcmm::SmartPointer, 658
- ~Sorter
 - gdcmm::Sorter, 664
- ~Spacing
 - gdcmm::Spacing, 666
- ~SplitMosaicFilter
 - gdcmm::SplitMosaicFilter, 668
- ~StreamImageReader
 - gdcmm::StreamImageReader, 671
- ~StreamImageWriter
 - gdcmm::StreamImageWriter, 676
- ~StringFilter
 - gdcmm::StringFilter, 683
- ~Subject
 - gdcmm::Subject, 686
- ~Surface
 - gdcmm::Surface, 690
- ~SurfaceReader
 - gdcmm::SurfaceReader, 698
- ~SurfaceWriter
 - gdcmm::SurfaceWriter, 700
- ~Table
 - gdcmm::Table, 708
- ~TableEntry
 - gdcmm::TableEntry, 709
- ~TableReader
 - gdcmm::TableReader, 710
- ~TableRow
 - gdcmm::network::TableRow, 712
- ~TagPath
 - gdcmm::TagPath, 719
- ~Testing
 - gdcmm::Testing, 721
- ~Trace
 - gdcmm::Trace, 725
- ~Transition
 - gdcmm::network::Transition, 733
- ~ULAction
 - gdcmm::network::ULAction, 758
- ~ULBasicCallback
 - gdcmm::network::ULBasicCallback, 792
- ~ULConnection
 - gdcmm::network::ULConnection, 794
- ~ULConnectionCallback
 - gdcmm::network::ULConnectionCallback, 796
- ~ULConnectionManager
 - gdcmm::network::ULConnectionManager, 800
- ~ULEvent
 - gdcmm::network::ULEvent, 801
- ~ULWritingCallback
 - gdcmm::network::ULWritingCallback, 804
- ~UserInformation
 - gdcmm::network::UserInformation, 812
- ~Validate
 - gdcmm::Validate, 814
- ~Value
 - gdcmm::Value, 815
- ~Version
 - gdcmm::Version, 817
- ~Writer
 - gdcmm::Writer, 887
- ~XMLDictReader
 - gdcmm::XMLDictReader, 890
- ~XMLPrivateDictReader
 - gdcmm::XMLPrivateDictReader, 892
- ~vtkGDCMImageReader
 - vtkGDCMImageReader, 835
- ~vtkGDCMImageWriter
 - vtkGDCMImageWriter, 841
- ~vtkGDCMMedicalImageProperties
 - vtkGDCMMedicalImageProperties, 844
- ~vtkGDCMPolyDataReader
 - vtkGDCMPolyDataReader, 847
- ~vtkGDCMPolyDataWriter
 - vtkGDCMPolyDataWriter, 850
- ~vtkGDCMTesting
 - vtkGDCMTesting, 853
- ~vtkGDCMThreadedImageReader
 - vtkGDCMThreadedImageReader, 855
- ~vtkGDCMThreadedImageReader2
 - vtkGDCMThreadedImageReader2, 857
- ~vtkImageColorViewer
 - vtkImageColorViewer, 862
- ~vtkImageMapToColors16
 - vtkImageMapToColors16, 867
- ~vtkImageMapToWindowLevelColors2
 - vtkImageMapToWindowLevelColors2, 870
- ~vtkImagePlanarComponentsToComponents
 - vtkImagePlanarComponentsToComponents, 872
- ~vtkImageRGBToYBR
 - vtkImageRGBToYBR, 874
- ~vtkImageYBRToRGB
 - vtkImageYBRToRGB, 876
- ~vtkLookupTable16

- vtkLookupTable16, [877](#)
- ~vtkRTStructSetProperties
 - vtkRTStructSetProperties, [880](#)
- AE
 - gdcm::VR, [826](#)
- AES128_CIPHER
 - gdcm::CryptographicMessageSyntax, [253](#)
- AES192_CIPHER
 - gdcm::CryptographicMessageSyntax, [253](#)
- AES256_CIPHER
 - gdcm::CryptographicMessageSyntax, [253](#)
- ALGOType_END
 - gdcm::Segment, [623](#)
- ARGB
 - gdcm::PhotometricInterpretation, [543](#)
- AS
 - gdcm::VR, [826](#)
- AT
 - gdcm::VR, [826](#)
- AUTOMATIC
 - gdcm::Segment, [623](#)
- AXIAL
 - gdcm::Orientation, [518](#)
- AAAbortPDU
 - gdcm::network::AAAbortPDU, [134](#)
- AAAssociateACPDU
 - gdcm::network::AAAssociateACPDU, [137](#)
 - gdcm::network::AAAssociateRQPDU, [143](#)
- AAAssociateRJPDU
 - gdcm::network::AAAssociateRJPDU, [139](#)
- AAAssociateRQPDU
 - gdcm::network::AAAssociateACPDU, [137](#)
 - gdcm::network::AAAssociateRQPDU, [141](#)
- AEComp
 - gdcm, [117](#)
- ALGOType
 - gdcm::Segment, [623](#)
- ARTIMTimer
 - gdcm::network::ARTIMTimer, [161](#)
- AReleaseRPPDU
 - gdcm::network::AReleaseRPPDU, [158](#)
- AReleaseRQPDU
 - gdcm::network::AReleaseRQPDU, [159](#)
- ASComp
 - gdcm, [117](#)
- ASN1
 - gdcm::ASN1, [162](#)
- AbstractSyntax
 - gdcm::network::AbstractSyntax, [145](#)
- ActiveComponent
 - vtkImageMapToColors16, [868](#)
- Add
 - gdcm::GroupDict, [390](#)
- AddAcceptedPresentationContext
 - gdcm::network::ULConnection, [794](#)
- AddCSAHeaderDictEntry
 - gdcm::CSAHeaderDict, [263](#)
- AddContourReferencedFrameOfReference
 - vtkRTStructSetProperties, [880](#)
- AddDerivationDescription
 - gdcm::FileDerivation, [364](#)
- AddDictEntry
 - gdcm::Dict, [304](#)
 - gdcm::PrivateDict, [578](#)
- AddFile
 - gdcm::FileSet, [378](#)
 - gdcm::SerieHelper, [644](#)
- AddFileName
 - gdcm::SerieHelper, [644](#)
- AddFragment
 - gdcm::SequenceOfFragments, [634](#)
- AddGroupLength
 - gdcm::DictConverter, [306](#)
- AddIOD
 - gdcm::IODs, [446](#)
- AddIODEntry
 - gdcm::IOD, [442](#)
- AddImageDirectoryRecord
 - gdcm::DICOMDIRGenerator, [302](#)
- AddInput
 - vtkImageColorViewer, [862](#)
- AddInputConnection
 - vtkImageColorViewer, [862](#)
- AddItem
 - gdcm::SequenceOfItems, [639](#)
- AddMacro
 - gdcm::Macros, [483](#)
 - gdcm::Module, [502](#)
- AddMacroEntry
 - gdcm::Macro, [481](#)
- AddModule
 - gdcm::Modules, [506](#)
- AddModuleEntry
 - gdcm::Module, [502](#)
 - gdcm::NestedModuleEntries, [513](#)
- AddObserver
 - gdcm::Subject, [686](#)
- AddPatientDirectoryRecord
 - gdcm::DICOMDIRGenerator, [302](#)
- AddPresentationContext
 - gdcm::network::AAAssociateRQPDU, [142](#)
 - gdcm::PresentationContextGenerator, [569](#)
- AddPresentationContextAC
 - gdcm::network::AAAssociateACPDU, [137](#)
- AddPresentationDataValue
 - gdcm::network::PDataTFPDU, [531](#)
- AddPrimitiveData

- gdcmm::MeshPrimitive, 499
- AddPrivateTag
 - gdcmm::Scanner, 618
- AddPurposeOfReferenceCodeSequence
 - gdcmm::FileDerivation, 364
- AddQueryDataSet
 - gdcmm::BaseRootQuery, 199
- AddReference
 - gdcmm::FileDerivation, 364
- AddReferencedFrameOfReference
 - vtkRTStructSetProperties, 881
- AddRestriction
 - gdcmm::SerieHelper, 644
- AddRoleSelectionSub
 - gdcmm::network::UserInformation, 812
- AddSOPClassExtendedNegociationSub
 - gdcmm::network::UserInformation, 812
- AddSegment
 - gdcmm::SegmentWriter, 631
- AddSelect
 - gdcmm::Sorter, 664
- AddSeriesDirectoryRecord
 - gdcmm::DICOMDIRGenerator, 302
- AddSkipTag
 - gdcmm::Scanner, 618
- AddSourceImageSequence
 - gdcmm::FileDerivation, 364
- AddStructureSetROI
 - vtkRTStructSetProperties, 881
- AddStructureSetROIObservation
 - vtkRTStructSetProperties, 881
- AddStudyDirectoryRecord
 - gdcmm::DICOMDIRGenerator, 302
- AddSurface
 - gdcmm::Segment, 623
- AddTag
 - gdcmm::Scanner, 618
- AddTransferSyntax
 - gdcmm::network::PresentationContextRQ, 571
 - gdcmm::PresentationContext, 566
- AffectedSOPClassUID
 - gdcmm::network::CEchoRQ, 227
- Allocate
 - gdcmm::LookupTable, 478
- AmbulatoryECGWaveformStorage
 - gdcmm::MediaStorage, 489
 - gdcmm::UIDs, 745
- AnatomicRegion
 - gdcmm::Segment, 624
- AnonymizeEvent
 - gdcmm::Anonymizer, 147
- Anonymizer
 - gdcmm::Anonymizer, 150
- Append
 - gdcmm::Global, 388
- AppendImplementationClassUID
 - gdcmm::FileMetaInformation, 370
- ApplicationContext
 - gdcmm::network::ApplicationContext, 154
- Apply
 - gdcmm::ImageApplyLookupTable, 402
- ApplyInverseVideo
 - vtkGDCMImageReader, 838
- ApplyLookupTable
 - vtkGDCMImageReader, 838
- ApplyPlanarConfiguration
 - vtkGDCMImageReader, 838
- ApplyShiftScale
 - vtkGDCMImageReader, 838
- ApplyYBRToRGB
 - vtkGDCMImageReader, 838
- AreOverlaysInPixelData
 - gdcmm::Bitmap, 208
 - gdcmm::Pixmap, 551
- Area
 - gdcmm::BoxRegion, 217
 - gdcmm::Region, 605
- ArrayIncludeMacrosType
 - gdcmm::Macro, 481
 - gdcmm::Module, 502
- ArrayType
 - gdcmm::Attribute, 165
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 179
- AsynchronousOperationsWindowSub
 - gdcmm::network::AsynchronousOperationsWindow-Sub, 163
- Attribute
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, 179
 - gdcmm::terminal, 131
- Audio
 - gdcmm::MediaStorage, 490
- AudioSRStorageTrialRetired
 - gdcmm::UIDs, 746
- AudioCodec
 - gdcmm::AudioCodec, 189
- AutoPixelMinMax
 - gdcmm::IconImageGenerator, 394
- BLUE
 - gdcmm::LookupTable, 478
- BALCPPProtect
 - gdcmm::Anonymizer, 150
- backslash
 - gdcmm, 119

- BadBigEndian
 - gdcm::SwapCode, [701](#)
- BadLittleEndian
 - gdcm::SwapCode, [701](#)
- Base64
 - gdcm::Base64, [190](#)
- BaseRootQuery
 - gdcm::BaseRootQuery, [198](#)
- BasicAnnotationBoxSOPClass
 - gdcm::UIDs, [744](#)
- BasicColorImageBoxSOPClass
 - gdcm::UIDs, [744](#)
- BasicColorPrintManagementMetaSOPClass
 - gdcm::UIDs, [744](#)
- BasicFilmBoxSOPClass
 - gdcm::UIDs, [744](#)
- BasicFilmSessionSOPClass
 - gdcm::UIDs, [744](#)
- BasicGrayscaleImageBoxSOPClass
 - gdcm::UIDs, [744](#)
- BasicGrayscalePrintManagementMetaSOPClass
 - gdcm::UIDs, [744](#)
- BasicPrintImageOverlayBoxSOPClassRetired
 - gdcm::UIDs, [745](#)
- BasicStudyContentNotificationSOPClassRetired
 - gdcm::UIDs, [744](#)
- BasicTextSR
 - gdcm::MediaStorage, [489](#)
- BasicTextSRStorage
 - gdcm::UIDs, [746](#)
- BasicVoiceAudioWaveformStorage
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [745](#)
- BasicApplicationLevelConfidentialityProfile
 - gdcm::Anonymizer, [150](#)
- BasicCodedEntry
 - gdcm::SegmentHelper::BasicCodedEntry, [202](#)
- BasicOffsetTable
 - gdcm::BasicOffsetTable, [204](#)
- Begin
 - gdcm::CSAHeaderDict, [263](#)
 - gdcm::DataSet, [287](#)
 - gdcm::Dict, [304](#)
 - gdcm::IODs, [446](#)
 - gdcm::Scanner, [618](#)
 - gdcm::SequenceOfFragments, [634](#)
 - gdcm::SequenceOfItems, [639](#)
- BigEndian
 - gdcm::SwapCode, [701](#)
- BitSample
 - gdcm::JPEGCodec, [467](#)
 - gdcm::LookupTable, [480](#)
- Bitmap
 - gdcm::Bitmap, [208](#)
 - gdcm::JPEG2000Codec, [461](#)
 - gdcm::PixelFormat, [549](#)
- BitmapToBitmapFilter
 - gdcm::BitmapToBitmapFilter, [214](#)
- black
 - gdcm::terminal, [131](#)
- BlendingSoftcopyPresentationStateStorageSOPClass
 - gdcm::UIDs, [746](#)
- blink
 - gdcm::terminal, [131](#)
- blue
 - gdcm::terminal, [131](#)
- BoundingBox
 - gdcm::BoxRegion, [217](#)
- BoxRegion
 - gdcm::BoxRegion, [217](#)
- BreakConnection
 - gdcm::network::ULConnectionManager, [800](#)
- BreakConnectionNow
 - gdcm::network::ULConnectionManager, [800](#)
- BreastImagingRelevantPatientInformationQuery
 - gdcm::UIDs, [747](#)
- BreastTomosynthesisImageStorage
 - gdcm::MediaStorage, [490](#)
 - gdcm::UIDs, [749](#)
- bright
 - gdcm::terminal, [131](#)
- Build
 - vtkLookupTable16, [877](#)
- ByteBuffer
 - gdcm::ByteBuffer, [219](#)
- ByteSwap
 - gdcm::ByteSwapFilter, [221](#)
- ByteSwapFilter
 - gdcm::ByteSwapFilter, [221](#)
- ByteValue
 - gdcm::ByteValue, [223](#)
- bytes
 - gdcm::Tag, [718](#)
- C_CANCEL_RQ
 - gdcm::network::DIMSE, [316](#)
- C_ECHO_RQ
 - gdcm::network::DIMSE, [315](#)
- C_ECHO_RSP
 - gdcm::network::DIMSE, [315](#)
- C_FIND_RQ
 - gdcm::network::DIMSE, [315](#)
- C_FIND_RSP
 - gdcm::network::DIMSE, [315](#)
- C_GET_RQ
 - gdcm::network::DIMSE, [315](#)
- C_GET_RSP
 - gdcm::network::DIMSE, [315](#)

- C_MOVE_RQ
 - gdcm::network::DIMSE, [315](#)
- C_MOVE_RSP
 - gdcm::network::DIMSE, [315](#)
- C_STORE_RQ
 - gdcm::network::DIMSE, [315](#)
- C_STORE_RSP
 - gdcm::network::DIMSE, [315](#)
- CALIBRATED
 - gdcm::Spacing, [666](#)
- CMYK
 - gdcm::PhotometricInterpretation, [543](#)
- CONDENSED_STYLE
 - gdcm::Printer, [576](#)
- CONSOLE
 - gdcm::terminal, [131](#)
- CORONAL
 - gdcm::Orientation, [518](#)
- CS
 - gdcm::VR, [826](#)
- CSANonImageStorage
 - gdcm::MediaStorage, [489](#)
- CT_private_ELE
 - gdcm::TransferSyntax, [729](#)
- CTImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- CEcho
 - gdcm::CompositeNetworkFunctions, [248](#)
- CFind
 - gdcm::CompositeNetworkFunctions, [248](#)
- CM
 - gdcm::SegmentHelper::BasicCodedEntry, [202](#)
- cMaxEventID
 - gdcm::network, [129](#)
- cMaxStateID
 - gdcm::network, [129](#)
- CMove
 - gdcm::CompositeNetworkFunctions, [248](#)
- CSAElement
 - gdcm::CSAElement, [255](#)
- CSAHeader
 - gdcm::CSAHeader, [260](#)
 - gdcm::DataSet, [291](#)
- CSAHeaderDict
 - gdcm::CSAHeaderDict, [263](#)
- CSAHeaderDictEntry
 - gdcm::CSAHeaderDictEntry, [265](#)
- CSAHeaderType
 - gdcm::CSAHeader, [260](#)
- CSComp
 - gdcm, [117](#)
- CSD
 - gdcm::SegmentHelper::BasicCodedEntry, [202](#)
- CSV
 - gdcm::SegmentHelper::BasicCodedEntry, [202](#)
- CStore
 - gdcm::CompositeNetworkFunctions, [249](#)
- CV
 - gdcm::SegmentHelper::BasicCodedEntry, [202](#)
- CanCode
 - gdcm::AudioCodec, [189](#)
 - gdcm::Coder, [238](#)
 - gdcm::ImageCodec, [415](#)
 - gdcm::JPEG2000Codec, [460](#)
 - gdcm::JPEGCodec, [466](#)
 - gdcm::JPEGLSCodec, [470](#)
 - gdcm::KAKADUCodec, [473](#)
 - gdcm::PDFCodec, [537](#)
 - gdcm::PGXCodec, [541](#)
 - gdcm::PNMCodec, [563](#)
 - gdcm::PVRGCodec, [584](#)
 - gdcm::RAWCodec, [598](#)
 - gdcm::RLECodec, [611](#)
- CanDecode
 - gdcm::AudioCodec, [190](#)
 - gdcm::Decoder, [295](#)
 - gdcm::DeltaEncodingCodec, [299](#)
 - gdcm::ImageCodec, [415](#)
 - gdcm::JPEG2000Codec, [460](#)
 - gdcm::JPEGCodec, [466](#)
 - gdcm::JPEGLSCodec, [470](#)
 - gdcm::KAKADUCodec, [473](#)
 - gdcm::PDFCodec, [537](#)
 - gdcm::PGXCodec, [541](#)
 - gdcm::PNMCodec, [563](#)
 - gdcm::PVRGCodec, [584](#)
 - gdcm::RAWCodec, [599](#)
 - gdcm::RLECodec, [611](#)
- CanDisplay
 - gdcm::VR, [827](#)
- CanEmptyTag
 - gdcm::Anonymizer, [151](#)
- CanRead
 - gdcm::Reader, [602](#)
- CanReadFile
 - vtkGDCMImageReader, [835](#)
- CanReadImage
 - gdcm::StreamImageReader, [671](#)
- CanStoreLossy
 - gdcm::TransferSyntax, [729](#)
- CanWriteFile
 - gdcm::StreamImageWriter, [676](#)
- CardiacElectrophysiologyWaveformStorage
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [745](#)
- CardiacRelevantPatientInformationQuery
 - gdcm::UIDs, [748](#)

- Change
 - gdcm::FileExplicitFilter, [366](#)
 - gdcm::ImageChangePhotometricInterpretation, [405](#)
 - gdcm::ImageChangePlanarConfiguration, [408](#)
 - gdcm::ImageChangeTransferSyntax, [411](#)
- ChangeFMI
 - gdcm::FileExplicitFilter, [367](#)
- ChangeMonochrome
 - gdcm::ImageChangePhotometricInterpretation, [405](#)
- CharacterDataHandler
 - gdcm::TableReader, [710](#)
 - gdcm::XMLDictReader, [890](#)
 - gdcm::XMLPrivateDictReader, [892](#)
- CheckEvent
 - gdcm::AnonymizeEvent, [147](#)
 - gdcm::DataEvent, [284](#)
 - gdcm::DataSetEvent, [293](#)
 - gdcm::Event, [349](#)
 - gdcm::ProgressEvent, [582](#)
- CheckFileMetaInformationOff
 - gdcm::Writer, [887](#)
- CheckFileMetaInformationOn
 - gdcm::Writer, [887](#)
- ChestCADSRStorage
 - gdcm::UIDs, [747](#)
- CipherTypes
 - gdcm::CryptographicMessageSyntax, [253](#)
- Clear
 - gdcm::Bitmap, [208](#)
 - gdcm::ByteValue, [223](#)
 - gdcm::DataElement, [276](#)
 - gdcm::DataSet, [287](#)
 - gdcm::IOD, [442](#)
 - gdcm::IODs, [446](#)
 - gdcm::Item, [452](#)
 - gdcm::LookupTable, [478](#)
 - gdcm::Macro, [481](#)
 - gdcm::Macros, [483](#)
 - gdcm::Module, [502](#)
 - gdcm::Modules, [506](#)
 - gdcm::Preamble, [565](#)
 - gdcm::SequenceOfFragments, [634](#)
 - gdcm::SequenceOfItems, [639](#)
 - gdcm::SerieHelper, [644](#)
 - gdcm::Value, [815](#)
 - vtkGDCMMedicalImageProperties, [844](#)
 - vtkRTStructSetProperties, [881](#)
- ClearSkipTags
 - gdcm::Scanner, [618](#)
- ClearTags
 - gdcm::Scanner, [618](#)
- Clone
 - gdcm::BoxRegion, [217](#)
 - gdcm::Region, [605](#)
- Code
 - gdcm::Coder, [238](#)
 - gdcm::JPEG2000Codec, [461](#)
 - gdcm::JPEGCodec, [466](#)
 - gdcm::JPEGLSCodec, [470](#)
 - gdcm::KAKADUCodec, [473](#)
 - gdcm::PVRGCodec, [584](#)
 - gdcm::RAWCodec, [599](#)
 - gdcm::RLECodec, [611](#)
- CodeString
 - gdcm::CodeString, [241](#)
- Color
 - gdcm::terminal, [131](#)
- ColorSoftcopyPresentationStateStorageSOPClass
 - gdcm::UIDs, [746](#)
- ColorArray
 - gdcm::SurfaceHelper, [694](#)
- Command
 - gdcm::Command, [243](#)
- CommandDataSet
 - gdcm::CommandDataSet, [245](#)
- CommandTypes
 - gdcm::network::DIMSE, [315](#)
- CompOperators
 - gdcm, [118](#)
- Compatible
 - gdcm::VM, [823](#)
 - gdcm::VR, [827](#)
- Component
 - gdcm::PersonName, [539](#)
- ComprehensiveSR
 - gdcm::MediaStorage, [489](#)
- ComprehensiveSRStorage
 - gdcm::UIDs, [746](#)
- ComprehensiveSRStorageTrialRetired
 - gdcm::UIDs, [746](#)
- CompressionTypes
 - vtkGDCMImageWriter, [841](#)
- Compute
 - gdcm::MD5, [485](#)
 - gdcm::SHA1, [651](#)
- ComputeBoundingBox
 - gdcm::BoxRegion, [217](#)
 - gdcm::Region, [606](#)
- ComputeBufferLength
 - gdcm::ImageRegionReader, [430](#)
- ComputeByteLength
 - gdcm::SequenceOfFragments, [634](#)
- ComputeDataElement
 - gdcm::DataSet, [287](#)
- ComputeDataSetMediaStorageSOPClass
 - gdcm::FileMetaInformation, [370](#)
- ComputeDataSetTransferSyntax
 - gdcm::FileMetaInformation, [370](#)

- ComputeDistAlongNormal
 - gdcm::DirectionCosines, [317](#)
- ComputeFile
 - gdcm::MD5, [485](#)
 - gdcm::SHA1, [652](#)
- ComputeFileMD5
 - gdcm::Testing, [721](#)
- ComputeGroupLength
 - gdcm::DataSet, [287](#)
- ComputeInterceptSlopePixelType
 - gdcm::Rescaler, [608](#)
- ComputeLength
 - gdcm::SequenceOfFragments, [634](#)
 - gdcm::SequenceOfItems, [639](#)
- ComputeLossyFlag
 - gdcm::Bitmap, [208](#)
- ComputeMD5
 - gdcm::Testing, [721](#)
- ComputeMOSAICDimensions
 - gdcm::SplitMosaicFilter, [668](#)
- ComputeNumberOfSurfaces
 - gdcm::SurfaceWriter, [700](#)
- ComputeOffsetTable
 - gdcm::JPEGCodec, [466](#)
- ComputePixelAspectRatioFromPixelSpacing
 - gdcm::Spacing, [666](#)
- ComputePixelTypeFromMinMax
 - gdcm::Rescaler, [608](#)
- ComputeSpacingFromImagePositionPatient
 - gdcm::ImageHelper, [422](#)
- ComputeVR
 - gdcm::DataSetHelper, [293](#)
- ComputeZSpacing
 - gdcm::IPPSorter, [449](#)
- ComputedRadiographyImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- ConcatenatePDVBlobs
 - gdcm::network::PresentationDataValue, [573](#)
- Conditional
 - gdcm::Usage, [810](#)
- const
 - gdcm::SOPClassUIDToIOD, [661](#)
- const_iterator
 - gdcm::CodeString, [240](#)
 - gdcm::LO, [475](#)
 - gdcm::String, [680](#)
- const_reference
 - gdcm::CodeString, [240](#)
 - gdcm::LO, [475](#)
 - gdcm::String, [680](#)
- const_reverse_iterator
 - gdcm::CodeString, [240](#)
 - gdcm::LO, [475](#)
- gdcm::String, [681](#)
- ConstCharWrapper
 - gdcm::ConstCharWrapper, [250](#)
- ConstIterator
 - gdcm::CSAHeaderDict, [263](#)
 - gdcm::DataSet, [287](#)
 - gdcm::Dict, [304](#)
 - gdcm::Scanner, [617](#)
 - gdcm::SequenceOfFragments, [634](#)
 - gdcm::SequenceOfItems, [639](#)
- Construct
 - gdcm::BaseRootQuery, [199](#)
- ConstructAbortPDU
 - gdcm::network::PDUFactory, [538](#)
- ConstructCEchoRQ
 - gdcm::network::CompositeMessageFactory, [246](#)
- ConstructCFindRQ
 - gdcm::network::CompositeMessageFactory, [246](#)
- ConstructCMoveRQ
 - gdcm::network::CompositeMessageFactory, [246](#)
- ConstructCStoreRQ
 - gdcm::network::CompositeMessageFactory, [246](#)
- ConstructCStoreRSP
 - gdcm::network::CompositeMessageFactory, [246](#)
- ConstructFromString
 - gdcm::TagPath, [719](#)
- ConstructFromTagList
 - gdcm::TagPath, [719](#)
- ConstructPDU
 - gdcm::network::PDUFactory, [538](#)
- ConstructPDV
 - gdcm::network::BaseCompositeMessage, [194](#)
 - gdcm::network::CEchoRQ, [227](#)
 - gdcm::network::CFindRQ, [231](#)
 - gdcm::network::CMoveRQ, [235](#)
 - gdcm::network::CStoreRQ, [268](#)
 - gdcm::network::CStoreRSP, [269](#)
- ConstructPDVByDataSet
 - gdcm::network::CEchoRSP, [228](#)
 - gdcm::network::CFindCancelRQ, [230](#)
 - gdcm::network::CFindRSP, [232](#)
 - gdcm::network::CMoveCancelRq, [233](#)
 - gdcm::network::CMoveRSP, [236](#)
- ConstructQuery
 - gdcm::CompositeNetworkFunctions, [249](#)
- ConstructReleasePDU
 - gdcm::network::PDUFactory, [538](#)
- ConstructorType
 - gdcm::Dicts, [313](#)
- Convert
 - gdcm::DictConverter, [306](#)
 - gdcm::ImageConverter, [419](#)
- ConvertRGBToPaletteColor
 - gdcm::IconImageGenerator, [394](#)

- ConvertToCXX
 - gdcm::DictConverter, [306](#)
- ConvertToXML
 - gdcm::DictConverter, [306](#)
- Create
 - gdcm::Preamble, [565](#)
- CreateCEchoPDU
 - gdcm::network::PDUFactory, [538](#)
- CreateCFindPDU
 - gdcm::network::PDUFactory, [538](#)
- CreateCMovePDU
 - gdcm::network::PDUFactory, [538](#)
- CreateCStoreRQPDU
 - gdcm::network::PDUFactory, [538](#)
- CreateCStoreRSPPDU
 - gdcm::network::PDUFactory, [538](#)
- CreateDefaultUniqueSeriesIdentifier
 - gdcm::SerieHelper, [644](#)
- CreateUniqueSeriesIdentifier
 - gdcm::SerieHelper, [644](#)
- Cross
 - gdcm::DirectionCosines, [317](#)
- CrossDot
 - gdcm::DirectionCosines, [317](#)
- CryptographicMessageSyntax
 - gdcm::CryptographicMessageSyntax, [253](#)
- Curve
 - gdcm::Curve, [271](#)
 - vtkGDCMImageReader, [838](#)
- Curves
 - gdcm::Pixmap, [552](#)
- cyan
 - gdcm::terminal, [131](#)
- DA
 - gdcm::VR, [826](#)
- DATASET_FORMAT
 - gdcm::CSAHeader, [260](#)
- DES3_CIPHER
 - gdcm::CryptographicMessageSyntax, [253](#)
- DES_CIPHER
 - gdcm::CryptographicMessageSyntax, [253](#)
- DETECTOR
 - gdcm::Spacing, [666](#)
- DICOMApplicationContextName
 - gdcm::UIDs, [744](#)
- DICOMControlledTerminology
 - gdcm::UIDs, [744](#)
- DICOMUIDRegistry
 - gdcm::UIDs, [744](#)
- DICT_DEBUG
 - gdcm::DictConverter, [306](#)
- DICT_DEFAULT
 - gdcm::DictConverter, [306](#)
- DICT_XML
 - gdcm::DictConverter, [306](#)
- DS
 - gdcm::VR, [826](#)
- DT
 - gdcm::VR, [827](#)
- DAComp
 - gdcm, [117](#)
- DICOMDIR
 - gdcm::DICOMDIR, [300](#)
- DICOMDIRGenerator
 - gdcm::DICOMDIRGenerator, [302](#)
- DTComp
 - gdcm, [117](#)
- DataElement
 - gdcm::DataElement, [275](#)
- DataElementSet
 - gdcm::DataSet, [287](#)
- DataElementType
 - gdcm::ModuleEntry, [505](#)
- DataEvent
 - gdcm::DataEvent, [284](#)
- DataField
 - gdcm::CSAElement, [257](#)
- DataPtr
 - gdcm::CSAElement, [255](#)
- DataSetEvent
 - gdcm::DataSetEvent, [293](#)
- DataSetHandled
 - gdcm::network::ULConnectionCallback, [796](#)
- DataSetHandles
 - gdcm::network::ULConnectionCallback, [796](#)
- DataSetMS
 - gdcm::FileMetaInformation, [372](#)
- DataSetTS
 - gdcm::FileMetaInformation, [372](#)
- DataWasPassed
 - vtkImageMapToColors16, [868](#)
- DebugOff
 - gdcm::Trace, [725](#)
- DebugOn
 - gdcm::Trace, [725](#)
- Decode
 - gdcm::AudioCodec, [190](#)
 - gdcm::Base64, [191](#)
 - gdcm::Curve, [271](#)
 - gdcm::Decoder, [295](#)
 - gdcm::DeltaEncodingCodec, [299](#)
 - gdcm::ImageCodec, [415](#)
 - gdcm::JPEG2000Codec, [461](#)
 - gdcm::JPEGCodec, [466](#)
 - gdcm::JPEGLSCCodec, [470](#)
 - gdcm::KAKADUCoDec, [473](#)
 - gdcm::LookupTable, [478](#)

- gdcmm::Overlay, [522](#)
- gdcmm::PDFCodec, [537](#)
- gdcmm::PVRGCodec, [584](#)
- gdcmm::RAWCodec, [599](#)
- gdcmm::RLECodec, [612](#)
- DecodeByStreams
 - gdcmm::Decoder, [295](#)
 - gdcmm::ImageCodec, [415](#)
 - gdcmm::JPEG12Codec, [456](#)
 - gdcmm::JPEG16Codec, [458](#)
 - gdcmm::JPEG2000Codec, [461](#)
 - gdcmm::JPEG8Codec, [463](#)
 - gdcmm::JPEGCodec, [466](#)
 - gdcmm::RAWCodec, [599](#)
 - gdcmm::RLECodec, [612](#)
- DecodeBytes
 - gdcmm::RAWCodec, [599](#)
- DecodeExtent
 - gdcmm::JPEG2000Codec, [461](#)
 - gdcmm::JPEGCodec, [467](#)
 - gdcmm::JPEGLSCodec, [470](#)
 - gdcmm::RLECodec, [612](#)
- Decompress
 - gdcmm::Overlay, [522](#)
- Decrypt
 - gdcmm::CryptographicMessageSyntax, [253](#)
- DeepCopy
 - vtkRTStructSetProperties, [881](#)
- Default
 - gdcmm::FileMetaInformation, [370](#)
- DefinePixelExtent
 - gdcmm::StreamImageReader, [671](#)
 - gdcmm::StreamImageWriter, [676](#)
- DefineProperBufferLength
 - gdcmm::StreamImageReader, [672](#)
 - gdcmm::StreamImageWriter, [676](#)
- DefinedTerms
 - gdcmm::DefinedTerms, [296](#)
- DeflatedExplicitVRLittleEndian
 - gdcmm::TransferSyntax, [729](#)
 - gdcmm::UIDs, [742](#)
- DeformableSpatialRegistrationStorage
 - gdcmm::UIDs, [746](#)
- Defs
 - gdcmm::Defs, [297](#)
- DeleteDirectory
 - gdcmm::System, [704](#)
- DeltaEncodingCodec
 - gdcmm::DeltaEncodingCodec, [299](#)
- Derive
 - gdcmm::FileDerivation, [364](#)
- Description
 - gdcmm::ModuleEntry, [504](#)
- DescriptionField
 - gdcmm::ModuleEntry, [505](#)
- DetachedInterpretationManagementSOPClassRetired
 - gdcmm::UIDs, [744](#)
- DetachedPatientManagementMetaSOPClassRetired
 - gdcmm::UIDs, [744](#)
- DetachedPatientManagementSOPClass
 - gdcmm::MediaStorage, [489](#)
- DetachedPatientManagementSOPClassRetired
 - gdcmm::UIDs, [744](#)
- DetachedResultsManagementMetaSOPClassRetired
 - gdcmm::UIDs, [744](#)
- DetachedResultsManagementSOPClassRetired
 - gdcmm::UIDs, [744](#)
- DetachedStudyManagementMetaSOPClassRetired
 - gdcmm::UIDs, [744](#)
- DetachedStudyManagementSOPClass
 - gdcmm::MediaStorage, [489](#)
- DetachedStudyManagementSOPClassRetired
 - gdcmm::UIDs, [744](#)
- DetachedVisitManagementSOPClass
 - gdcmm::MediaStorage, [489](#)
- DetachedVisitManagementSOPClassRetired
 - gdcmm::UIDs, [744](#)
- DetailSRStorageTrialRetired
 - gdcmm::UIDs, [746](#)
- DetermineEventByPDU
 - gdcmm::network::PDUFactory, [538](#)
- dicomAETitle
 - gdcmm::UIDs, [748](#)
- dicomApplicationCluster
 - gdcmm::UIDs, [748](#)
- dicomAssociationAcceptor
 - gdcmm::UIDs, [748](#)
- dicomAssociationInitiator
 - gdcmm::UIDs, [748](#)
- dicomAuthorizedNodeCertificateReference
 - gdcmm::UIDs, [748](#)
- dicomConfigurationRoot
 - gdcmm::UIDs, [748](#)
- dicomDescription
 - gdcmm::UIDs, [748](#)
- dicomDevice
 - gdcmm::UIDs, [748](#)
- dicomDeviceName
 - gdcmm::UIDs, [748](#)
- dicomDeviceSerialNumber
 - gdcmm::UIDs, [748](#)
- dicomDevicesRoot
 - gdcmm::UIDs, [748](#)
- dicomHostname
 - gdcmm::UIDs, [748](#)
- dicomInstalled
 - gdcmm::UIDs, [748](#)
- dicomInstitutionAddress

- gdcmm::UIDs, [748](#)
- dicomInstitutionDepartmentName
 - gdcmm::UIDs, [748](#)
- dicomInstitutionName
 - gdcmm::UIDs, [748](#)
- dicomIssuerOfPatientID
 - gdcmm::UIDs, [748](#)
- dicomManufacturer
 - gdcmm::UIDs, [748](#)
- dicomManufacturerModelName
 - gdcmm::UIDs, [748](#)
- dicomNetworkAE
 - gdcmm::UIDs, [748](#)
- dicomNetworkConnection
 - gdcmm::UIDs, [749](#)
- dicomNetworkConnectionReference
 - gdcmm::UIDs, [748](#)
- dicomPort
 - gdcmm::UIDs, [748](#)
- dicomPreferredCalledAETitle
 - gdcmm::UIDs, [748](#)
- dicomPreferredCallingAETitle
 - gdcmm::UIDs, [748](#)
- dicomPrimaryDeviceType
 - gdcmm::UIDs, [748](#)
- dicomRelatedDeviceReference
 - gdcmm::UIDs, [748](#)
- dicomSOPClass
 - gdcmm::UIDs, [748](#)
- dicomSoftwareVersion
 - gdcmm::UIDs, [748](#)
- dicomStationName
 - gdcmm::UIDs, [748](#)
- dicomSupportedCharacterSet
 - gdcmm::UIDs, [748](#)
- dicomTLSCyphersuite
 - gdcmm::UIDs, [748](#)
- dicomThisNodeCertificateReference
 - gdcmm::UIDs, [748](#)
- dicomTransferCapability
 - gdcmm::UIDs, [749](#)
- dicomTransferRole
 - gdcmm::UIDs, [748](#)
- dicomTransferSyntax
 - gdcmm::UIDs, [748](#)
- dicomUniqueAETitle
 - gdcmm::UIDs, [749](#)
- dicomUniqueAETitlesRegistryRoot
 - gdcmm::UIDs, [748](#)
- dicomVendorData
 - gdcmm::UIDs, [748](#)
- Dict
 - gdcmm::Dict, [304](#)
- DictConverter
 - gdcmm::DictConverter, [306](#)
- DictEntry
 - gdcmm::DictEntry, [308](#)
- DictPrinter
 - gdcmm::DictPrinter, [311](#)
- Dicts
 - gdcmm::CSAHeaderDict, [264](#)
 - gdcmm::Dict, [305](#)
 - gdcmm::Dicts, [313](#)
 - gdcmm::PrivateDict, [578](#)
- difference_type
 - gdcmm::CodeString, [240](#)
 - gdcmm::LO, [475](#)
 - gdcmm::String, [681](#)
- DigitalIntraoralXRayImageStorageForPresentation
 - gdcmm::UIDs, [745](#)
- DigitalIntraoralXRayImageStorageForProcessing
 - gdcmm::MediaStorage, [488](#)
 - gdcmm::UIDs, [745](#)
- DigitalIntraoralXrayImageStorageForPresentation
 - gdcmm::MediaStorage, [488](#)
- DigitalMammographyImageStorageForPresentation
 - gdcmm::MediaStorage, [488](#)
- DigitalMammographyImageStorageForProcessing
 - gdcmm::MediaStorage, [488](#)
- DigitalMammographyXRayImageStorageForPresentation
 - gdcmm::UIDs, [745](#)
- DigitalMammographyXRayImageStorageForProcessing
 - gdcmm::UIDs, [745](#)
- DigitalXRayImageStorageForPresentation
 - gdcmm::MediaStorage, [488](#)
 - gdcmm::UIDs, [745](#)
- DigitalXRayImageStorageForProcessing
 - gdcmm::MediaStorage, [488](#)
 - gdcmm::UIDs, [745](#)
- dim
 - gdcmm::terminal, [131](#)
- Dimensions
 - gdcmm::Bitmap, [212](#)
 - gdcmm::ImageCodec, [417](#)
- DirCosTolerance
 - gdcmm::IPPSorter, [449](#)
- DirectionCosines
 - gdcmm::DirectionCosines, [317](#)
 - vtkGDCMImageReader, [838](#)
- Directory
 - gdcmm::Directory, [319](#)
- DoByteSwap
 - gdcmm::ImageCodec, [416](#)
- DolconImage
 - gdcmm::PixmapWriter, [560](#)
- DoInvertMonochrome
 - gdcmm::ImageCodec, [416](#)
- DoOverlayCleanup

- gdcmm::ImageCodec, 416
- DoPaddedCompositePixelCode
 - gdcmm::ImageCodec, 416
- DoPlanarConfiguration
 - gdcmm::ImageCodec, 416
- DoSimpleCopy
 - gdcmm::ImageCodec, 416
- DoYBR
 - gdcmm::ImageCodec, 416
- Dot
 - gdcmm::DirectionCosines, 317
- DropDuplicatePositions
 - gdcmm::IPPSorter, 449
- Dumper
 - gdcmm::Dumper, 324
- DuplicateAttributeError
 - gdcmm::Parser, 528
- eAABORTPDURceivedOpen
 - gdcmm::network, 128
- eAABORTRequest
 - gdcmm::network, 128
- eAASSOCIATE_RQPDURceived
 - gdcmm::network, 128
- eAASSOCIATERequestLocalUser
 - gdcmm::network, 128
- eAASSOCIATEresponseAccept
 - gdcmm::network, 128
- eAASSOCIATEresponseReject
 - gdcmm::network, 128
- eARELEASE_RPPDURceived
 - gdcmm::network, 128
- eARELEASE_RQPDURceivedOpen
 - gdcmm::network, 128
- eARELEASERequest
 - gdcmm::network, 128
- eARELEASEResponse
 - gdcmm::network, 128
- eARTIMTimerExpired
 - gdcmm::network, 129
- eASSOCIATE_ACPDURceived
 - gdcmm::network, 128
- eASSOCIATE_RJPDURceived
 - gdcmm::network, 128
- eArabic
 - gdcmm, 118
- eCyrillic
 - gdcmm, 118
- EDGE
 - gdcmm::MeshPrimitive, 498
- eEventDoesNotExist
 - gdcmm::network, 129
- eFind
 - gdcmm, 119
- eGB18030
 - gdcmm, 119
- eGreek
 - gdcmm, 118
- eHebrew
 - gdcmm, 118
- eImage
 - gdcmm, 119
- eJapanese
 - gdcmm, 119
- eJapaneseKanjiMultibyte
 - gdcmm, 119
- eJapaneseSupplementaryKanjiMultibyte
 - gdcmm, 119
- eKoreanHangulHanjaMultibyte
 - gdcmm, 119
- eLatin1
 - gdcmm, 118
- eLatin2
 - gdcmm, 118
- eLatin3
 - gdcmm, 118
- eLatin4
 - gdcmm, 118
- eLatin5
 - gdcmm, 119
- eMove
 - gdcmm, 119
- ePDATATFPDU
 - gdcmm::network, 128
- ePDATArequest
 - gdcmm::network, 128
- ePatient
 - gdcmm, 119
- ePatientRootType
 - gdcmm, 119
- eSeries
 - gdcmm, 119
- eSta10ReleaseCollisionAc
 - gdcmm::network, 129
- eSta11ReleaseCollisionRq
 - gdcmm::network, 129
- eSta12ReleaseCollisionAcLocal
 - gdcmm::network, 129
- eSta13AwaitingClose
 - gdcmm::network, 129
- eSta1Idle
 - gdcmm::network, 129
- eSta2Open
 - gdcmm::network, 129
- eSta3WaitLocalAssoc
 - gdcmm::network, 129
- eSta4LocalAssocDone
 - gdcmm::network, 129

- eSta5WaitRemoteAssoc
 - gdcm::network, [129](#)
- eSta6TransferReady
 - gdcm::network, [129](#)
- eSta7WaitRelease
 - gdcm::network, [129](#)
- eSta8WaitLocalRelease
 - gdcm::network, [129](#)
- eSta9ReleaseCollisionRqLocal
 - gdcm::network, [129](#)
- eStaDoesNotExist
 - gdcm::network, [129](#)
- eStudy
 - gdcm, [119](#)
- eStudyRootType
 - gdcm, [119](#)
- eThai
 - gdcm, [119](#)
- eTransportConnConfirmLocal
 - gdcm::network, [128](#)
- eTransportConnIndicLocal
 - gdcm::network, [128](#)
- eTransportConnectionClosed
 - gdcm::network, [128](#)
- eUTF8
 - gdcm, [119](#)
- eUnrecognizedPDURceived
 - gdcm::network, [129](#)
- ECharSet
 - gdcm, [118](#)
- EEventID
 - gdcm::network, [128](#)
- EQueryLevel
 - gdcm, [119](#)
- EQueryType
 - gdcm, [119](#)
- ERootType
 - gdcm, [119](#)
- EStateID
 - gdcm::network, [129](#)
- elem
 - gdcm::SerieHelper::Rule, [614](#)
- Element
 - gdcm::Element< TVR, VM::VM1_n >, [331](#)
- Empty
 - gdcm::Anonymizer, [151](#)
 - gdcm::BoxRegion, [217](#)
 - gdcm::DataElement, [276](#)
 - gdcm::FileAnonymizer, [362](#)
 - gdcm::Region, [606](#)
- EncapsulatedCDASStorage
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [747](#)
- EncapsulatedPDFStorage
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [747](#)
- EncapsulatedDocument
 - gdcm::EncapsulatedDocument, [343](#)
- Encode
 - gdcm::Base64, [192](#)
- EncodeBytes
 - gdcm::System, [704](#)
- Encrypt
 - gdcm::CryptographicMessageSyntax, [253](#)
- End
 - gdcm::CSAHeaderDict, [263](#)
 - gdcm::DataSet, [288](#)
 - gdcm::Dict, [304](#)
 - gdcm::IODs, [446](#)
 - gdcm::Scanner, [618](#)
 - gdcm::SequenceOfFragments, [634](#)
 - gdcm::SequenceOfItems, [639](#)
- EndElement
 - gdcm::TableReader, [710](#)
 - gdcm::XMLDictReader, [890](#)
 - gdcm::XMLPrivateDictReader, [892](#)
- EndElementHandler
 - gdcm::Parser, [528](#)
- EndFilter
 - gdcm::SimpleSubjectWatcher, [656](#)
- EndWith
 - gdcm::Filename, [374](#)
- EnhancedCTImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- EnhancedMRImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- EnhancedSR
 - gdcm::MediaStorage, [489](#)
- EnhancedSRStorage
 - gdcm::UIDs, [746](#)
- EnhancedUSVolumeStorage
 - gdcm::MediaStorage, [490](#)
 - gdcm::UIDs, [749](#)
- EnhancedXAImageStorage
 - gdcm::MediaStorage, [490](#)
 - gdcm::UIDs, [746](#)
- EnhancedXRImageStorage
 - gdcm::UIDs, [746](#)
- EnumeratedValues
 - gdcm::EnumeratedValues, [348](#)
- ErrorOff
 - gdcm::Trace, [725](#)
- ErrorOn
 - gdcm::Trace, [725](#)
- ErrorType
 - gdcm::Parser, [528](#)

- EstablishConnection
 - gdcm::network::ULConnectionManager, [800](#)
- EstablishConnectionMove
 - gdcm::network::ULConnectionManager, [800](#)
- Event
 - gdcm::Event, [349](#)
- Exception
 - gdcm::Exception, [351](#)
- Execute
 - gdcm::Command, [243](#)
 - gdcm::MemberCommand, [495](#)
 - gdcm::SimpleMemberCommand, [654](#)
- ExecuteData
 - vtkGDCMImageReader, [835](#)
 - vtkGDCMThreadedImageReader, [855](#)
- ExecuteInformation
 - vtkGDCMImageReader, [835](#)
 - vtkGDCMThreadedImageReader, [855](#)
- ExecuteQuery
 - gdcm::StringFilter, [683](#)
- Explicit
 - gdcm::TransferSyntax, [729](#)
- ExplicitVRBigEndian
 - gdcm::TransferSyntax, [729](#)
 - gdcm::UIDs, [742](#)
- ExplicitVRLittleEndian
 - gdcm::TransferSyntax, [729](#)
 - gdcm::UIDs, [742](#)
- Explore
 - gdcm::Directory, [319](#)
- Extract
 - gdcm::IconImageFilter, [392](#)
- ExtractIconImages
 - gdcm::IconImageFilter, [392](#)
- ExtractVeprolIconImages
 - gdcm::IconImageFilter, [392](#)
- F
 - gdcm::Printer, [577](#)
 - gdcm::Reader, [604](#)
 - gdcm::Validate, [814](#)
- FACET
 - gdcm::MeshPrimitive, [498](#)
- FD
 - gdcm::VR, [827](#)
- FL
 - gdcm::VR, [827](#)
- FLOAT16
 - gdcm::PixelFormat, [546](#)
- FLOAT32
 - gdcm::PixelFormat, [546](#)
- FLOAT64
 - gdcm::PixelFormat, [546](#)
- Fiducials
 - gdcm::Fiducials, [357](#)
- File
 - gdcm::File, [359](#)
- FileAnonymizer
 - gdcm::FileAnonymizer, [362](#)
- FileDerivation
 - gdcm::FileDerivation, [364](#)
- FileExists
 - gdcm::System, [704](#)
- FileExplicitFilter
 - gdcm::FileExplicitFilter, [366](#)
- FilesDirectory
 - gdcm::System, [705](#)
- FilesSymlink
 - gdcm::System, [705](#)
- FileList
 - gdcm, [117](#)
- FileMetaInformation
 - gdcm::FileMetaInformation, [370](#)
- FileName
 - vtkGDCMPolyDataReader, [848](#)
- FileNameOrdering
 - gdcm::SerieHelper, [644](#)
- FileNames
 - vtkGDCMImageReader, [838](#)
- FileSet
 - gdcm::FileSet, [378](#)
- FileSize
 - gdcm::System, [705](#)
- FileTime
 - gdcm::System, [705](#)
- FileType
 - gdcm::FileSet, [378](#)
- FileWithName
 - gdcm::FileWithName, [380](#)
- Filename
 - gdcm::Filename, [374](#)
- filename
 - gdcm::FileWithName, [380](#)
- FilenameGenerator
 - gdcm::FilenameGenerator, [376](#)
- FilenameType
 - gdcm::DICOMDIRGenerator, [302](#)
 - gdcm::Directory, [319](#)
 - gdcm::FilenameGenerator, [376](#)
- Filenames
 - gdcm::Sorter, [665](#)
- FilenamesType
 - gdcm::DICOMDIRGenerator, [302](#)
 - gdcm::Directory, [319](#)
 - gdcm::FilenameGenerator, [376](#)
- FileType
 - gdcm::FileSet, [378](#)
- Fill

- gdcm::ByteValue, 223
- FillFromDataSet
 - gdcm::FileMetaInformation, 370
- FillMedicalImageInformation
 - vtkGDCMImageReader, 835
 - vtkGDCMPolyDataReader, 847
- FindCSAElementByName
 - gdcm::CSAHeader, 260
- FindContext
 - gdcm::network::ULConnection, 794
- FindDataElement
 - gdcm::DataSet, 288
 - gdcm::Item, 452
 - gdcm::SequenceOfItems, 640
- FindDictEntry
 - gdcm::PrivateDict, 578
- FindMacroEntry
 - gdcm::Macro, 482
- FindModuleEntryInMacros
 - gdcm::Module, 502
- FindNextDataElement
 - gdcm::DataSet, 288
- FindPDBelementByName
 - gdcm::PDBHeader, 535
- FindPatientRootQuery
 - gdcm::FindPatientRootQuery, 381
- FindStudyRootQuery
 - gdcm::FindStudyRootQuery, 384
- FirstRender
 - vtkImageColorViewer, 865
- ForceRescale
 - vtkGDCMImageReader, 838
- FormatDateTime
 - gdcm::System, 705
- Fragment
 - gdcm::Fragment, 386
- FragmentVector
 - gdcm::SequenceOfFragments, 634
- FromString
 - gdcm::StringFilter, 683
- FujiPrivateCRImageStorage
 - gdcm::MediaStorage, 490
- GDCM_DIFFERENT
 - gdcm, 118
- GDCM_EQUAL
 - gdcm, 118
- GDCM_GREATER
 - gdcm, 118
- GDCM_GREATEROREQUAL
 - gdcm, 118
- GDCM_LESS
 - gdcm, 118
- GDCM_LESSOREQUAL
 - gdcm, 118
- GEMS
 - gdcm::Dicts, 313
- GEPrivate3DModelStorage
 - gdcm::MediaStorage, 489
- GRAY
 - gdcm::LookupTable, 478
- GREEN
 - gdcm::LookupTable, 478
- GDCM_DO_JOIN
 - gdcmStaticAssert.h, 1096
- GDCM_DO_JOIN2
 - gdcmStaticAssert.h, 1096
- GDCM_EXPORT
 - gdcmWin32.h, 1152
- GDCM_FUNCTION
 - gdcmTrace.h, 1118
- GDCM_JOIN
 - gdcmStaticAssert.h, 1096
- GDCM_LEGACY
 - gdcmLegacyMacro.h, 1012
- GDCM_LEGACY_BODY
 - gdcmLegacyMacro.h, 1012
- GDCM_STATIC_ASSERT
 - gdcm::Attribute, 165
 - gdcmStaticAssert.h, 1096
- GDCMMACROENTRY_H
 - gdcmMacroEntry.h, 1018
- gdcm, 103
 - AEComp, 117
 - ASComp, 117
 - backslash, 119
 - CSComp, 117
 - CompOperators, 118
 - DAComp, 117
 - DTComp, 117
 - eArabic, 118
 - eCyrillic, 118
 - eFind, 119
 - eGB18030, 119
 - eGreek, 118
 - eHebrew, 118
 - eImage, 119
 - eJapanese, 119
 - eJapaneseKanjiMultibyte, 119
 - eJapaneseSupplementaryKanjiMultibyte, 119
 - eKoreanHangulHanjaMultibyte, 119
 - eLatin1, 118
 - eLatin2, 118
 - eLatin3, 118
 - eLatin4, 118
 - eLatin5, 119
 - eMove, 119
 - ePatient, 119

- ePatientRootType, [119](#)
- eSeries, [119](#)
- eStudy, [119](#)
- eStudyRootType, [119](#)
- eThai, [119](#)
- eUTF8, [119](#)
- ECharSet, [118](#)
- EQueryLevel, [119](#)
- EQueryType, [119](#)
- ERootType, [119](#)
- FileList, [117](#)
- GDCM_DIFFERENT, [118](#)
- GDCM_EQUAL, [118](#)
- GDCM_GREATER, [118](#)
- GDCM_GREATEROREQUAL, [118](#)
- GDCM_LESS, [118](#)
- GDCM_LESOREQUAL, [118](#)
- GetVRFromTag, [119](#)
- GlobalInstance, [124](#)
- IconImage, [117](#)
- LD_ALL, [119](#)
- LD_NOSEQ, [119](#)
- LD_NOSHADOW, [119](#)
- LD_NOSHADOWSEQ, [119](#)
- LOComp, [118](#)
- LTComp, [118](#)
- LodModeType, [119](#)
- MacroEntry, [118](#)
- NestedMacroEntries, [118](#)
- operator<<, [120–123](#)
- operator>>, [124](#)
- operator==, [123](#)
- PNComp, [118](#)
- SHComp, [118](#)
- STComp, [118](#)
- TMComp, [118](#)
- TYPETOENCODING, [124](#)
- to_string, [124](#)
- UIComp, [118](#)
- UTComp, [118](#)
- VRBINARY, [124](#)
- gdcm2pnm.man, [893](#)
- gdcm2vtk.man, [893](#)
- gdcm::Attribute
 - VMType, [165](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM1 >
 - VMType, [172](#)
- gdcm::CSAHeader
 - DATASET_FORMAT, [260](#)
 - INTERFILE, [260](#)
 - NOMAGIC, [260](#)
 - SV10, [260](#)
 - UNKNOWN, [260](#)
 - ZEROED_OUT, [260](#)
- gdcm::CryptographicMessageSyntax
 - AES128_CIPHER, [253](#)
 - AES192_CIPHER, [253](#)
 - AES256_CIPHER, [253](#)
 - DES3_CIPHER, [253](#)
 - DES_CIPHER, [253](#)
- gdcm::DictConverter
 - DICT_DEBUG, [306](#)
 - DICT_DEFAULT, [306](#)
 - DICT_XML, [306](#)
- gdcm::Dicts
 - GEMS, [313](#)
 - PHILIPS, [313](#)
 - SIEMENS, [313](#)
- gdcm::LookupTable
 - BLUE, [478](#)
 - GRAY, [478](#)
 - GREEN, [478](#)
 - RED, [478](#)
 - UNKNOWN, [478](#)
- gdcm::MediaStorage
 - AmbulatoryECGWaveformStorage, [489](#)
 - Audio, [490](#)
 - BasicTextSR, [489](#)
 - BasicVoiceAudioWaveformStorage, [489](#)
 - BreastTomosynthesisImageStorage, [490](#)
 - CSANonImageStorage, [489](#)
 - CTImageStorage, [488](#)
 - CardiacElectrophysiologyWaveformStorage, [489](#)
 - ComprehensiveSR, [489](#)
 - ComputedRadiographyImageStorage, [488](#)
 - DetachedPatientManagementSOPClass, [489](#)
 - DetachedStudyManagementSOPClass, [489](#)
 - DetachedVisitManagementSOPClass, [489](#)
 - DigitalIntraoralXRayImageStorageForProcessing, [488](#)
 - DigitalIntraoralXrayImageStorageForPresentation, [488](#)
 - DigitalMammographyImageStorageForPresentation, [488](#)
 - DigitalMammographyImageStorageForProcessing, [488](#)
 - DigitalXRayImageStorageForPresentation, [488](#)
 - DigitalXRayImageStorageForProcessing, [488](#)
 - EncapsulatedCDASStorage, [489](#)
 - EncapsulatedPDFStorage, [489](#)
 - EnhancedCTImageStorage, [488](#)
 - EnhancedMRIImageStorage, [488](#)
 - EnhancedSR, [489](#)
 - EnhancedUSVolumeStorage, [490](#)
 - EnhancedXAImageStorage, [490](#)
 - FujiPrivateCRLImageStorage, [490](#)
 - GEPrivate3DModelStorage, [489](#)
 - GeneralECGWaveformStorage, [489](#)

- GeneralElectricMagneticResonanceImageStorage, 489
- GrayscaleSoftcopyPresentationStateStorageSOP-Class, 489
- HangingProtocolStorage, 490
- HardcopyGrayscaleImageStorage, 489
- HemodynamicWaveformStorage, 489
- KeyObjectSelectionDocument, 489
- LeadECGWaveformStorage, 489
- MRImageStorage, 488
- MRSpectroscopyStorage, 488
- MS_END, 490
- MammographyCADSR, 489
- MediaStorageDirectoryStorage, 488
- ModalityPerformedProcedureStepSOPClass, 490
- MultiframeGrayscaleByteSecondaryCaptureImageStorage, 488
- MultiframeGrayscaleWordSecondaryCaptureImageStorage, 488
- MultiframeSingleBitSecondaryCaptureImageStorage, 488
- MultiframeTrueColorSecondaryCaptureImageStorage, 489
- NoObject, 490
- NuclearMedicineImageStorage, 489
- NuclearMedicineImageStorageRetired, 488
- ObjectEnd, 490
- OphthalmicPhotography8BitImageStorage, 490
- OphthalmicTomographyImageStorage, 490
- PDF, 490
- PETImageStorage, 489
- Philips3D, 489
- PhilipsPrivateMRSyntheticImageStorage, 490
- RTDoseStorage, 489
- RTImageStorage, 489
- RTIonBeamsTreatmentRecordStorage, 490
- RTIonPlanStorage, 490
- RTPlanStorage, 489
- RTStructureSetStorage, 489
- RTTreatmentSummaryRecordStorage, 490
- RawDataStorage, 489
- SecondaryCaptureImageStorage, 488
- Segmentation, 490
- SegmentationStorage, 490
- SpacialFiducialsStorage, 489
- SpacialRegistrationStorage, 489
- StandaloneCurveStorage, 489
- StandaloneModalityLUTStorage, 489
- StandaloneOverlayStorage, 489
- StandaloneVOILUTStorage, 489
- StudyComponentManagementSOPClass, 489
- SurfaceSegmentationStorage, 490
- ToshibaPrivateDataStorage, 489
- URI, 490
- UltrasoundImageStorage, 488
- UltrasoundImageStorageRetired, 488
- UltrasoundMultiFrameImageStorage, 488
- UltrasoundMultiFrameImageStorageRetired, 488
- VLEndoscopicImageStorage, 490
- VLPhotographicImageStorage, 490
- VLWholeSlideMicroscopyImageStorage, 490
- Video, 490
- VideoEndoscopicImageStorage, 489
- Waveform, 490
- XRay3DAngiographicImageStorage, 490
- XRayAngiographicBiPlaneImageStorageRetired, 489
- XRayAngiographicImageStorage, 489
- XRayRadiationDoseSR, 490
- XRayRadiofluoroscopicImageStorage, 489
- gdcmmesh::MeshPrimitive
 - EDGE, 498
 - FACET, 498
 - LINE, 498
 - MPTYPE_END, 498
 - TRIANGLE, 498
 - TRIANGLE_FAN, 498
 - TRIANGLE_STRIP, 498
 - VERTEX, 498
- gdcmmath::Orientation
 - AXIAL, 518
 - CORONAL, 518
 - OBLIQUE, 518
 - SAGITTAL, 518
 - UNKNOWN, 518
- gdcmmath::Overlay
 - Graphics, 521
 - Invalid, 521
 - ROI, 521
- gdcmmath::Parser
 - DuplicateAttributeError, 528
 - JunkAfterDocElementError, 528
 - NoElementsError, 528
 - NoError, 528
 - NoMemoryError, 528
 - SyntaxError, 528
 - TagMismatchError, 528
 - UndefinedEntityError, 528
 - UnexpectedStateError, 528
- gdcmmath::PhotometricInterpretation
 - ARGB, 543
 - CMYK, 543
 - HSV, 543
 - MONOCHROME1, 543
 - MONOCHROME2, 543
 - PALETTE_COLOR, 543
 - PI_END, 543
 - RGB, 543
 - UNKNOWN, 543

- YBR_FULL, [543](#)
- YBR_FULL_422, [543](#)
- YBR_ICT, [543](#)
- YBR_PARTIAL_420, [543](#)
- YBR_PARTIAL_422, [543](#)
- YBR_RCT, [543](#)
- gdcmm::PixelFormat
 - FLOAT16, [546](#)
 - FLOAT32, [546](#)
 - FLOAT64, [546](#)
 - INT12, [546](#)
 - INT16, [546](#)
 - INT32, [546](#)
 - INT8, [546](#)
 - SINGLEBIT, [546](#)
 - UINT12, [546](#)
 - UINT16, [546](#)
 - UINT32, [546](#)
 - UINT8, [546](#)
 - UNKNOWN, [546](#)
- gdcmm::Printer
 - CONDENSED_STYLE, [576](#)
 - VERBOSE_STYLE, [576](#)
 - XML, [576](#)
- gdcmm::STATIC_ASSERTION_FAILURE< true >
 - value, [670](#)
- gdcmm::Segment
 - ALGOType_END, [623](#)
 - AUTOMATIC, [623](#)
 - MANUAL, [623](#)
- gdcmm::Spacing
 - CALIBRATED, [666](#)
 - DETECTOR, [666](#)
 - MAGNIFIED, [666](#)
 - UNKNOWN, [666](#)
- gdcmm::Surface
 - NO, [690](#)
 - POINTS, [690](#)
 - STATES_END, [690](#)
 - SURFACE, [690](#)
 - UNKNOWN, [690](#)
 - VIEWType_END, [690](#)
 - WIREFRAME, [690](#)
 - YES, [690](#)
- gdcmm::SwapCode
 - BadBigEndian, [701](#)
 - BadLittleEndian, [701](#)
 - BigEndian, [701](#)
 - LittleEndian, [701](#)
 - Unknown, [701](#)
- gdcmm::TransferSyntax
 - CT_private_ELE, [729](#)
 - DeflatedExplicitVRLittleEndian, [729](#)
 - Explicit, [729](#)
 - ExplicitVRBigEndian, [729](#)
 - ExplicitVRLittleEndian, [729](#)
 - Implicit, [729](#)
 - ImplicitVRBigEndianACRNEMA, [729](#)
 - ImplicitVRBigEndianPrivateGE, [729](#)
 - ImplicitVRLittleEndian, [729](#)
 - JPEG2000, [729](#)
 - JPEG2000Lossless, [729](#)
 - JPEG2000Part2, [729](#)
 - JPEG2000Part2Lossless, [729](#)
 - JPEGBaselineProcess1, [729](#)
 - JPEGExtendedProcess2_4, [729](#)
 - JPEGExtendedProcess3_5, [729](#)
 - JPEGFullProgressionProcess10_12, [729](#)
 - JPEGLSLossless, [729](#)
 - JPEGLSNearLossless, [729](#)
 - JPEGLosslessProcess14, [729](#)
 - JPEGLosslessProcess14_1, [729](#)
 - JPEGSpectralSelectionProcess6_8, [729](#)
 - JPIPRendered, [729](#)
 - MPEG2MainProfile, [729](#)
 - RLELossless, [729](#)
 - TS_END, [729](#)
 - Unknown, [729](#)
- gdcmm::Type
 - T1, [734](#)
 - T1C, [734](#)
 - T2, [734](#)
 - T2C, [734](#)
 - T3, [734](#)
 - UNKNOWN, [734](#)
- gdcmm::UIDs
 - AmbulatoryECGWaveformStorage, [745](#)
 - AudioSRStorageTrialRetired, [746](#)
 - BasicAnnotationBoxSOPClass, [744](#)
 - BasicColorImageBoxSOPClass, [744](#)
 - BasicColorPrintManagementMetaSOPClass, [744](#)
 - BasicFilmBoxSOPClass, [744](#)
 - BasicFilmSessionSOPClass, [744](#)
 - BasicGrayscaleImageBoxSOPClass, [744](#)
 - BasicGrayscalePrintManagementMetaSOPClass, [744](#)
 - BasicPrintImageOverlayBoxSOPClassRetired, [745](#)
 - BasicStudyContentNotificationSOPClassRetired, [744](#)
 - BasicTextSRStorage, [746](#)
 - BasicVoiceAudioWaveformStorage, [745](#)
 - BlendingSoftcopyPresentationStateStorageSOPClass, [746](#)
 - BreastImagingRelevantPatientInformationQuery, [747](#)
 - BreastTomosynthesisImageStorage, [749](#)
 - CTImageStorage, [745](#)
 - CardiacElectrophysiologyWaveformStorage, [745](#)
 - CardiacRelevantPatientInformationQuery, [748](#)

- ChestCADSRStorage, [747](#)
- ColorSoftcopyPresentationStateStorageSOPClass, [746](#)
- ComprehensiveSRStorage, [746](#)
- ComprehensiveSRStorageTrialRetired, [746](#)
- ComputedRadiographyImageStorage, [745](#)
- DICOMApplicationContextName, [744](#)
- DICOMControlledTerminology, [744](#)
- DICOMUIDRegistry, [744](#)
- DeflatedExplicitVRLittleEndian, [742](#)
- DeformableSpatialRegistrationStorage, [746](#)
- DetachedInterpretationManagementSOPClass-Retired, [744](#)
- DetachedPatientManagementMetaSOPClass-Retired, [744](#)
- DetachedPatientManagementSOPClassRetired, [744](#)
- DetachedResultsManagementMetaSOPClass-Retired, [744](#)
- DetachedResultsManagementSOPClassRetired, [744](#)
- DetachedStudyManagementMetaSOPClassRetired, [744](#)
- DetachedStudyManagementSOPClassRetired, [744](#)
- DetachedVisitManagementSOPClassRetired, [744](#)
- DetailSRStorageTrialRetired, [746](#)
- dicomAETitle, [748](#)
- dicomApplicationCluster, [748](#)
- dicomAssociationAcceptor, [748](#)
- dicomAssociationInitiator, [748](#)
- dicomAuthorizedNodeCertificateReference, [748](#)
- dicomConfigurationRoot, [748](#)
- dicomDescription, [748](#)
- dicomDevice, [748](#)
- dicomDeviceName, [748](#)
- dicomDeviceSerialNumber, [748](#)
- dicomDevicesRoot, [748](#)
- dicomHostname, [748](#)
- dicomInstalled, [748](#)
- dicomInstitutionAddress, [748](#)
- dicomInstitutionDepartmentName, [748](#)
- dicomInstitutionName, [748](#)
- dicomIssuerOfPatientID, [748](#)
- dicomManufacturer, [748](#)
- dicomManufacturerModelName, [748](#)
- dicomNetworkAE, [748](#)
- dicomNetworkConnection, [749](#)
- dicomNetworkConnectionReference, [748](#)
- dicomPort, [748](#)
- dicomPreferredCalledAETitle, [748](#)
- dicomPreferredCallingAETitle, [748](#)
- dicomPrimaryDeviceType, [748](#)
- dicomRelatedDeviceReference, [748](#)
- dicomSOPClass, [748](#)
- dicomSoftwareVersion, [748](#)
- dicomStationName, [748](#)
- dicomSupportedCharacterSet, [748](#)
- dicomTLSCyphersuite, [748](#)
- dicomThisNodeCertificateReference, [748](#)
- dicomTransferCapability, [749](#)
- dicomTransferRole, [748](#)
- dicomTransferSyntax, [748](#)
- dicomUniqueAETitle, [749](#)
- dicomUniqueAETitlesRegistryRoot, [748](#)
- dicomVendorData, [748](#)
- DigitalIntraoralXRayImageStorageForPresentation, [745](#)
- DigitalIntraoralXRayImageStorageForProcessing, [745](#)
- DigitalMammographyXRayImageStorageForPresentation, [745](#)
- DigitalMammographyXRayImageStorageForProcessing, [745](#)
- DigitalXRayImageStorageForPresentation, [745](#)
- DigitalXRayImageStorageForProcessing, [745](#)
- EncapsulatedCDASStorage, [747](#)
- EncapsulatedPDFStorage, [747](#)
- EnhancedCTImageStorage, [745](#)
- EnhancedMRIImageStorage, [745](#)
- EnhancedSRStorage, [746](#)
- EnhancedUSVolumeStorage, [749](#)
- EnhancedXAImageStorage, [746](#)
- EnhancedXRFImageStorage, [746](#)
- ExplicitVRBigEndian, [742](#)
- ExplicitVRLittleEndian, [742](#)
- GeneralECGWaveformStorage, [745](#)
- GeneralPurposePerformedProcedureStepSOP-Class, [747](#)
- GeneralPurposeScheduledProcedureStepSOP-Class, [747](#)
- GeneralPurposeWorklistInformationModelFIND, [747](#)
- GeneralPurposeWorklistManagementMetaSOP-Class, [747](#)
- GeneralRelevantPatientInformationQuery, [747](#)
- GrayscaleSoftcopyPresentationStateStorageSOP-Class, [746](#)
- HangingProtocolInformationModelFIND, [748](#)
- HangingProtocolInformationModelMOVE, [748](#)
- HangingProtocolStorage, [748](#)
- HardcopyColorImageStorageSOPClassRetired, [745](#)
- HardcopyGrayscaleImageStorageSOPClassRetired, [745](#)
- HemodynamicWaveformStorage, [745](#)
- ICBM452T1FrameofReference, [744](#)
- ICBMSingleSubjectMRIFrameofReference, [744](#)
- ImageOverlayBoxSOPClassRetired, [745](#)
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM, [742](#)
- InstanceAvailabilityNotificationSOPClass, [747](#)

- JPEG2000ImageCompression, [743](#)
- JPEG2000ImageCompressionLosslessOnly, [743](#)
- JPEG2000Part2MulticomponentImageCompression, [743](#)
- JPEG2000Part2MulticomponentImageCompression-LosslessOnly, [743](#)
- JPEGBaselineProcess1DefaultTransferSyntaxfor-LossyJPEG8BitImageCompression, [742](#)
- JPEGExtendedHierarchicalProcess1618Retired, [743](#)
- JPEGExtendedHierarchicalProcess1719Retired, [743](#)
- JPEGExtendedProcess24DefaultTransferSyntaxfor-LossyJPEG12BitImageCompressionProcess4only, [742](#)
- JPEGExtendedProcess35Retired, [742](#)
- JPEGFullProgressionHierarchicalProcess2426-Retired, [743](#)
- JPEGFullProgressionHierarchicalProcess2527-Retired, [743](#)
- JPEGFullProgressionNonHierarchicalProcess1012-Retired, [742](#)
- JPEGFullProgressionNonHierarchicalProcess1113-Retired, [742](#)
- JPEGLSLosslessImageCompression, [743](#)
- JPEGLSLossyNearLosslessImageCompression, [743](#)
- JPEGLosslessHierarchicalProcess28Retired, [743](#)
- JPEGLosslessHierarchicalProcess29Retired, [743](#)
- JPEGLosslessNonHierarchicalFirstOrderPrediction-Process14SelectionValue1DefaultTransfer-SyntaxforLosslessJPEGImageCompression, [743](#)
- JPEGLosslessNonHierarchicalProcess14, [742](#)
- JPEGLosslessNonHierarchicalProcess15Retired, [743](#)
- JPEGSpectralSelectionHierarchicalProcess2022-Retired, [743](#)
- JPEGSpectralSelectionHierarchicalProcess2123-Retired, [743](#)
- JPEGSpectralSelectionNonHierarchicalProcess68-Retired, [742](#)
- JPEGSpectralSelectionNonHierarchicalProcess79-Retired, [742](#)
- JPIPRReferenced, [743](#)
- JPIPRReferencedDeflate, [743](#)
- KeyObjectSelectionDocumentStorage, [747](#)
- MPEG2MainProfileMainLevel, [743](#)
- MRImageStorage, [745](#)
- MRSpectroscopyStorage, [745](#)
- MammographyCADSRStorage, [746](#)
- MediaCreationManagementSOPClassUID, [745](#)
- MediaStorageDirectoryStorage, [743](#)
- ModalityPerformedProcedureStepNotificationSOP-Class, [744](#)
- ModalityPerformedProcedureStepRetrieveSOP-Class, [744](#)
- ModalityPerformedProcedureStepSOPClass, [744](#)
- ModalityWorklistInformationModelFIND, [747](#)
- MultiframeGrayscaleByteSecondaryCaptureImage-Storage, [745](#)
- MultiframeGrayscaleWordSecondaryCaptureImage-Storage, [745](#)
- MultiframeSingleBitSecondaryCaptureImageStorage, [745](#)
- MultiframeTrueColorSecondaryCaptureImage-Storage, [745](#)
- NuclearMedicineImageStorage, [746](#)
- NuclearMedicineImageStorageRetired, [745](#)
- OphthalmicPhotography16BitImageStorage, [746](#)
- OphthalmicPhotography8BitImageStorage, [746](#)
- OphthalmicTomographyImageStorage, [746](#)
- PatientRootQueryRetrieveInformationModelFIND, [747](#)
- PatientRootQueryRetrieveInformationModelGET, [747](#)
- PatientRootQueryRetrieveInformationModelMOVE, [747](#)
- PatientStudyOnlyQueryRetrieveInformationModelFI-NDRetired, [747](#)
- PatientStudyOnlyQueryRetrieveInformationModelG-ETRetired, [747](#)
- PatientStudyOnlyQueryRetrieveInformationModelM-OVERetired, [747](#)
- PositronEmissionTomographyImageStorage, [747](#)
- PresentationLUTSOPClass, [745](#)
- PrintJobSOPClass, [744](#)
- PrintQueueManagementSOPClassRetired, [745](#)
- PrintQueueSOPInstanceRetired, [745](#)
- PrinterConfigurationRetrieveSOPClass, [744](#)
- PrinterConfigurationRetrieveSOPInstance, [744](#)
- PrinterSOPClass, [744](#)
- PrinterSOPInstance, [744](#)
- ProceduralEventLoggingSOPClass, [744](#)
- ProceduralEventLoggingSOPInstance, [744](#)
- ProcedureLogStorage, [746](#)
- ProductCharacteristicsQuerySOPClass, [748](#)
- PseudoColorSoftcopyPresentationStateStorageSO-PClass, [746](#)
- PullPrintRequestSOPClassRetired, [745](#)
- PullStoredPrintManagementMetaSOPClassRetired, [745](#)
- RFC2557MIMEencapsulation, [743](#)
- RLELossless, [743](#)
- RTBeamsDeliveryInstructionStorageSupplement74-FrozenDraft, [747](#)
- RTBeamsTreatmentRecordStorage, [747](#)
- RTBrachyTreatmentRecordStorage, [747](#)
- RTConventionalMachineVerificationSupplement74-FrozenDraft, [747](#)
- RTDoseStorage, [747](#)

- RTImageStorage, [747](#)
- RTIonBeamsTreatmentRecordStorage, [747](#)
- RTIonMachineVerificationSupplement74FrozenDraft, [747](#)
- RTIonPlanStorage, [747](#)
- RTPlanStorage, [747](#)
- RTStructureSetStorage, [747](#)
- RTTreatmentSummaryRecordStorage, [747](#)
- RawDataStorage, [746](#)
- RealWorldValueMappingStorage, [746](#)
- ReferencedColorPrintManagementMetaSOPClass-Retired, [744](#)
- ReferencedGrayscalePrintManagementMetaSOP-ClassRetired, [744](#)
- ReferencedImageBoxSOPClassRetired, [744](#)
- SPM2AVG152PDFFrameofReference, [743](#)
- SPM2AVG152T1FrameofReference, [743](#)
- SPM2AVG152T2FrameofReference, [743](#)
- SPM2AVG305T1FrameofReference, [743](#)
- SPM2BRAINMASKFrameofReference, [743](#)
- SPM2CSFFFrameofReference, [743](#)
- SPM2EPIFrameofReference, [743](#)
- SPM2FILT1FrameofReference, [743](#)
- SPM2GRAYFrameofReference, [743](#)
- SPM2PDFFrameofReference, [743](#)
- SPM2PETFrameofReference, [743](#)
- SPM2SINGLESUBJT1FrameofReference, [743](#)
- SPM2SPECTFrameofReference, [743](#)
- SPM2T1FrameofReference, [743](#)
- SPM2T2FrameofReference, [743](#)
- SPM2TRANSMFrameofReference, [743](#)
- SPM2WHITEFrameofReference, [743](#)
- SecondaryCaptureImageStorage, [745](#)
- SegmentationStorage, [746](#)
- SpatialFiducialsStorage, [746](#)
- SpatialRegistrationStorage, [746](#)
- StandaloneCurveStorageRetired, [745](#)
- StandaloneModalityLUTStorageRetired, [746](#)
- StandaloneOverlayStorageRetired, [745](#)
- StandalonePETCurveStorageRetired, [747](#)
- StandaloneVOILUTStorageRetired, [746](#)
- StereometricRelationshipStorage, [746](#)
- StorageCommitmentPullModelSOPClassRetired, [744](#)
- StorageCommitmentPullModelSOPInstanceRetired, [744](#)
- StorageCommitmentPushModelSOPClass, [744](#)
- StorageCommitmentPushModelSOPInstance, [744](#)
- StorageServiceClass, [744](#)
- StoredPrintStorageSOPClassRetired, [745](#)
- StudyComponentManagementSOPClassRetired, [744](#)
- StudyRootQueryRetrieveInformationModelIFIND, [747](#)
- StudyRootQueryRetrieveInformationModelGET, [747](#)
- StudyRootQueryRetrieveInformationModelMOVE, [747](#)
- SubstanceAdministrationLoggingSOPClass, [744](#)
- SubstanceAdministrationLoggingSOPInstance, [744](#)
- SubstanceApprovalQuerySOPClass, [748](#)
- SurfaceSegmentationStorage, [749](#)
- TalairachBrainAtlasFrameofReference, [743](#)
- TextSRStorageTrialRetired, [746](#)
- uid_1_2_840_10008_15_0_3_1, [754](#)
- uid_1_2_840_10008_15_0_3_10, [754](#)
- uid_1_2_840_10008_15_0_3_11, [754](#)
- uid_1_2_840_10008_15_0_3_12, [755](#)
- uid_1_2_840_10008_15_0_3_13, [755](#)
- uid_1_2_840_10008_15_0_3_14, [755](#)
- uid_1_2_840_10008_15_0_3_15, [755](#)
- uid_1_2_840_10008_15_0_3_16, [755](#)
- uid_1_2_840_10008_15_0_3_17, [755](#)
- uid_1_2_840_10008_15_0_3_18, [755](#)
- uid_1_2_840_10008_15_0_3_19, [755](#)
- uid_1_2_840_10008_15_0_3_2, [754](#)
- uid_1_2_840_10008_15_0_3_20, [755](#)
- uid_1_2_840_10008_15_0_3_21, [755](#)
- uid_1_2_840_10008_15_0_3_22, [755](#)
- uid_1_2_840_10008_15_0_3_23, [755](#)
- uid_1_2_840_10008_15_0_3_24, [755](#)
- uid_1_2_840_10008_15_0_3_25, [755](#)
- uid_1_2_840_10008_15_0_3_26, [755](#)
- uid_1_2_840_10008_15_0_3_27, [755](#)
- uid_1_2_840_10008_15_0_3_28, [755](#)
- uid_1_2_840_10008_15_0_3_29, [755](#)
- uid_1_2_840_10008_15_0_3_3, [754](#)
- uid_1_2_840_10008_15_0_3_30, [755](#)
- uid_1_2_840_10008_15_0_3_31, [755](#)
- uid_1_2_840_10008_15_0_3_4, [754](#)
- uid_1_2_840_10008_15_0_3_5, [754](#)
- uid_1_2_840_10008_15_0_3_6, [754](#)
- uid_1_2_840_10008_15_0_3_7, [754](#)
- uid_1_2_840_10008_15_0_3_8, [754](#)
- uid_1_2_840_10008_15_0_3_9, [754](#)
- uid_1_2_840_10008_15_0_4_1, [755](#)
- uid_1_2_840_10008_15_0_4_2, [755](#)
- uid_1_2_840_10008_15_0_4_3, [755](#)
- uid_1_2_840_10008_15_0_4_4, [755](#)
- uid_1_2_840_10008_15_0_4_5, [755](#)
- uid_1_2_840_10008_15_0_4_6, [755](#)
- uid_1_2_840_10008_15_0_4_7, [755](#)
- uid_1_2_840_10008_15_0_4_8, [755](#)
- uid_1_2_840_10008_1_1, [749](#)
- uid_1_2_840_10008_1_2, [749](#)
- uid_1_2_840_10008_1_20_1, [750](#)
- uid_1_2_840_10008_1_20_1_1, [750](#)
- uid_1_2_840_10008_1_20_2, [750](#)
- uid_1_2_840_10008_1_20_2_1, [750](#)
- uid_1_2_840_10008_1_2_1, [749](#)

uid_1_2_840_10008_1_2_1_99, 749
 uid_1_2_840_10008_1_2_2, 749
 uid_1_2_840_10008_1_2_4_100, 750
 uid_1_2_840_10008_1_2_4_50, 749
 uid_1_2_840_10008_1_2_4_51, 749
 uid_1_2_840_10008_1_2_4_52, 749
 uid_1_2_840_10008_1_2_4_53, 749
 uid_1_2_840_10008_1_2_4_54, 749
 uid_1_2_840_10008_1_2_4_55, 749
 uid_1_2_840_10008_1_2_4_56, 749
 uid_1_2_840_10008_1_2_4_57, 749
 uid_1_2_840_10008_1_2_4_58, 749
 uid_1_2_840_10008_1_2_4_59, 749
 uid_1_2_840_10008_1_2_4_60, 749
 uid_1_2_840_10008_1_2_4_61, 749
 uid_1_2_840_10008_1_2_4_62, 749
 uid_1_2_840_10008_1_2_4_63, 749
 uid_1_2_840_10008_1_2_4_64, 749
 uid_1_2_840_10008_1_2_4_65, 749
 uid_1_2_840_10008_1_2_4_66, 749
 uid_1_2_840_10008_1_2_4_70, 749
 uid_1_2_840_10008_1_2_4_80, 749
 uid_1_2_840_10008_1_2_4_81, 749
 uid_1_2_840_10008_1_2_4_90, 749
 uid_1_2_840_10008_1_2_4_91, 749
 uid_1_2_840_10008_1_2_4_92, 749
 uid_1_2_840_10008_1_2_4_93, 749
 uid_1_2_840_10008_1_2_4_94, 749
 uid_1_2_840_10008_1_2_4_95, 750
 uid_1_2_840_10008_1_2_5, 750
 uid_1_2_840_10008_1_2_6_1, 750
 uid_1_2_840_10008_1_2_6_2, 750
 uid_1_2_840_10008_1_3_10, 750
 uid_1_2_840_10008_1_40, 750
 uid_1_2_840_10008_1_40_1, 750
 uid_1_2_840_10008_1_42, 750
 uid_1_2_840_10008_1_42_1, 750
 uid_1_2_840_10008_1_4_1_1, 750
 uid_1_2_840_10008_1_4_1_10, 750
 uid_1_2_840_10008_1_4_1_11, 750
 uid_1_2_840_10008_1_4_1_12, 750
 uid_1_2_840_10008_1_4_1_13, 750
 uid_1_2_840_10008_1_4_1_14, 750
 uid_1_2_840_10008_1_4_1_15, 750
 uid_1_2_840_10008_1_4_1_16, 750
 uid_1_2_840_10008_1_4_1_17, 750
 uid_1_2_840_10008_1_4_1_18, 750
 uid_1_2_840_10008_1_4_1_2, 750
 uid_1_2_840_10008_1_4_1_3, 750
 uid_1_2_840_10008_1_4_1_4, 750
 uid_1_2_840_10008_1_4_1_5, 750
 uid_1_2_840_10008_1_4_1_6, 750
 uid_1_2_840_10008_1_4_1_7, 750
 uid_1_2_840_10008_1_4_1_8, 750
 uid_1_2_840_10008_1_4_1_9, 750
 uid_1_2_840_10008_1_4_2_1, 750
 uid_1_2_840_10008_1_4_2_2, 750
 uid_1_2_840_10008_1_9, 750
 uid_1_2_840_10008_2_16_4, 750
 uid_1_2_840_10008_2_6_1, 750
 uid_1_2_840_10008_3_1_1_1, 750
 uid_1_2_840_10008_3_1_2_1_1, 750
 uid_1_2_840_10008_3_1_2_1_4, 750
 uid_1_2_840_10008_3_1_2_2_1, 750
 uid_1_2_840_10008_3_1_2_3_1, 750
 uid_1_2_840_10008_3_1_2_3_2, 751
 uid_1_2_840_10008_3_1_2_3_3, 751
 uid_1_2_840_10008_3_1_2_3_4, 751
 uid_1_2_840_10008_3_1_2_3_5, 751
 uid_1_2_840_10008_3_1_2_5_1, 751
 uid_1_2_840_10008_3_1_2_5_4, 751
 uid_1_2_840_10008_3_1_2_5_5, 751
 uid_1_2_840_10008_3_1_2_6_1, 751
 uid_1_2_840_10008_4_2, 751
 uid_1_2_840_10008_5_1_1_1, 751
 uid_1_2_840_10008_5_1_1_14, 751
 uid_1_2_840_10008_5_1_1_15, 751
 uid_1_2_840_10008_5_1_1_16, 751
 uid_1_2_840_10008_5_1_1_16_376, 751
 uid_1_2_840_10008_5_1_1_17, 751
 uid_1_2_840_10008_5_1_1_17_376, 751
 uid_1_2_840_10008_5_1_1_18, 751
 uid_1_2_840_10008_5_1_1_18_1, 751
 uid_1_2_840_10008_5_1_1_2, 751
 uid_1_2_840_10008_5_1_1_22, 751
 uid_1_2_840_10008_5_1_1_23, 751
 uid_1_2_840_10008_5_1_1_24, 751
 uid_1_2_840_10008_5_1_1_24_1, 751
 uid_1_2_840_10008_5_1_1_25, 751
 uid_1_2_840_10008_5_1_1_26, 751
 uid_1_2_840_10008_5_1_1_27, 751
 uid_1_2_840_10008_5_1_1_29, 751
 uid_1_2_840_10008_5_1_1_30, 751
 uid_1_2_840_10008_5_1_1_31, 751
 uid_1_2_840_10008_5_1_1_32, 751
 uid_1_2_840_10008_5_1_1_33, 751
 uid_1_2_840_10008_5_1_1_4, 751
 uid_1_2_840_10008_5_1_1_4_1, 751
 uid_1_2_840_10008_5_1_1_4_2, 751
 uid_1_2_840_10008_5_1_1_9, 751
 uid_1_2_840_10008_5_1_1_9_1, 751
 uid_1_2_840_10008_5_1_4_1_1_1, 751
 uid_1_2_840_10008_5_1_4_1_1_10, 752
 uid_1_2_840_10008_5_1_4_1_1_104_1, 753
 uid_1_2_840_10008_5_1_4_1_1_104_2, 753
 uid_1_2_840_10008_5_1_4_1_1_11, 752
 uid_1_2_840_10008_5_1_4_1_1_11_1, 752
 uid_1_2_840_10008_5_1_4_1_1_11_2, 752

uid_1_2_840_10008_5_1_4_1_1_11_3, [752](#)
uid_1_2_840_10008_5_1_4_1_1_11_4, [752](#)
uid_1_2_840_10008_5_1_4_1_1_128, [753](#)
uid_1_2_840_10008_5_1_4_1_1_129, [753](#)
uid_1_2_840_10008_5_1_4_1_1_12_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_12_1_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_12_2, [752](#)
uid_1_2_840_10008_5_1_4_1_1_12_2_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_12_3, [752](#)
uid_1_2_840_10008_5_1_4_1_1_13_1_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_13_1_2, [752](#)
uid_1_2_840_10008_5_1_4_1_1_13_1_3, [755](#)
uid_1_2_840_10008_5_1_4_1_1_1_1, [751](#)
uid_1_2_840_10008_5_1_4_1_1_1_1_1, [751](#)
uid_1_2_840_10008_5_1_4_1_1_1_2, [751](#)
uid_1_2_840_10008_5_1_4_1_1_1_2_1, [751](#)
uid_1_2_840_10008_5_1_4_1_1_1_3, [751](#)
uid_1_2_840_10008_5_1_4_1_1_1_3_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_2, [752](#)
uid_1_2_840_10008_5_1_4_1_1_20, [752](#)
uid_1_2_840_10008_5_1_4_1_1_2_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_3, [752](#)
uid_1_2_840_10008_5_1_4_1_1_3_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_4, [752](#)
uid_1_2_840_10008_5_1_4_1_1_481_1, [753](#)
uid_1_2_840_10008_5_1_4_1_1_481_2, [753](#)
uid_1_2_840_10008_5_1_4_1_1_481_3, [753](#)
uid_1_2_840_10008_5_1_4_1_1_481_4, [753](#)
uid_1_2_840_10008_5_1_4_1_1_481_5, [753](#)
uid_1_2_840_10008_5_1_4_1_1_481_6, [753](#)
uid_1_2_840_10008_5_1_4_1_1_481_7, [753](#)
uid_1_2_840_10008_5_1_4_1_1_481_8, [753](#)
uid_1_2_840_10008_5_1_4_1_1_481_9, [753](#)
uid_1_2_840_10008_5_1_4_1_1_4_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_4_2, [752](#)
uid_1_2_840_10008_5_1_4_1_1_5, [752](#)
uid_1_2_840_10008_5_1_4_1_1_6, [752](#)
uid_1_2_840_10008_5_1_4_1_1_66, [752](#)
uid_1_2_840_10008_5_1_4_1_1_66_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_66_2, [752](#)
uid_1_2_840_10008_5_1_4_1_1_66_3, [753](#)
uid_1_2_840_10008_5_1_4_1_1_66_4, [753](#)
uid_1_2_840_10008_5_1_4_1_1_66_5, [755](#)
uid_1_2_840_10008_5_1_4_1_1_67, [753](#)
uid_1_2_840_10008_5_1_4_1_1_6_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_6_2, [755](#)
uid_1_2_840_10008_5_1_4_1_1_7, [752](#)
uid_1_2_840_10008_5_1_4_1_1_77_1, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_1, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_1_1, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_2, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_2_1, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_3, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_4, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_4_1, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_5_1, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_5_2, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_5_3, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_5_4, [753](#)
uid_1_2_840_10008_5_1_4_1_1_77_1_6, [755](#)
uid_1_2_840_10008_5_1_4_1_1_77_2, [753](#)
uid_1_2_840_10008_5_1_4_1_1_7_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_7_2, [752](#)
uid_1_2_840_10008_5_1_4_1_1_7_3, [752](#)
uid_1_2_840_10008_5_1_4_1_1_7_4, [752](#)
uid_1_2_840_10008_5_1_4_1_1_8, [752](#)
uid_1_2_840_10008_5_1_4_1_1_88_1, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_11, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_2, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_22, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_3, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_33, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_4, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_40, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_50, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_59, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_65, [753](#)
uid_1_2_840_10008_5_1_4_1_1_88_67, [753](#)
uid_1_2_840_10008_5_1_4_1_1_9, [752](#)
uid_1_2_840_10008_5_1_4_1_1_9_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_9_1_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_9_1_2, [752](#)
uid_1_2_840_10008_5_1_4_1_1_9_1_3, [752](#)
uid_1_2_840_10008_5_1_4_1_1_9_2_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_9_3_1, [752](#)
uid_1_2_840_10008_5_1_4_1_1_9_4_1, [752](#)
uid_1_2_840_10008_5_1_4_1_2_1_1, [753](#)
uid_1_2_840_10008_5_1_4_1_2_1_2, [754](#)
uid_1_2_840_10008_5_1_4_1_2_1_3, [754](#)
uid_1_2_840_10008_5_1_4_1_2_2_1, [754](#)
uid_1_2_840_10008_5_1_4_1_2_2_2, [754](#)
uid_1_2_840_10008_5_1_4_1_2_2_3, [754](#)
uid_1_2_840_10008_5_1_4_1_2_3_1, [754](#)
uid_1_2_840_10008_5_1_4_1_2_3_2, [754](#)
uid_1_2_840_10008_5_1_4_1_2_3_3, [754](#)
uid_1_2_840_10008_5_1_4_31, [754](#)
uid_1_2_840_10008_5_1_4_32, [754](#)
uid_1_2_840_10008_5_1_4_32_1, [754](#)
uid_1_2_840_10008_5_1_4_32_2, [754](#)
uid_1_2_840_10008_5_1_4_32_3, [754](#)
uid_1_2_840_10008_5_1_4_33, [754](#)
uid_1_2_840_10008_5_1_4_34_1, [754](#)
uid_1_2_840_10008_5_1_4_34_2, [754](#)
uid_1_2_840_10008_5_1_4_34_3, [754](#)
uid_1_2_840_10008_5_1_4_34_4, [754](#)
uid_1_2_840_10008_5_1_4_34_4_1, [754](#)
uid_1_2_840_10008_5_1_4_34_4_2, [754](#)
uid_1_2_840_10008_5_1_4_34_4_3, [754](#)

- uid_1_2_840_10008_5_1_4_34_4_4, [754](#)
- uid_1_2_840_10008_5_1_4_34_5, [754](#)
- uid_1_2_840_10008_5_1_4_37_1, [754](#)
- uid_1_2_840_10008_5_1_4_37_2, [754](#)
- uid_1_2_840_10008_5_1_4_37_3, [754](#)
- uid_1_2_840_10008_5_1_4_38_1, [754](#)
- uid_1_2_840_10008_5_1_4_38_2, [754](#)
- uid_1_2_840_10008_5_1_4_38_3, [754](#)
- uid_1_2_840_10008_5_1_4_41, [754](#)
- uid_1_2_840_10008_5_1_4_42, [754](#)
- UltrasoundImageStorage, [745](#)
- UltrasoundImageStorageRetired, [745](#)
- UltrasoundMultiframeImageStorage, [745](#)
- UltrasoundMultiframeImageStorageRetired, [745](#)
- UnifiedProcedureStepEventSOPClass, [747](#)
- UnifiedProcedureStepPullSOPClass, [747](#)
- UnifiedProcedureStepPushSOPClass, [747](#)
- UnifiedProcedureStepWatchSOPClass, [747](#)
- UnifiedWorklistandProcedureStepSOPInstance, [747](#)
- UnifiedWorklistandProcedureStepServiceClass, [747](#)
- VLEndoscopicImageStorage, [746](#)
- VLImageStorageTrialRetired, [746](#)
- VLMicroscopicImageStorage, [746](#)
- VLMultiframeImageStorageTrialRetired, [746](#)
- VLPhotographicImageStorage, [746](#)
- VLSlideCoordinatesMicroscopicImageStorage, [746](#)
- VLWholeSlideMicroscopyImageStorage, [749](#)
- VOILUTBoxSOPClass, [745](#)
- VerificationSOPClass, [742](#)
- VideoEndoscopicImageStorage, [746](#)
- VideoMicroscopicImageStorage, [746](#)
- VideoPhotographicImageStorage, [746](#)
- WaveformStorageTrialRetired, [745](#)
- XMLEncoding, [743](#)
- XRay3DAngiographicImageStorage, [746](#)
- XRay3DCraniofacialImageStorage, [746](#)
- XRayAngiographicBiPlaneImageStorageRetired, [746](#)
- XRayAngiographicImageStorage, [746](#)
- XRayRadiationDoseSRStorage, [747](#)
- XRayRadiofluoroscopicImageStorage, [746](#)
- gdcmm::Usage
 - Conditional, [810](#)
 - Invalid, [810](#)
 - Mandatory, [810](#)
 - UserOption, [810](#)
- gdcmm::VM
 - VM0, [822](#)
 - VM1, [822](#)
 - VM10, [822](#)
 - VM12, [822](#)
 - VM16, [822](#)
 - VM18, [822](#)
 - VM1_2, [823](#)
 - VM1_3, [823](#)
 - VM1_32, [823](#)
 - VM1_4, [823](#)
 - VM1_5, [823](#)
 - VM1_8, [823](#)
 - VM1_99, [823](#)
 - VM1_n, [823](#)
 - VM2, [822](#)
 - VM24, [822](#)
 - VM256, [823](#)
 - VM28, [822](#)
 - VM2_2n, [823](#)
 - VM2_n, [823](#)
 - VM3, [822](#)
 - VM30_30n, [823](#)
 - VM32, [822](#)
 - VM35, [822](#)
 - VM3_3n, [823](#)
 - VM3_4, [823](#)
 - VM3_n, [823](#)
 - VM4, [822](#)
 - VM47_47n, [823](#)
 - VM4_4n, [823](#)
 - VM5, [822](#)
 - VM6, [822](#)
 - VM6_6n, [823](#)
 - VM7_7n, [823](#)
 - VM8, [822](#)
 - VM9, [822](#)
 - VM99, [823](#)
 - VM_END, [823](#)
- gdcmm::VR
 - AE, [826](#)
 - AS, [826](#)
 - AT, [826](#)
 - CS, [826](#)
 - DA, [826](#)
 - DS, [826](#)
 - DT, [827](#)
 - FD, [827](#)
 - FL, [827](#)
 - INVALID, [826](#)
 - IS, [827](#)
 - LO, [827](#)
 - LT, [827](#)
 - OB, [827](#)
 - OB_OW, [827](#)
 - OF, [827](#)
 - OW, [827](#)
 - PN, [827](#)
 - SH, [827](#)
 - SL, [827](#)
 - SQ, [827](#)
 - SS, [827](#)
 - ST, [827](#)

- TM, [827](#)
- UI, [827](#)
- UL, [827](#)
- UN, [827](#)
- US, [827](#)
- US_SS, [827](#)
- US_SS_OW, [827](#)
- UT, [827](#)
- VL16, [827](#)
- VL32, [827](#)
- VR_END, [827](#)
- VR_VM1, [827](#)
- VRALL, [827](#)
- VRASCII, [827](#)
- VRBINARY, [827](#)
- gdcmm::network
 - eAABORTPDUReceivedOpen, [128](#)
 - eAABORTRequest, [128](#)
 - eAASSOCIATE_RQPDUreceived, [128](#)
 - eAASSOCIATERequestLocalUser, [128](#)
 - eAASSOCIATEResponseAccept, [128](#)
 - eAASSOCIATEResponseReject, [128](#)
 - eARELEASE_RPPDUReceived, [128](#)
 - eARELEASE_RQPDUReceivedOpen, [128](#)
 - eARELEASERequest, [128](#)
 - eARELEASEResponse, [128](#)
 - eARTIMTimerExpired, [129](#)
 - eASSOCIATE_ACPDUreceived, [128](#)
 - eASSOCIATE_RJPDUreceived, [128](#)
 - eEventDoesNotExist, [129](#)
 - ePDATATFPDU, [128](#)
 - ePDATArequest, [128](#)
 - eSta10ReleaseCollisionAc, [129](#)
 - eSta11ReleaseCollisionRq, [129](#)
 - eSta12ReleaseCollisionAcLocal, [129](#)
 - eSta13AwaitingClose, [129](#)
 - eSta1Idle, [129](#)
 - eSta2Open, [129](#)
 - eSta3WaitLocalAssoc, [129](#)
 - eSta4LocalAssocDone, [129](#)
 - eSta5WaitRemoteAssoc, [129](#)
 - eSta6TransferReady, [129](#)
 - eSta7WaitRelease, [129](#)
 - eSta8WaitLocalRelease, [129](#)
 - eSta9ReleaseCollisionRqLocal, [129](#)
 - eStaDoesNotExist, [129](#)
 - eTransportConnConfirmLocal, [128](#)
 - eTransportConnIndicLocal, [128](#)
 - eTransportConnectionClosed, [128](#)
 - eUnrecognizedPDUReceived, [129](#)
- gdcmm::network::DIMSE
 - C_CANCEL_RQ, [316](#)
 - C_ECHO_RQ, [315](#)
 - C_ECHO_RSP, [315](#)
 - C_FIND_RQ, [315](#)
 - C_FIND_RSP, [315](#)
 - C_GET_RQ, [315](#)
 - C_GET_RSP, [315](#)
 - C_MOVE_RQ, [315](#)
 - C_MOVE_RSP, [315](#)
 - C_STORE_RQ, [315](#)
 - C_STORE_RSP, [315](#)
 - N_ACTION_RQ, [316](#)
 - N_ACTION_RSP, [316](#)
 - N_CREATE_RQ, [316](#)
 - N_CREATE_RSP, [316](#)
 - N_DELETE_RQ, [316](#)
 - N_DELETE_RSP, [316](#)
 - N_EVENT_REPORT_RQ, [315](#)
 - N_EVENT_REPORT_RSP, [315](#)
 - N_GET_RQ, [315](#)
 - N_GET_RSP, [316](#)
 - N_SET_RQ, [316](#)
 - N_SET_RSP, [316](#)
- gdcmm::terminal
 - black, [131](#)
 - blink, [131](#)
 - blue, [131](#)
 - bright, [131](#)
 - CONSOLE, [131](#)
 - cyan, [131](#)
 - dim, [131](#)
 - green, [131](#)
 - hidden, [131](#)
 - magenta, [131](#)
 - red, [131](#)
 - reset, [131](#)
 - reverse, [131](#)
 - underline, [131](#)
 - VT100, [131](#)
 - white, [131](#)
 - yellow, [131](#)
- gdcmm::ASN1, [161](#)
 - ~ASN1, [162](#)
 - ASN1, [162](#)
 - ParseDump, [162](#)
 - ParseDumpFile, [162](#)
 - TestPBKDF2, [162](#)
- gdcmm::AbortEvent, [143](#)
- gdcmm::AnonymizeEvent, [145](#)
 - ~AnonymizeEvent, [147](#)
 - AnonymizeEvent, [147](#)
 - CheckEvent, [147](#)
 - GetEventName, [147](#)
 - GetTag, [147](#)
 - MakeObject, [147](#)
 - Self, [147](#)
 - SetTag, [147](#)

- Superclass, [147](#)
- gdcmm::Anonymizer, [148](#)
 - ~Anonymizer, [150](#)
 - Anonymizer, [150](#)
 - BALCPPProtect, [150](#)
 - BasicApplicationLevelConfidentialityProfile, [150](#)
 - CanEmptyTag, [151](#)
 - Empty, [151](#)
 - GetBasicApplicationLevelConfidentialityProfile-Attributes, [151](#)
 - GetCryptographicMessageSyntax, [151](#)
 - GetFile, [151](#)
 - New, [151](#)
 - RecurseDataSet, [151](#)
 - Remove, [151](#)
 - RemoveGroupLength, [151](#)
 - RemovePrivateTags, [151](#)
 - RemoveRetired, [152](#)
 - Replace, [152](#)
 - SetCryptographicMessageSyntax, [152](#)
 - SetFile, [152](#)
- gdcmm::AnyEvent, [152](#)
- gdcmm::ApplicationEntity, [155](#)
 - Internal, [156](#)
 - IsValid, [156](#)
 - MaxLength, [156](#)
 - MaxNumberOfComponents, [156](#)
 - Padding, [156](#)
 - Print, [156](#)
 - Separator, [156](#)
 - SetBlob, [156](#)
 - Squeeze, [156](#)
- gdcmm::Attribute
 - ArrayType, [165](#)
 - GDCM_STATIC_ASSERT, [165](#)
 - GetAsDataElement, [165](#)
 - GetDictVM, [166](#)
 - GetDictVR, [166](#)
 - GetNumberOfValues, [166](#)
 - GetTag, [166](#)
 - GetVM, [167](#)
 - GetVR, [167](#)
 - GetValue, [166](#), [167](#)
 - GetValues, [167](#)
 - Internal, [170](#)
 - operator<, [167](#)
 - operator==, [168](#)
 - Print, [168](#)
 - Set, [168](#)
 - SetByteValue, [168](#)
 - SetByteValueNoSwap, [168](#)
 - SetFromDataElement, [169](#)
 - SetFromDataSet, [169](#)
 - SetValue, [169](#)
 - SetValues, [169](#)
- gdcmm::Attribute< Group, Element, TVR, TVM >, [163](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, [170](#)
 - ArrayType, [172](#)
 - GetAsDataElement, [172](#)
 - GetDictVM, [173](#)
 - GetDictVR, [173](#)
 - GetNumberOfValues, [173](#)
 - GetTag, [173](#)
 - GetVM, [173](#)
 - GetVR, [173](#)
 - GetValue, [173](#)
 - GetValues, [173](#)
 - Internal, [175](#)
 - operator<, [173](#)
 - operator==, [173](#)
 - Print, [174](#)
 - Set, [174](#)
 - SetByteValue, [174](#)
 - SetByteValueNoSwap, [174](#)
 - SetFromDataElement, [174](#)
 - SetFromDataSet, [174](#)
 - SetValue, [174](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 >, [175](#)
 - GetVM, [176](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 >, [176](#)
 - GetVM, [177](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >, [177](#)
 - ~Attribute, [179](#)
 - ArrayType, [179](#)
 - Attribute, [179](#)
 - GetAsDataElement, [179](#)
 - GetDictVM, [179](#)
 - GetDictVR, [179](#)
 - GetNumberOfValues, [179](#)
 - GetTag, [180](#)
 - GetVM, [180](#)
 - GetVR, [180](#)
 - GetValue, [180](#)
 - GetValues, [180](#)
 - Print, [180](#)
 - Set, [180](#)
 - SetByteValue, [181](#)
 - SetFromDataElement, [181](#)
 - SetFromDataSet, [181](#)
 - SetNumberOfValues, [181](#)
 - SetValue, [181](#)
 - SetValues, [181](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM2_2n >, [182](#)
 - GetVM, [183](#)

- gdcmm::Attribute< Group, Element, TVR, VM::VM2_n >, 183
 - GetVM, 184
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n >, 185
 - GetVM, 186
- gdcmm::Attribute< Group, Element, TVR, VM::VM3_n >, 186
 - GetVM, 187
- gdcmm::AudioCodec, 188
 - ~AudioCodec, 189
 - AudioCodec, 189
 - CanCode, 189
 - CanDecode, 190
 - Decode, 190
- gdcmm::Base64, 190
 - ~Base64, 190
 - Base64, 190
 - Decode, 191
 - Encode, 192
 - GetDecodeLength, 192
 - GetEncodeLength, 192
- gdcmm::BaseRootQuery, 196
 - ~BaseRootQuery, 198
 - AddQueryDataSet, 199
 - BaseRootQuery, 198
 - Construct, 199
 - GetAbstractSyntaxUID, 199
 - GetQueryDataSet, 199
 - GetQueryLevelFromQueryRoot, 199
 - GetQueryLevelFromString, 199
 - GetQueryLevelString, 199
 - GetTagListByLevel, 199
 - InitializeDataSet, 199
 - mDataSet, 200
 - mHelpDescription, 200
 - mImage, 200
 - mPatient, 200
 - mRootType, 200
 - mSeries, 200
 - mStudy, 200
 - Print, 199
 - QueryFactory, 200
 - SetSearchParameter, 199
 - ValidateQuery, 200
 - WriteHelpFile, 200
 - WriteQuery, 200
- gdcmm::BasicOffsetTable, 203
 - BasicOffsetTable, 204
 - operator<<, 205
 - Read, 205
- gdcmm::Bitmap, 205
 - ~Bitmap, 208
 - AreOverlaysInPixelData, 208
 - Bitmap, 208
 - Clear, 208
 - ComputeLossyFlag, 208
 - Dimensions, 212
 - GetBuffer, 208
 - GetBuffer2, 208
 - GetBufferLength, 208
 - GetColumns, 209
 - GetDataElement, 209
 - GetDimension, 209
 - GetDimensions, 209
 - GetLUT, 209
 - GetNeedByteSwap, 209
 - GetNumberOfDimensions, 209
 - GetPhotometricInterpretation, 209
 - GetPixelFormat, 209, 210
 - GetPlanarConfiguration, 210
 - GetRows, 210
 - GetTransferSyntax, 210
 - ImageChangeTransferSyntax, 212
 - IsEmpty, 210
 - IsLossy, 210
 - IsTransferSyntaxCompatible, 210
 - LUT, 212
 - LUTPtr, 208
 - LossyFlag, 212
 - NeedByteSwap, 212
 - NumberOfDimensions, 212
 - PF, 212
 - PI, 212
 - PixelData, 212
 - PixmapReader, 212
 - PlanarConfiguration, 212
 - Print, 210
 - SetColumns, 210
 - SetDataElement, 210
 - SetDimension, 210
 - SetDimensions, 211
 - SetLUT, 211
 - SetLossyFlag, 211
 - SetNeedByteSwap, 211
 - SetNumberOfDimensions, 211
 - SetPhotometricInterpretation, 211
 - SetPixelFormat, 211
 - SetPlanarConfiguration, 211
 - SetRows, 211
 - SetTransferSyntax, 212
 - TS, 213
 - TryJPEG2000Codec, 212
 - TryJPEG2000Codec2, 212
 - TryJPEGCodec, 212
 - TryJPEGCodec2, 212
 - TryJPEGLSCodec, 212
 - TryKAKADUCodec, 212

- TryPVRGCodec, [212](#)
- TryRAWCodec, [212](#)
- TryRLECodec, [212](#)
- gdcmm::BitmapToBitmapFilter, [213](#)
- ~BitmapToBitmapFilter, [214](#)
- BitmapToBitmapFilter, [214](#)
- GetOutput, [215](#)
- GetOutputAsBitmap, [215](#)
- Input, [215](#)
- Output, [215](#)
- SetInput, [215](#)
- gdcmm::BoxRegion, [215](#)
- ~BoxRegion, [217](#)
- Area, [217](#)
- BoundingBox, [217](#)
- BoxRegion, [217](#)
- Clone, [217](#)
- ComputeBoundingBox, [217](#)
- Empty, [217](#)
- GetXMax, [217](#)
- GetXMin, [217](#)
- GetYMax, [217](#)
- GetYMin, [218](#)
- GetZMax, [218](#)
- GetZMin, [218](#)
- IsValid, [218](#)
- operator=, [218](#)
- Print, [218](#)
- SetDomain, [218](#)
- gdcmm::ByteBuffer, [218](#)
- ByteBuffer, [219](#)
- Get, [219](#)
- GetStart, [219](#)
- ShiftEnd, [219](#)
- UpdatePosition, [219](#)
- gdcmm::ByteSwap
- Swap, [220](#)
- SwapFromSwapCodeIntoSystem, [220](#)
- SwapRange, [220](#)
- SwapRangeFromSwapCodeIntoSystem, [220](#)
- SystemIsBigEndian, [220](#)
- SystemIsLittleEndian, [220](#)
- gdcmm::ByteSwap< T >, [219](#)
- gdcmm::ByteSwapFilter, [220](#)
- ~ByteSwapFilter, [221](#)
- ByteSwap, [221](#)
- ByteSwapFilter, [221](#)
- SetByteSwapTag, [221](#)
- gdcmm::ByteValue, [221](#)
- ~ByteValue, [223](#)
- ByteValue, [223](#)
- Clear, [223](#)
- Fill, [223](#)
- GetBuffer, [223](#)
- GetLength, [223](#)
- GetPointer, [224](#)
- IsEmpty, [224](#)
- IsPrintable, [224](#)
- operator const std::vector< char > &, [224](#)
- operator=, [224](#)
- operator==, [224](#)
- Print, [224](#)
- PrintASCII, [224](#)
- PrintGroupLength, [225](#)
- PrintHex, [225](#)
- Read, [225](#)
- SetLength, [225](#)
- Write, [225](#)
- WriteBuffer, [225](#)
- gdcmm::CP246ExplicitDataElement, [250](#)
- GetLength, [251](#)
- Read, [251](#)
- ReadPreValue, [252](#)
- ReadValue, [252](#)
- ReadWithLength, [252](#)
- gdcmm::CSAElement, [253](#)
- CSAElement, [255](#)
- DataField, [257](#)
- DataPtr, [255](#)
- GetByteValue, [255](#)
- GetKey, [256](#)
- GetName, [256](#)
- GetNoOfItems, [256](#)
- GetSyngoDT, [256](#)
- GetVM, [256](#)
- GetVR, [256](#)
- GetValue, [256](#)
- IsEmpty, [256](#)
- KeyField, [257](#)
- NameField, [258](#)
- NoOfItemsField, [258](#)
- operator<, [257](#)
- operator<<, [257](#)
- operator=, [257](#)
- operator==, [257](#)
- SetByteValue, [257](#)
- SetKey, [257](#)
- SetName, [257](#)
- SetNoOfItems, [257](#)
- SetSyngoDT, [257](#)
- SetVM, [257](#)
- SetVR, [257](#)
- SetValue, [257](#)
- SyngoDTField, [258](#)
- VRField, [258](#)
- ValueMultiplicityField, [258](#)
- gdcmm::CSAHeader, [258](#)
- ~CSAHeader, [260](#)

- CSAHeader, 260
- CSAHeaderType, 260
- FindCSAElementByName, 260
- GetCSADatInfo, 260
- GetCSAEEnd, 261
- GetCSAElementByName, 261
- GetCSAImageHeaderInfoTag, 261
- GetCSASeriesHeaderInfoTag, 261
- GetDataSet, 261
- GetFormat, 261
- GetInterfile, 261
- LoadFromDataElement, 261
- operator<<, 262
- Print, 262
- Read, 262
- Write, 262
- gdcmm::CSAHeaderDict, 262
 - AddCSAHeaderDictEntry, 263
 - Begin, 263
 - CSAHeaderDict, 263
 - ConstIterator, 263
 - Dicts, 264
 - End, 263
 - GetCSAHeaderDictEntry, 263
 - IsEmpty, 263
 - Iterator, 263
 - LoadDefault, 264
 - MapCSAHeaderDictEntry, 263
 - operator<<, 264
- gdcmm::CSAHeaderDictEntry, 264
 - CSAHeaderDictEntry, 265
 - GetDescription, 265
 - GetName, 265
 - GetVM, 265
 - GetVR, 265
 - operator<, 265
 - operator<<, 266
 - SetDescription, 265
 - SetName, 265
 - SetVM, 265
 - SetVR, 265
- gdcmm::CSAHeaderDictException, 266
- gdcmm::CodeString, 239
 - CodeString, 241
 - const_iterator, 240
 - const_reference, 240
 - const_reverse_iterator, 240
 - difference_type, 240
 - GetAsString, 241
 - IsValid, 241
 - iterator, 240
 - operator<<, 241
 - operator==, 241
 - pointer, 240
 - reference, 240
 - reverse_iterator, 240
 - Size, 241
 - size_type, 240
 - TrimInternal, 241
 - value_type, 241
- gdcmm::Codec, 236
- gdcmm::Coder, 237
 - ~Coder, 238
 - CanCode, 238
 - Code, 238
 - InternalCode, 239
- gdcmm::Command, 241
 - ~Command, 243
 - Command, 243
 - Execute, 243
- gdcmm::CommandDataSet, 243
 - ~CommandDataSet, 245
 - CommandDataSet, 245
 - Insert, 245
 - operator<<, 245
 - Read, 245
 - Replace, 245
 - Write, 245
- gdcmm::CompositeNetworkFunctions, 246
 - CEcho, 248
 - CFind, 248
 - CMove, 248
 - CStore, 249
 - ConstructQuery, 249
 - KeyValuePairArrayType, 247
 - KeyValuePairType, 247
- gdcmm::ConstCharWrapper, 249
 - ConstCharWrapper, 250
 - operator const char *, 250
- gdcmm::CryptographicMessageSyntax, 252
 - ~CryptographicMessageSyntax, 253
 - CipherTypes, 253
 - CryptographicMessageSyntax, 253
 - Decrypt, 253
 - Encrypt, 253
 - GetCipherType, 253
 - ParseCertificateFile, 253
 - ParseKeyFile, 253
 - SetCipherType, 253
- gdcmm::Curve, 269
 - ~Curve, 271
 - Curve, 271
 - Decode, 271
 - GetAsPoints, 271
 - GetCurveDataDescriptor, 271
 - GetDataValueRepresentation, 271
 - GetDimensions, 271
 - GetGroup, 271

- GetNumberOfCurves, [271](#)
- GetNumberOfPoints, [271](#)
- GetTypeOfData, [271](#)
- GetTypeOfDataDescription, [271](#)
- IsEmpty, [271](#)
- Print, [271](#)
- SetCoordinateStartValue, [271](#)
- SetCoordinateStepValue, [272](#)
- SetCurve, [272](#)
- SetCurveDataDescriptor, [272](#)
- SetCurveDescription, [272](#)
- SetDataValueRepresentation, [272](#)
- SetDimensions, [272](#)
- SetGroup, [272](#)
- SetNumberOfPoints, [272](#)
- SetTypeOfData, [272](#)
- Update, [272](#)
- gdcmm::DICOMDIR, [300](#)
 - DICOMDIR, [300](#)
- gdcmm::DICOMDIRGenerator, [300](#)
 - ~DICOMDIRGenerator, [302](#)
 - AddImageDirectoryRecord, [302](#)
 - AddPatientDirectoryRecord, [302](#)
 - AddSeriesDirectoryRecord, [302](#)
 - AddStudyDirectoryRecord, [302](#)
 - DICOMDIRGenerator, [302](#)
 - FilenameType, [302](#)
 - FileNamesType, [302](#)
 - Generate, [302](#)
 - GetFile, [302](#)
 - GetScanner, [302](#)
 - SetDescriptor, [302](#)
 - SetFile, [302](#)
 - SetFileNames, [302](#)
 - SetRootDirectory, [302](#)
- gdcmm::DataElement, [272](#)
 - Clear, [276](#)
 - DataElement, [275](#)
 - Empty, [276](#)
 - GetByteValue, [276](#)
 - GetLength, [276](#)
 - GetSequenceOfFragments, [276](#)
 - GetSequenceOfItems, [276, 277](#)
 - GetTag, [277](#)
 - GetVL, [278](#)
 - GetVR, [278](#)
 - GetValue, [277](#)
 - GetValueAsSQ, [277](#)
 - IsEmpty, [278](#)
 - IsUndefinedLength, [278](#)
 - operator<, [278](#)
 - operator<<, [281](#)
 - operator=, [278](#)
 - operator==, [279](#)
 - Read, [279](#)
 - ReadOrSkip, [279](#)
 - ReadPreValue, [279](#)
 - ReadValue, [279](#)
 - ReadWithLength, [279](#)
 - SetByteValue, [279](#)
 - SetTag, [279](#)
 - SetVL, [280](#)
 - SetVLToUndefined, [280](#)
 - SetVR, [280](#)
 - SetValue, [279](#)
 - TagField, [281](#)
 - VRField, [281](#)
 - ValueField, [281](#)
 - ValueLengthField, [281](#)
 - ValuePtr, [275](#)
 - Write, [280](#)
- gdcmm::DataElementException, [281](#)
- gdcmm::DataEvent, [282](#)
 - ~DataEvent, [284](#)
 - CheckEvent, [284](#)
 - DataEvent, [284](#)
 - GetData, [284](#)
 - GetDataLength, [284](#)
 - GetEventName, [284](#)
 - MakeObject, [284](#)
 - Self, [283](#)
 - SetData, [284](#)
 - Superclass, [283](#)
- gdcmm::DataSet, [284](#)
 - Begin, [287](#)
 - CSAHeader, [291](#)
 - Clear, [287](#)
 - ComputeDataElement, [287](#)
 - ComputeGroupLength, [287](#)
 - ConstIterator, [287](#)
 - DataElementSet, [287](#)
 - End, [288](#)
 - FindDataElement, [288](#)
 - FindNextDataElement, [288](#)
 - GetDEEnd, [289](#)
 - GetDES, [289](#)
 - GetDataElement, [288, 289](#)
 - GetLength, [289](#)
 - GetMediaStorage, [289](#)
 - GetPrivateCreator, [289](#)
 - Insert, [289](#)
 - InsertDataElement, [289](#)
 - IsEmpty, [289](#)
 - Iterator, [287](#)
 - operator<<, [291](#)
 - operator(), [290](#)
 - operator=, [290](#)
 - Print, [290](#)

- Read, [290](#)
- ReadNested, [290](#)
- ReadSelectedTags, [290](#)
- ReadSelectedTagsWithLength, [290](#)
- ReadUpToTag, [290](#)
- ReadUpToTagWithLength, [290](#)
- ReadWithLength, [290](#)
- Remove, [290](#)
- Replace, [290](#)
- ReplaceEmpty, [290](#)
- Size, [291](#)
- SizeType, [287](#)
- Write, [291](#)
- gdcmm::DataSetEvent, [291](#)
 - ~DataSetEvent, [293](#)
 - CheckEvent, [293](#)
 - DataSetEvent, [293](#)
 - GetDataSet, [293](#)
 - GetEventName, [293](#)
 - MakeObject, [293](#)
 - Self, [292](#)
 - Superclass, [292](#)
- gdcmm::DataSetHelper, [293](#)
 - ComputeVR, [293](#)
- gdcmm::Decoder, [294](#)
 - ~Decoder, [294](#)
 - CanDecode, [295](#)
 - Decode, [295](#)
 - DecodeByStreams, [295](#)
- gdcmm::DefinedTerms, [295](#)
 - DefinedTerms, [296](#)
- gdcmm::Defs, [296](#)
 - ~Defs, [297](#)
 - Defs, [297](#)
 - GetIODFromFile, [297](#)
 - GetIODNameFromMediaStorage, [297](#)
 - GetIODs, [297](#)
 - GetMacros, [297](#)
 - GetModules, [297](#)
 - GetTypeFromTag, [297](#)
 - Global, [298](#)
 - IsEmpty, [297](#)
 - LoadDefaults, [297](#)
 - LoadFromFile, [297](#)
 - Verify, [297](#), [298](#)
- gdcmm::DeltaEncodingCodec, [298](#)
 - ~DeltaEncodingCodec, [299](#)
 - CanDecode, [299](#)
 - Decode, [299](#)
 - DeltaEncodingCodec, [299](#)
- gdcmm::Dict, [303](#)
 - AddDictEntry, [304](#)
 - Begin, [304](#)
 - ConstIterator, [304](#)
 - Dict, [304](#)
 - Dicts, [305](#)
 - End, [304](#)
 - GetDictEntry, [304](#)
 - GetDictEntryByKeyword, [304](#)
 - GetDictEntryByName, [304](#)
 - GetKeywordFromTag, [304](#)
 - IsEmpty, [305](#)
 - Iterator, [304](#)
 - LoadDefault, [305](#)
 - MapDictEntry, [304](#)
 - operator<<, [305](#)
- gdcmm::DictConverter, [305](#)
 - ~DictConverter, [306](#)
 - AddGroupLength, [306](#)
 - Convert, [306](#)
 - ConvertToCXX, [306](#)
 - ConvertToXML, [306](#)
 - DictConverter, [306](#)
 - GetDictName, [307](#)
 - GetInputFilename, [307](#)
 - GetOutputFilename, [307](#)
 - GetOutputType, [307](#)
 - OutputTypes, [306](#)
 - ReadVM, [307](#)
 - ReadVR, [307](#)
 - Readuint16, [307](#)
 - SetDictName, [307](#)
 - SetInputFileName, [307](#)
 - SetOutputFileName, [307](#)
 - SetOutputType, [307](#)
 - WriteFooter, [307](#)
 - WriteHeader, [307](#)
- gdcmm::DictEntry, [307](#)
 - DictEntry, [308](#)
 - GetKeyword, [308](#)
 - GetName, [308](#)
 - GetRetired, [308](#)
 - GetVM, [309](#)
 - GetVR, [309](#)
 - IsUnique, [309](#)
 - operator<<, [310](#)
 - SetElementXX, [309](#)
 - SetGroupXX, [309](#)
 - SetKeyword, [309](#)
 - SetName, [309](#)
 - SetRetired, [309](#)
 - SetVM, [309](#)
 - SetVR, [309](#)
- gdcmm::DictPrinter, [310](#)
 - ~DictPrinter, [311](#)
 - DictPrinter, [311](#)
 - Print, [312](#)
 - PrintDataElement2, [312](#)

- PrintDataSet2, [312](#)
- gdcm::Dicts, [312](#)
 - ~Dicts, [313](#)
 - ConstructorType, [313](#)
 - Dicts, [313](#)
 - GetCSAHeaderDict, [313](#)
 - GetConstructorString, [313](#)
 - GetDictEntry, [313](#), [314](#)
 - GetPrivateDict, [314](#)
 - GetPublicDict, [314](#)
 - Global, [314](#)
 - IsEmpty, [314](#)
 - LoadDefaults, [314](#)
 - operator<<, [314](#)
- gdcm::DirectionCosines, [316](#)
 - ~DirectionCosines, [317](#)
 - ComputeDistAlongNormal, [317](#)
 - Cross, [317](#)
 - CrossDot, [317](#)
 - DirectionCosines, [317](#)
 - Dot, [317](#)
 - IsValid, [317](#)
 - Normalize, [317](#)
 - operator const double *, [317](#)
 - Print, [317](#)
 - SetFromString, [318](#)
- gdcm::Directory, [318](#)
 - ~Directory, [319](#)
 - Directory, [319](#)
 - Explore, [319](#)
 - FilenameType, [319](#)
 - FileNamesType, [319](#)
 - GetDirectories, [319](#)
 - GetFileNames, [319](#)
 - GetToplevel, [320](#)
 - Load, [320](#)
 - operator<<, [320](#)
 - Print, [320](#)
- gdcm::DirectoryHelper, [320](#)
 - GetCTImageSeriesUIDs, [321](#)
 - GetFileNamesFromSeriesUIDs, [321](#)
 - GetFrameOfReference, [321](#)
 - GetMRImageSeriesUIDs, [321](#)
 - GetRTStructSeriesUIDs, [321](#)
 - GetSOPClassUID, [322](#)
 - GetSeriesUIDsBySOPClassUID, [322](#)
 - GetStringValueFromTag, [322](#)
 - LoadImageFromFiles, [322](#)
 - RetrieveSOPInstanceUIDFromIndex, [322](#)
 - RetrieveSOPInstanceUIDFromZPosition, [322](#)
- gdcm::DummyValueGenerator, [322](#)
 - Generate, [322](#)
- gdcm::Dumper, [323](#)
 - ~Dumper, [324](#)
- Dumper, [324](#)
- gdcm::Element
 - GetAsDataElement, [327](#)
 - GetLength, [327](#)
 - GetVM, [327](#)
 - GetVR, [327](#)
 - GetValue, [327](#)
 - GetValues, [327](#)
 - Internal, [328](#)
 - Print, [327](#)
 - Read, [327](#)
 - Set, [328](#)
 - SetFromDataElement, [328](#)
 - SetNoSwap, [328](#)
 - SetValue, [328](#)
 - Type, [327](#)
 - Write, [328](#)
- gdcm::Element< TVR, TVM >, [325](#)
- gdcm::Element< TVR, VM::VM1_2 >, [328](#)
 - Parent, [329](#)
 - SetLength, [329](#)
- gdcm::Element< TVR, VM::VM1_n >, [329](#)
 - ~Element, [331](#)
 - Element, [331](#)
 - GetAsDataElement, [331](#)
 - GetLength, [331](#)
 - GetVM, [331](#)
 - GetVR, [331](#)
 - GetValue, [331](#)
 - operator=, [331](#)
 - Print, [331](#)
 - Read, [331](#)
 - Set, [331](#)
 - SetArray, [331](#)
 - SetFromDataElement, [332](#)
 - SetLength, [332](#)
 - SetNoSwap, [332](#)
 - SetValue, [332](#)
 - Type, [331](#)
 - Write, [332](#)
 - WriteASCII, [332](#)
- gdcm::Element< TVR, VM::VM2_2n >, [332](#)
 - Parent, [334](#)
 - SetLength, [334](#)
- gdcm::Element< TVR, VM::VM2_n >, [334](#)
 - Parent, [335](#)
 - SetLength, [335](#)
- gdcm::Element< TVR, VM::VM3_3n >, [335](#)
 - Parent, [337](#)
 - SetLength, [337](#)
- gdcm::Element< TVR, VM::VM3_n >, [337](#)
 - Parent, [338](#)
 - SetLength, [338](#)
- gdcm::Element< VR::AS, VM::VM5 >, [338](#)

- GetLength, 339
- Internal, 339
- Print, 339
- gdcmm::Element< VR::OB, VM::VM1 >, 339
- gdcmm::Element< VR::OW, VM::VM1 >, 340
- gdcmm::ElementDisableCombinations< TVR, TVM >, 342
- gdcmm::ElementDisableCombinations< VR::OB, VM::VM1-
_n >, 343
- gdcmm::ElementDisableCombinations< VR::OW, VM::V-
M1_n >, 343
- gdcmm::EncapsulatedDocument, 343
 - EncapsulatedDocument, 343
- gdcmm::EncodingImplementation< T >, 344
- gdcmm::EncodingImplementation< VR::VRASCII >, 344
 - Read, 344
 - ReadComputeLength, 344
 - ReadNoSwap, 345
 - Write, 345
- gdcmm::EncodingImplementation< VR::VRBINARY >, 345
 - Read, 345
 - ReadComputeLength, 345
 - ReadNoSwap, 346
 - Write, 346
- gdcmm::EndEvent, 346
- gdcmm::EnumeratedValues, 347
 - EnumeratedValues, 348
- gdcmm::Event, 348
 - ~Event, 349
 - CheckEvent, 349
 - Event, 349
 - GetEventName, 349
 - MakeObject, 349
 - Print, 349
- gdcmm::Exception, 350
 - ~Exception, 351
 - Exception, 351
 - GetDescription, 351
 - what, 351
- gdcmm::ExitEvent, 351
- gdcmm::ExplicitDataElement, 353
 - GetLength, 354
 - Read, 354
 - ReadPreValue, 354
 - ReadValue, 354
 - ReadWithLength, 354
 - Write, 354
- gdcmm::ExplicitImplicitDataElement, 354
 - GetLength, 356
 - Read, 356
 - ReadPreValue, 356
 - ReadValue, 356
 - ReadWithLength, 356
- gdcmm::Fiducials, 356
 - Fiducials, 357
- gdcmm::File, 357
 - ~File, 359
 - File, 359
 - GetDataSet, 359
 - GetHeader, 359
 - operator<<, 360
 - Read, 360
 - SetDataSet, 360
 - SetHeader, 360
 - Write, 360
- gdcmm::FileAnonymizer, 360
 - ~FileAnonymizer, 362
 - Empty, 362
 - FileAnonymizer, 362
 - Remove, 362
 - Replace, 362
 - SetInputFileName, 363
 - SetOutputFileName, 363
 - Write, 363
- gdcmm::FileDerivation, 363
 - ~FileDerivation, 364
 - AddDerivationDescription, 364
 - AddPurposeOfReferenceCodeSequence, 364
 - AddReference, 364
 - AddSourceImageSequence, 364
 - Derive, 364
 - FileDerivation, 364
 - GetFile, 365
 - SetDerivationCodeSequenceCodeValue, 365
 - SetDerivationDescription, 365
 - SetFile, 365
 - SetPurposeOfReferenceCodeSequenceCodeValue,
365
- gdcmm::FileExplicitFilter, 365
 - ~FileExplicitFilter, 366
 - Change, 366
 - ChangeFMI, 367
 - FileExplicitFilter, 366
 - GetFile, 367
 - ProcessDataSet, 367
 - SetChangePrivateTags, 367
 - SetFile, 367
 - SetRecomputeItemLength, 367
 - SetRecomputeSequenceLength, 367
 - SetUseVRUN, 367
- gdcmm::FileMetaInformation, 367
 - ~FileMetaInformation, 370
 - AppendImplementationClassUID, 370
 - ComputeDataSetMediaStorageSOPClass, 370
 - ComputeDataSetTransferSyntax, 370
 - DataSetMS, 372
 - DataSetTS, 372
 - Default, 370
 - FileMetaInformation, 370

- FillFromDataSet, [370](#)
- GetDataSetTransferSyntax, [370](#)
- GetFileMetaInformationVersion, [370](#)
- GetFullLength, [371](#)
- GetGDCMImplementationClassUID, [371](#)
- GetGDCMImplementationVersionName, [371](#)
- GetGDCMSourceApplicationEntityTitle, [371](#)
- GetImplementationClassUID, [371](#)
- GetImplementationVersionName, [371](#)
- GetMediaStorage, [371](#)
- GetMetaInformationTS, [371](#)
- GetPreamble, [371](#)
- GetSourceApplicationEntityTitle, [371](#)
- Insert, [371](#)
- IsValid, [371](#)
- MetaInformationTS, [372](#)
- operator<<, [372](#)
- Read, [371](#)
- ReadCompat, [371](#)
- ReadCompatInternal, [371](#)
- Replace, [371](#)
- SetDataSetTransferSyntax, [372](#)
- SetImplementationClassUID, [372](#)
- SetImplementationVersionName, [372](#)
- SetPreamble, [372](#)
- SetSourceApplicationEntityTitle, [372](#)
- Write, [372](#)
- gdcmm::FileSet, [377](#)
 - AddFile, [378](#)
 - FileSet, [378](#)
 - FileType, [378](#)
 - FilesType, [378](#)
 - GetFiles, [378](#)
 - operator<<, [378](#)
 - SetFiles, [378](#)
- gdcmm::FileWithName, [378](#)
 - FileWithName, [380](#)
 - filename, [380](#)
- gdcmm::Filename, [373](#)
 - EndWith, [374](#)
 - Filename, [374](#)
 - GetExtension, [374](#)
 - GetFileName, [374](#)
 - GetName, [374](#)
 - GetPath, [374](#)
 - IsEmpty, [374](#)
 - IsIdentical, [374](#)
 - Join, [374](#)
 - operator const char *, [374](#)
 - ToUnixSlashes, [374](#)
 - ToWindowsSlashes, [374](#)
- gdcmm::FilenameGenerator, [375](#)
 - ~FilenameGenerator, [376](#)
 - FilenameGenerator, [376](#)
- FilenameType, [376](#)
- FileNamesType, [376](#)
- Generate, [376](#)
- GetFilename, [376](#)
- GetFileNames, [376](#)
- GetNumberOfFileNames, [376](#)
- GetPattern, [376](#)
- GetPrefix, [377](#)
- SetNumberOfFileNames, [377](#)
- SetPattern, [377](#)
- SetPrefix, [377](#)
- SizeType, [376](#)
- gdcmm::FindPatientRootQuery, [380](#)
 - FindPatientRootQuery, [381](#)
 - GetAbstractSyntaxUID, [381](#)
 - GetTagListByLevel, [381](#)
 - InitializeDataSet, [382](#)
 - QueryFactory, [382](#)
 - ValidateQuery, [382](#)
- gdcmm::FindStudyRootQuery, [382](#)
 - FindStudyRootQuery, [384](#)
 - GetAbstractSyntaxUID, [384](#)
 - GetTagListByLevel, [384](#)
 - InitializeDataSet, [384](#)
 - QueryFactory, [384](#)
 - ValidateQuery, [384](#)
- gdcmm::Fragment, [384](#)
 - Fragment, [386](#)
 - GetLength, [386](#)
 - operator<<, [387](#)
 - Read, [386](#)
 - ReadBacktrack, [386](#)
 - ReadPreValue, [386](#)
 - ReadValue, [386](#)
 - Write, [386](#)
- gdcmm::Global, [387](#)
 - ~Global, [388](#)
 - Append, [388](#)
 - GetDefs, [388](#)
 - GetDicts, [388](#)
 - GetInstance, [388](#)
 - Global, [388](#)
 - LoadResourcesFiles, [389](#)
 - Locate, [389](#)
 - operator<<, [389](#)
 - Prepend, [389](#)
- gdcmm::GroupDict, [389](#)
 - ~GroupDict, [390](#)
 - Add, [390](#)
 - GetAbbreviation, [390](#)
 - GetName, [390](#)
 - GroupDict, [390](#)
 - GroupStringVector, [390](#)
 - Insert, [390](#)

- operator<<, 391
- Size, 391
- gdcmm::IOD, 441
 - AddIODEntry, 442
 - Clear, 442
 - GetIODEntry, 442
 - GetNumberOfIODs, 442
 - GetTypeFromTag, 442
 - IOD, 442
 - MapIODEntry, 442
 - operator<<, 443
 - SizeType, 442
- gdcmm::IODEntry, 443
 - GetIE, 444
 - GetName, 444
 - GetRef, 444
 - GetUsage, 444
 - GetUsageType, 444
 - IODEntry, 444
 - operator<<, 444
 - SetIE, 444
 - SetName, 444
 - SetRef, 444
 - SetUsage, 444
- gdcmm::IODs, 445
 - AddIOD, 446
 - Begin, 446
 - Clear, 446
 - End, 446
 - GetIOD, 446
 - IODMapType, 445
 - IODMapTypeConstIterator, 445
 - IODName, 445
 - IODs, 446
 - operator<<, 446
- gdcmm::IPPSorter, 446
 - ~IPPSorter, 448
 - ComputeZSpacing, 449
 - DirCosTolerance, 449
 - DropDuplicatePositions, 449
 - GetDirectionCosinesTolerance, 448
 - GetZSpacing, 448
 - GetZSpacingTolerance, 448
 - IPPSorter, 448
 - SetComputeZSpacing, 448
 - SetDirectionCosinesTolerance, 449
 - SetDropDuplicatePositions, 449
 - SetZSpacingTolerance, 449
 - Sort, 449
 - ZSpacing, 450
 - ZTolerance, 450
- gdcmm::IconImageFilter, 391
 - ~IconImageFilter, 392
 - Extract, 392
 - ExtractIconImages, 392
 - ExtractVeprolIconImages, 392
 - GetFile, 392
 - GetIconImage, 392
 - GetNumberOfIconImages, 393
 - IconImageFilter, 392
 - SetFile, 393
- gdcmm::IconImageGenerator, 393
 - ~IconImageGenerator, 394
 - AutoPixelMinMax, 394
 - ConvertRGBToPaletteColor, 394
 - Generate, 394
 - GetIconImage, 394
 - GetPixmap, 395
 - IconImageGenerator, 394
 - SetOutputDimensions, 395
 - SetOutsideValuePixel, 395
 - SetPixelMinMax, 395
 - SetPixmap, 395
- gdcmm::Image, 396
 - ~Image, 398
 - GetDirectionCosines, 398
 - GetIntercept, 398
 - GetOrigin, 398
 - GetSlope, 399
 - GetSpacing, 399
 - Image, 398
 - Print, 399
 - SetDirectionCosines, 399
 - SetIntercept, 399
 - SetOrigin, 399
 - SetSlope, 399
 - SetSpacing, 399
- gdcmm::ImageApplyLookupTable, 400
 - ~ImageApplyLookupTable, 402
 - Apply, 402
 - ImageApplyLookupTable, 402
- gdcmm::ImageChangePhotometricInterpretation, 402
 - ~ImageChangePhotometricInterpretation, 405
 - Change, 405
 - ChangeMonochrome, 405
 - GetPhotometricInterpretation, 405
 - ImageChangePhotometricInterpretation, 405
 - RGB2YBR, 405
 - SetPhotometricInterpretation, 405
 - YBR2RGB, 405
- gdcmm::ImageChangePlanarConfiguration, 406
 - ~ImageChangePlanarConfiguration, 408
 - Change, 408
 - GetPlanarConfiguration, 408
 - ImageChangePlanarConfiguration, 408
 - RGBPixelsToRGBPlanes, 408
 - RGBPlanesToRGBPixels, 408
 - SetPlanarConfiguration, 408

- gdcm::ImageChangeTransferSyntax, 409
 - ~ImageChangeTransferSyntax, 411
 - Change, 411
 - GetTransferSyntax, 411
 - ImageChangeTransferSyntax, 411
 - SetCompressIconImage, 411
 - SetForce, 412
 - SetTransferSyntax, 412
 - SetUserCodec, 412
 - TryJPEG2000Codec, 412
 - TryJPEGCodec, 412
 - TryJPEGLSCodec, 412
 - TryRAWCodec, 412
 - TryRLECodec, 412
- gdcm::ImageCodec, 413
 - ~ImageCodec, 415
 - CanCode, 415
 - CanDecode, 415
 - Decode, 415
 - DecodeByStreams, 415
 - Dimensions, 417
 - DoByteSwap, 416
 - DoInvertMonochrome, 416
 - DoOverlayCleanup, 416
 - DoPaddedCompositePixelCode, 416
 - DoPlanarConfiguration, 416
 - DoSimpleCopy, 416
 - DoYBR, 416
 - GetDimensions, 416
 - GetHeaderInfo, 416
 - GetLUT, 416
 - GetLossyFlag, 416
 - GetNeedByteSwap, 416
 - GetNumberOfDimensions, 416
 - GetPhotometricInterpretation, 416
 - GetPixelFormat, 416
 - GetPlanarConfiguration, 416
 - ImageChangePhotometricInterpretation, 417
 - ImageCodec, 415
 - IsLossy, 416
 - IsValid, 416
 - LUT, 418
 - LUTPtr, 415
 - LossyFlag, 418
 - NeedByteSwap, 418
 - NeedOverlayCleanup, 418
 - NumberOfDimensions, 418
 - PF, 418
 - PI, 418
 - PlanarConfiguration, 418
 - RequestPaddedCompositePixelCode, 418
 - RequestPlanarConfiguration, 418
 - SetDimensions, 417
 - SetLUT, 417
 - SetLossyFlag, 417
 - SetNeedByteSwap, 417
 - SetNeedOverlayCleanup, 417
 - SetNumberOfDimensions, 417
 - SetPhotometricInterpretation, 417
 - SetPixelFormat, 417
 - SetPlanarConfiguration, 417
- gdcm::ImageConverter, 418
 - ~ImageConverter, 419
 - Convert, 419
 - GetOutput, 419
 - ImageConverter, 419
 - SetInput, 419
- gdcm::ImageFragmentSplitter, 419
 - ~ImageFragmentSplitter, 421
 - GetFragmentSizeMax, 421
 - ImageFragmentSplitter, 421
 - SetForce, 421
 - SetFragmentSizeMax, 421
 - Split, 421
- gdcm::ImageHelper, 421
 - ComputeSpacingFromImagePositionPatient, 422
 - GetDimensionsValue, 422
 - GetDirectionCosinesFromDataSet, 423
 - GetDirectionCosinesValue, 423
 - GetForcePixelSpacing, 423
 - GetForceRescaleInterceptSlope, 423
 - GetLUT, 423
 - GetOriginValue, 423
 - GetPhotometricInterpretationValue, 423
 - GetPixelFormatValue, 423
 - GetPlanarConfigurationValue, 423
 - GetPointerFromElement, 423
 - GetRescaleInterceptSlopeValue, 423
 - GetSpacingTagFromMediaStorage, 423
 - GetSpacingValue, 424
 - GetZSpacingTagFromMediaStorage, 424
 - SetDimensionsValue, 424
 - SetDirectionCosinesValue, 424
 - SetForcePixelSpacing, 424
 - SetForceRescaleInterceptSlope, 424
 - SetOriginValue, 424
 - SetRescaleInterceptSlopeValue, 424
 - SetSpacingValue, 424
- gdcm::ImageReader, 424
 - ~ImageReader, 427
 - GetImage, 427
 - ImageReader, 427
 - Read, 427
 - ReadACRNEMAIImage, 428
 - ReadImage, 428
- gdcm::ImageRegionReader, 428
 - ~ImageRegionReader, 430
 - ComputeBufferLength, 430

- GetRegion, 430
- ImageRegionReader, 430
- Read, 430
- ReadInformation, 430
- ReadIntoBuffer, 430
- SetRegion, 431
- gdcmm::ImageToImageFilter, 431
 - ~ImageToImageFilter, 433
 - GetInput, 433
 - GetOutput, 433
 - ImageToImageFilter, 433
- gdcmm::ImageWriter, 433
 - ~ImageWriter, 435
 - GetImage, 435
 - ImageWriter, 435
 - Write, 435
- gdcmm::ImplicitDataElement, 438
 - GetLength, 440
 - Read, 440
 - ReadPreValue, 440
 - ReadValue, 440
 - ReadWithLength, 440
 - Write, 440
- gdcmm::InitializeEvent, 440
- gdcmm::Item, 450
 - Clear, 452
 - FindDataElement, 452
 - GetDataElement, 452
 - GetLength, 452
 - GetNestedDataSet, 452, 453
 - InsertDataElement, 453
 - Item, 452
 - operator<<, 453
 - Read, 453
 - SetNestedDataSet, 453
 - Write, 453
- gdcmm::IterationEvent, 453
- gdcmm::JPEG12Codec, 455
 - ~JPEG12Codec, 456
 - DecodeByStreams, 456
 - GetHeaderInfo, 456
 - InternalCode, 456
 - IsStateSuspension, 456
 - JPEG12Codec, 456
- gdcmm::JPEG16Codec, 457
 - ~JPEG16Codec, 458
 - DecodeByStreams, 458
 - GetHeaderInfo, 458
 - InternalCode, 458
 - IsStateSuspension, 458
 - JPEG16Codec, 458
- gdcmm::JPEG2000Codec, 459
 - ~JPEG2000Codec, 460
 - Bitmap, 461
 - CanCode, 460
 - CanDecode, 460
 - Code, 461
 - Decode, 461
 - DecodeByStreams, 461
 - DecodeExtent, 461
 - GetHeaderInfo, 461
 - GetQuality, 461
 - GetRate, 461
 - ImageRegionReader, 461
 - JPEG2000Codec, 460
 - SetNumberOfResolutions, 461
 - SetQuality, 461
 - SetRate, 461
 - SetReversible, 461
 - SetTitleSize, 461
- gdcmm::JPEG8Codec, 462
 - ~JPEG8Codec, 463
 - DecodeByStreams, 463
 - GetHeaderInfo, 463
 - InternalCode, 463
 - IsStateSuspension, 463
 - JPEG8Codec, 463
- gdcmm::JPEGCodec, 464
 - ~JPEGCodec, 466
 - BitSample, 467
 - CanCode, 466
 - CanDecode, 466
 - Code, 466
 - ComputeOffsetTable, 466
 - Decode, 466
 - DecodeByStreams, 466
 - DecodeExtent, 467
 - GetHeaderInfo, 467
 - GetLossless, 467
 - GetQuality, 467
 - ImageRegionReader, 467
 - IsStateSuspension, 467
 - IsValid, 467
 - JPEGCodec, 466
 - Lossless, 468
 - Quality, 468
 - SetBitSample, 467
 - SetLossless, 467
 - SetPixelFormat, 467
 - SetQuality, 467
- gdcmm::JPEGLSCodec, 468
 - ~JPEGLSCodec, 470
 - CanCode, 470
 - CanDecode, 470
 - Code, 470
 - Decode, 470
 - DecodeExtent, 470
 - GetBufferLength, 470

- GetHeaderInfo, [470](#)
- GetLossless, [471](#)
- ImageRegionReader, [471](#)
- JPEGLSCodec, [470](#)
- SetBufferLength, [471](#)
- SetLossless, [471](#)
- SetLossyError, [471](#)
- gdcmm::KAKADUCodec, [471](#)
 - ~KAKADUCodec, [473](#)
 - CanCode, [473](#)
 - CanDecode, [473](#)
 - Code, [473](#)
 - Decode, [473](#)
 - KAKADUCodec, [473](#)
- gdcmm::LO, [473](#)
 - const_iterator, [475](#)
 - const_reference, [475](#)
 - const_reverse_iterator, [475](#)
 - difference_type, [475](#)
 - IsValid, [475](#)
 - iterator, [475](#)
 - LO, [475](#)
 - pointer, [475](#)
 - reference, [475](#)
 - reverse_iterator, [475](#)
 - size_type, [475](#)
 - Superclass, [475](#)
 - value_type, [475](#)
- gdcmm::LookupTable, [476](#)
 - ~LookupTable, [478](#)
 - Allocate, [478](#)
 - BitSample, [480](#)
 - Clear, [478](#)
 - Decode, [478](#)
 - GetBitSample, [479](#)
 - GetBufferAsRGBA, [479](#)
 - GetLUT, [479](#)
 - GetLUTDescriptor, [479](#)
 - GetLUTLength, [479](#)
 - GetPointer, [479](#)
 - IncompleteLUT, [480](#)
 - InitializeBlueLUT, [479](#)
 - InitializeGreenLUT, [479](#)
 - InitializeLUT, [479](#)
 - InitializeRedLUT, [479](#)
 - Initialized, [479](#)
 - Internal, [480](#)
 - LookupTable, [478](#)
 - LookupTableType, [478](#)
 - Print, [479](#)
 - SetBlueLUT, [479](#)
 - SetGreenLUT, [480](#)
 - SetLUT, [480](#)
 - SetRedLUT, [480](#)
 - WriteBufferAsRGBA, [480](#)
- gdcmm::MD5, [484](#)
 - ~MD5, [485](#)
 - Compute, [485](#)
 - ComputeFile, [485](#)
 - MD5, [485](#)
- gdcmm::Macro, [480](#)
 - AddMacroEntry, [481](#)
 - ArrayIncludeMacrosType, [481](#)
 - Clear, [481](#)
 - FindMacroEntry, [482](#)
 - GetMacroEntry, [482](#)
 - GetName, [482](#)
 - Macro, [481](#)
 - MapModuleEntry, [481](#)
 - operator<<, [482](#)
 - SetName, [482](#)
 - Verify, [482](#)
- gdcmm::Macros, [482](#)
 - AddMacro, [483](#)
 - Clear, [483](#)
 - GetMacro, [483](#)
 - IsEmpty, [483](#)
 - Macros, [483](#)
 - ModuleMapType, [483](#)
 - operator<<, [483](#)
- gdcmm::MediaStorage, [485](#)
 - GetMSString, [491](#)
 - GetMSType, [491](#)
 - GetModality, [491](#)
 - GetModalityDimension, [491](#)
 - GetNumberOfMSString, [491](#)
 - GetNumberOfMSType, [491](#)
 - GetNumberOfModality, [491](#)
 - GetString, [491](#)
 - GuessFromModality, [491](#)
 - IsImage, [491](#)
 - IsUndefined, [491](#)
 - MSType, [488](#)
 - MediaStorage, [491](#)
 - ObjectType, [490](#)
 - operator MSType, [492](#)
 - operator<<, [492](#)
 - SetFromDataSet, [492](#)
 - SetFromFile, [492](#)
 - SetFromHeader, [492](#)
 - SetFromModality, [492](#)
 - SetFromSourceImageSequence, [492](#)
- gdcmm::MemberCommand
 - ~MemberCommand, [495](#)
 - Execute, [495](#)
 - m_ConstMemberFunction, [496](#)
 - m_MemberFunction, [496](#)
 - m_This, [496](#)

- MemberCommand, 495
- New, 495
- Self, 494
- SetCallbackFunction, 495
- TConstMemberFunctionPointer, 494
- TMemberFunctionPointer, 495
- gdcmmemberCommand< T >, 492
- gdcmmeshPrimitive, 496
 - ~MeshPrimitive, 499
 - AddPrimitiveData, 499
 - GetMPType, 499
 - GetMPTypeString, 499
 - GetNumberOfPrimitivesData, 499
 - GetPrimitiveData, 499
 - GetPrimitiveType, 499
 - GetPrimitivesData, 499
 - MPType, 498
 - MeshPrimitive, 499
 - PrimitiveData, 499
 - PrimitiveType, 499
 - PrimitivesData, 498
 - SetPrimitiveData, 499
 - SetPrimitiveType, 499
 - SetPrimitivesData, 499
- gdcmmodifiedEvent, 499
- gdcmmodule, 501
 - AddMacro, 502
 - AddModuleEntry, 502
 - ArrayIncludeMacrosType, 502
 - Clear, 502
 - FindModuleEntryInMacros, 502
 - GetModuleEntryInMacros, 502
 - GetName, 502
 - MapModuleEntry, 502
 - Module, 502
 - operator<<, 502
 - SetName, 502
 - Verify, 502
- gdcmmoduleEntry, 503
 - ~ModuleEntry, 504
 - DataElementType, 505
 - Description, 504
 - DescriptionField, 505
 - GetDescription, 505
 - GetName, 505
 - GetType, 505
 - ModuleEntry, 504
 - Name, 505
 - operator<<, 505
 - SetDescription, 505
 - SetName, 505
 - SetType, 505
- gdcmmodules, 505
 - AddModule, 506
 - Clear, 506
 - GetModule, 506
 - IsEmpty, 506
 - ModuleMapType, 506
 - Modules, 506
 - operator<<, 507
- gdcmmovePatientRootQuery, 507
 - GetAbstractSyntaxUID, 508
 - GetTagListByLevel, 508
 - InitializeDataSet, 508
 - MovePatientRootQuery, 508
 - QueryFactory, 509
 - ValidateQuery, 508
- gdcmmoveStudyRootQuery, 509
 - GetAbstractSyntaxUID, 510
 - GetTagListByLevel, 510
 - InitializeDataSet, 511
 - MoveStudyRootQuery, 510
 - QueryFactory, 511
 - ValidateQuery, 511
- gdcmnestedModuleEntries, 511
 - AddModuleEntry, 513
 - GetModuleEntry, 513
 - GetNumberOfModuleEntries, 513
 - NestedModuleEntries, 513
 - operator<<, 513
 - SizeType, 513
- gdcmnotionEvent, 514
- gdcmmoduleObject, 514
 - ~Object, 516
 - Object, 516
 - operator<<, 516
 - operator=, 516
 - Print, 516
 - Register, 516
 - SmartPointer, 516
 - UnRegister, 516
- gdcmmoduleOrientation, 517
 - ~Orientation, 518
 - GetLabel, 518
 - GetMajorAxisFromPatientRelativeDirectionCosine, 518
 - GetObliquityThresholdCosineValue, 518
 - GetType, 518
 - operator<<, 518
 - Orientation, 518
 - OrientationType, 518
 - Print, 518
 - SetObliquityThresholdCosineValue, 518
- gdcmmoduleOverlay, 519
 - ~Overlay, 522
 - Decode, 522
 - Decompress, 522
 - GetBitPosition, 522

- GetBitsAllocated, [522](#)
- GetBuffer, [522](#)
- GetColumns, [522](#)
- GetDescription, [522](#)
- GetGroup, [522](#)
- GetOrigin, [522](#)
- GetOverlayData, [522](#)
- GetOverlayTypeAsString, [523](#)
- GetOverlayTypeFromString, [523](#)
- GetRows, [523](#)
- GetType, [523](#)
- GetTypeAsEnum, [523](#)
- GetUnpackBuffer, [523](#)
- GetUnpackBufferLength, [523](#)
- GrabOverlayFromPixelData, [523](#)
- IsEmpty, [523](#)
- IsInPixelData, [523](#)
- IsZero, [523](#)
- Overlay, [522](#)
- OverlayType, [521](#)
- Print, [523](#)
- SetBitPosition, [524](#)
- SetBitsAllocated, [524](#)
- SetColumns, [524](#)
- SetDescription, [524](#)
- SetFrameOrigin, [524](#)
- SetGroup, [524](#)
- SetNumberOfFrames, [524](#)
- SetOrigin, [524](#)
- SetOverlay, [524](#)
- SetRows, [524](#)
- SetType, [524](#)
- Update, [525](#)
- gdcmm::PDBelement, [532](#)
 - GetName, [533](#)
 - GetValue, [533](#)
 - NameField, [533](#)
 - operator<<, [533](#)
 - operator==, [533](#)
 - PDBelement, [533](#)
 - SetName, [533](#)
 - SetValue, [533](#)
 - ValueField, [533](#)
- gdcmm::PDBHeader, [534](#)
 - ~PDBHeader, [535](#)
 - FindPDBelementByName, [535](#)
 - GetPDBeEnd, [535](#)
 - GetPDBelementByName, [535](#)
 - GetPDBInfoTag, [535](#)
 - LoadFromDataElement, [535](#)
 - operator<<, [535](#)
 - PDBHeader, [535](#)
 - Print, [535](#)
- gdcmm::PDFCodec, [536](#)
 - ~PDFCodec, [537](#)
 - CanCode, [537](#)
 - CanDecode, [537](#)
 - Decode, [537](#)
 - PDFCodec, [537](#)
- gdcmm::PGXCodec, [540](#)
 - ~PGXCodec, [541](#)
 - CanCode, [541](#)
 - CanDecode, [541](#)
 - GetHeaderInfo, [541](#)
 - PGXCodec, [541](#)
 - Read, [542](#)
 - Write, [542](#)
- gdcmm::PNMCodec, [561](#)
 - ~PNMCodec, [563](#)
 - CanCode, [563](#)
 - CanDecode, [563](#)
 - GetBufferLength, [563](#)
 - GetHeaderInfo, [563](#)
 - PNMCodec, [563](#)
 - Read, [563](#)
 - SetBufferLength, [563](#)
 - Write, [563](#)
- gdcmm::PVRGCodec, [583](#)
 - ~PVRGCodec, [584](#)
 - CanCode, [584](#)
 - CanDecode, [584](#)
 - Code, [584](#)
 - Decode, [584](#)
 - PVRGCodec, [584](#)
- gdcmm::ParseException, [525](#)
 - ~ParseException, [526](#)
 - GetLastElement, [526](#)
 - operator=, [526](#)
 - ParseException, [526](#)
 - SetLastElement, [526](#)
- gdcmm::Parser, [527](#)
 - ~Parser, [528](#)
 - EndElementHandler, [528](#)
 - ErrorType, [528](#)
 - GetBuffer, [528](#)
 - GetCurrentByteIndex, [528](#)
 - GetErrorCode, [528](#)
 - GetErrorString, [528](#)
 - GetUserData, [528](#)
 - Parse, [528](#)
 - ParseBuffer, [529](#)
 - Parser, [528](#)
 - Process, [529](#)
 - SetElementHandler, [529](#)
 - SetUserData, [529](#)
 - StartElementHandler, [528](#)
- gdcmm::Patient, [529](#)
 - Patient, [529](#)

- gdcm::PersonName, 539
 - Component, 539
 - GetMaxLength, 539
 - GetNumberOfComponents, 539
 - MaxLength, 540
 - MaxNumberOfComponents, 540
 - Padding, 540
 - Print, 539
 - Separator, 540
 - SetBlob, 539
 - SetComponents, 539
- gdcm::PhotometricInterpretation, 542
 - GetPIString, 543
 - GetPIType, 543
 - GetSamplesPerPixel, 544
 - GetString, 544
 - GetType, 544
 - IsLossless, 544
 - IsLossy, 544
 - IsRetired, 544
 - IsSameColorSpace, 544
 - operator PIType, 544
 - operator<=, 544
 - PIType, 543
 - PhotometricInterpretation, 543
- gdcm::PixelFormat, 544
 - ~PixelFormat, 546
 - Bitmap, 549
 - GetBitsAllocated, 546
 - GetBitsStored, 547
 - GetHighBit, 547
 - GetMax, 547
 - GetMin, 547
 - GetPixelRepresentation, 547
 - GetPixelSize, 547
 - GetSamplesPerPixel, 547
 - GetScalarType, 548
 - GetScalarTypeAsString, 548
 - IsValid, 548
 - operator ScalarType, 548
 - operator<=, 549
 - operator==, 548
 - PixelFormat, 546
 - Print, 548
 - ScalarType, 546
 - SetBitsAllocated, 548
 - SetBitsStored, 548
 - SetHighBit, 548
 - SetPixelRepresentation, 548
 - SetSamplesPerPixel, 548
 - SetScalarType, 548
 - Validate, 549
- gdcm::Pixmap, 549
 - ~Pixmap, 551
 - AreOverlaysInPixelData, 551
 - Curves, 552
 - GetCurve, 551, 552
 - GetIconImage, 552
 - GetNumberOfCurves, 552
 - GetNumberOfOverlays, 552
 - GetOverlay, 552
 - Icon, 552
 - Overlays, 552
 - Pixmap, 551
 - Print, 552
 - RemoveOverlay, 552
 - SetIconImage, 552
 - SetNumberOfCurves, 552
 - SetNumberOfOverlays, 552
- gdcm::PixmapReader, 552
 - ~PixmapReader, 555
 - GetPixmap, 555
 - PixelData, 556
 - PixmapReader, 555
 - Read, 555
 - ReadACRNEMAImage, 555
 - ReadImage, 555
 - ReadImageInternal, 555
- gdcm::PixmapToPixmapFilter, 556
 - ~PixmapToPixmapFilter, 557
 - GetInput, 558
 - GetOutput, 558
 - GetOutputAsPixmap, 558
 - PixmapToPixmapFilter, 557
- gdcm::PixmapWriter, 558
 - ~PixmapWriter, 560
 - DoIconImage, 560
 - GetImage, 560
 - GetPixmap, 560, 561
 - PixelData, 561
 - PixmapWriter, 560
 - PrepareWrite, 561
 - SetImage, 561
 - SetPixmap, 561
 - Write, 561
- gdcm::Preamble, 564
 - ~Preamble, 565
 - Clear, 565
 - Create, 565
 - GetInternal, 565
 - GetLength, 565
 - IsEmpty, 565
 - IsValid, 565
 - operator<=, 565
 - operator=, 565
 - Preamble, 565
 - Print, 565
 - Read, 565

- Remove, [565](#)
- Valid, [565](#)
- Write, [565](#)
- gdcmm::PresentationContext, [565](#)
 - AddTransferSyntax, [566](#)
 - GetAbstractSyntax, [566](#)
 - GetNumberOfTransferSyntaxes, [567](#)
 - GetPresentationContextID, [567](#)
 - GetTransferSyntax, [567](#)
 - operator==, [567](#)
 - PresentationContext, [566](#)
 - Print, [567](#)
 - SetAbstractSyntax, [567](#)
 - SetPresentationContextID, [567](#)
 - SizeType, [566](#)
 - TransferSyntaxArrayType, [566](#)
- gdcmm::PresentationContextGenerator, [568](#)
 - AddPresentationContext, [569](#)
 - GenerateFromFilenames, [570](#)
 - GenerateFromUID, [570](#)
 - GetDefaultTransferSyntax, [570](#)
 - GetPresentationContexts, [570](#)
 - PresentationContextArrayType, [569](#)
 - PresentationContextGenerator, [569](#)
 - SetDefaultTransferSyntax, [570](#)
 - SetMergeModeToAbstractSyntax, [570](#)
 - SetMergeModeToTransferSyntax, [570](#)
 - SizeType, [569](#)
- gdcmm::Printer, [574](#)
 - ~Printer, [576](#)
 - F, [577](#)
 - GetPrintStyle, [576](#)
 - MaxPrintLength, [577](#)
 - Print, [576](#)
 - PrintDataElement, [576](#)
 - PrintDataSet, [576](#)
 - PrintSQ, [576](#)
 - PrintStyle, [577](#)
 - PrintStyles, [576](#)
 - Printer, [576](#)
 - SetColor, [577](#)
 - SetFile, [577](#)
 - SetStyle, [577](#)
- gdcmm::PrivateDict, [577](#)
 - ~PrivateDict, [578](#)
 - AddDictEntry, [578](#)
 - Dicts, [578](#)
 - FindDictEntry, [578](#)
 - GetDictEntry, [578](#)
 - IsEmpty, [578](#)
 - LoadDefault, [578](#)
 - operator<<, [578](#)
 - PrintXML, [578](#)
 - PrivateDict, [578](#)
 - RemoveDictEntry, [578](#)
- gdcmm::PrivateTag, [579](#)
 - GetOwner, [580](#)
 - operator<, [580](#)
 - operator<<, [580](#)
 - PrivateTag, [580](#)
 - ReadFromCommaSeparatedString, [580](#)
 - SetOwner, [580](#)
- gdcmm::ProgressEvent, [580](#)
 - ~ProgressEvent, [582](#)
 - CheckEvent, [582](#)
 - GetEventName, [582](#)
 - GetProgress, [582](#)
 - MakeObject, [582](#)
 - ProgressEvent, [582](#)
 - Self, [582](#)
 - SetProgress, [582](#)
 - Superclass, [582](#)
- gdcmm::PythonFilter, [585](#)
 - ~PythonFilter, [585](#)
 - GetFile, [585](#)
 - PythonFilter, [585](#)
 - SetDicts, [585](#)
 - SetFile, [585](#)
 - ToPyObject, [585](#)
 - UseDictAlways, [585](#)
- gdcmm::QueryBase, [586](#)
 - ~QueryBase, [587](#)
 - GetAllRequiredTags, [587](#)
 - GetAllTags, [587](#)
 - GetHierarchicalSearchTags, [587](#)
 - GetName, [587](#)
 - GetOptionalTags, [587](#)
 - GetQueryLevel, [587](#)
 - GetRequiredTags, [587](#)
 - GetUniqueTags, [587](#)
- gdcmm::QueryFactory, [588](#)
 - GetCharacterFromCurrentLocale, [588](#)
 - ListCharSets, [588](#)
 - ProduceCharacterSetDataElement, [588](#)
 - ProduceQuery, [589](#)
- gdcmm::QueryImage, [589](#)
 - GetHierarchicalSearchTags, [590](#)
 - GetName, [590](#)
 - GetOptionalTags, [590](#)
 - GetQueryLevel, [590](#)
 - GetRequiredTags, [591](#)
 - GetUniqueTags, [591](#)
- gdcmm::QueryPatient, [591](#)
 - GetHierarchicalSearchTags, [592](#)
 - GetName, [592](#)
 - GetOptionalTags, [592](#)
 - GetQueryLevel, [592](#)
 - GetRequiredTags, [593](#)

- GetUniqueTags, [593](#)
- gdcm::QuerySeries, [593](#)
 - GetHierarchicalSearchTags, [594](#)
 - GetName, [594](#)
 - GetOptionalTags, [594](#)
 - GetQueryLevel, [594](#)
 - GetRequiredTags, [595](#)
 - GetUniqueTags, [595](#)
- gdcm::QueryStudy, [595](#)
 - GetHierarchicalSearchTags, [596](#)
 - GetName, [596](#)
 - GetOptionalTags, [596](#)
 - GetQueryLevel, [596](#)
 - GetRequiredTags, [597](#)
 - GetUniqueTags, [597](#)
- gdcm::RAWCodec, [597](#)
 - ~RAWCodec, [598](#)
 - CanCode, [598](#)
 - CanDecode, [599](#)
 - Code, [599](#)
 - Decode, [599](#)
 - DecodeByStreams, [599](#)
 - DecodeBytes, [599](#)
 - GetHeaderInfo, [599](#)
 - RAWCodec, [598](#)
- gdcm::RLECodec, [609](#)
 - ~RLECodec, [611](#)
 - CanCode, [611](#)
 - CanDecode, [611](#)
 - Code, [611](#)
 - Decode, [612](#)
 - DecodeByStreams, [612](#)
 - DecodeExtent, [612](#)
 - GetBufferLength, [612](#)
 - GetHeaderInfo, [612](#)
 - ImageRegionReader, [612](#)
 - RLECodec, [611](#)
 - SetBufferLength, [612](#)
 - SetLength, [612](#)
- gdcm::Reader, [599](#)
 - ~Reader, [602](#)
 - CanRead, [602](#)
 - F, [604](#)
 - GetFile, [602](#), [603](#)
 - GetStreamPtr, [603](#)
 - Read, [603](#)
 - ReadDataSet, [603](#)
 - ReadMetaInformation, [603](#)
 - ReadPreamble, [603](#)
 - ReadSelectedTags, [603](#)
 - ReadUpToTag, [603](#)
 - Reader, [602](#)
 - SetFile, [603](#)
 - SetFileName, [603](#)
 - SetStream, [604](#)
 - StreamImageReader, [604](#)
- gdcm::Region, [604](#)
 - ~Region, [605](#)
 - Area, [605](#)
 - Clone, [605](#)
 - ComputeBoundingBox, [606](#)
 - Empty, [606](#)
 - IsValid, [606](#)
 - Print, [606](#)
 - Region, [605](#)
- gdcm::Rescaler, [606](#)
 - ~Rescaler, [608](#)
 - ComputeInterceptSlopePixelType, [608](#)
 - ComputePixelTypeFromMinMax, [608](#)
 - GetIntercept, [608](#)
 - GetSlope, [608](#)
 - InverseRescale, [608](#)
 - InverseRescaleFunctionIntoBestFit, [608](#)
 - Rescale, [608](#)
 - RescaleFunctionIntoBestFit, [608](#)
 - Rescaler, [608](#)
 - SetIntercept, [608](#)
 - SetMinMaxForPixelType, [608](#)
 - SetPixelFormat, [608](#)
 - SetSlope, [609](#)
 - SetTargetPixelType, [609](#)
 - SetUseTargetPixelType, [609](#)
- gdcm::SHA1, [651](#)
 - ~SHA1, [651](#)
 - Compute, [651](#)
 - ComputeFile, [652](#)
 - SHA1, [651](#)
- gdcm::SOPClassUIDToIOD, [660](#)
 - const, [661](#)
 - GetIOD, [661](#)
- gdcm::Scanner, [614](#)
 - ~Scanner, [618](#)
 - AddPrivateTag, [618](#)
 - AddSkipTag, [618](#)
 - AddTag, [618](#)
 - Begin, [618](#)
 - ClearSkipTags, [618](#)
 - ClearTags, [618](#)
 - ConstIterator, [617](#)
 - End, [618](#)
 - GetAllFilenamesFromTagToValue, [618](#)
 - GetFilenameFromTagToValue, [618](#)
 - GetFilenames, [618](#)
 - GetKeys, [618](#)
 - GetMapping, [619](#)
 - GetMappingFromTagToValue, [619](#)
 - GetMappings, [619](#)
 - GetOrderedValues, [619](#)

- GetValue, 619
- GetValues, 619
- IsKey, 619
- MappingType, 617
- New, 620
- operator<<, 620
- Print, 620
- ProcessPublicTag, 620
- Scan, 620
- Scanner, 618
- TagToValue, 617
- TagToValueValueType, 617
- ValueType, 618
- gdcm::Scanner::ltstr, 480
 - operator(), 480
- gdcm::Segment, 620
 - ~Segment, 623
 - ALGOType, 623
 - AddSurface, 623
 - AnatomicRegion, 624
 - GetALGOType, 623
 - GetALGOTypeString, 623
 - GetAnatomicRegion, 623
 - GetPropertyCategory, 623
 - GetPropertyType, 623
 - GetSegmentAlgorithmName, 623
 - GetSegmentAlgorithmType, 623
 - GetSegmentDescription, 623
 - GetSegmentLabel, 623
 - GetSegmentNumber, 623
 - GetSurface, 623
 - GetSurfaceCount, 623
 - GetSurfaces, 624
 - PropertyCategory, 624
 - PropertyType, 624
 - Segment, 623
 - SegmentAlgorithmName, 624
 - SegmentAlgorithmType, 624
 - SegmentDescription, 624
 - SegmentLabel, 624
 - SegmentNumber, 624
 - SetAnatomicRegion, 624
 - SetPropertyCategory, 624
 - SetPropertyType, 624
 - SetSegmentAlgorithmName, 624
 - SetSegmentAlgorithmType, 624
 - SetSegmentDescription, 624
 - SetSegmentLabel, 624
 - SetSegmentNumber, 624
 - SetSurfaceCount, 624
 - SurfaceCount, 624
 - SurfaceVector, 623
 - Surfaces, 624
- gdcm::SegmentHelper, 130
 - GetValue, 619
 - GetValues, 619
 - IsKey, 619
 - MappingType, 617
 - New, 620
 - operator<<, 620
 - Print, 620
 - ProcessPublicTag, 620
 - Scan, 620
 - Scanner, 618
 - TagToValue, 617
 - TagToValueValueType, 617
 - ValueType, 618
- gdcm::SegmentHelper::BasicCodedEntry, 200
 - BasicCodedEntry, 202
 - CM, 202
 - CSD, 202
 - CSV, 202
 - CV, 202
 - IsEmpty, 202
- gdcm::SegmentReader, 626
 - ~SegmentReader, 628
 - GetSegments, 629
 - Read, 629
 - ReadSegment, 629
 - ReadSegments, 629
 - SegmentMap, 628
 - SegmentReader, 628
 - SegmentVector, 628
 - Segments, 629
- gdcm::SegmentWriter, 629
 - ~SegmentWriter, 631
 - AddSegment, 631
 - GetNumberOfSegments, 631
 - GetSegment, 631
 - GetSegments, 631
 - PrepareWrite, 631
 - SegmentVector, 631
 - SegmentWriter, 631
 - Segments, 631
 - SetNumberOfSegments, 631
 - SetSegments, 631
 - Write, 631
- gdcm::SegmentedPaletteColorLookupTable, 625
 - ~SegmentedPaletteColorLookupTable, 626
 - Print, 626
 - SegmentedPaletteColorLookupTable, 626
 - SetLUT, 626
- gdcm::SequenceOfFragments, 631
 - AddFragment, 634
 - Begin, 634
 - Clear, 634
 - ComputeByteLength, 634
 - ComputeLength, 634
 - ConstIterator, 634
 - End, 634
 - FragmentVector, 634
 - GetBuffer, 634
 - GetFragBuffer, 634
 - GetFragment, 634
 - GetLength, 635
 - GetNumberOfFragments, 635
 - GetTable, 635
 - Iterator, 634
 - New, 635
 - operator==, 635
 - Print, 635

- Read, [635](#)
- ReadPreValue, [635](#)
- ReadValue, [635](#)
- SequenceOfFragments, [634](#)
- SetLength, [635](#)
- SizeType, [634](#)
- Write, [636](#)
- WriteBuffer, [636](#)
- gdcmm::SequenceOfItems, [636](#)
 - AddItem, [639](#)
 - Begin, [639](#)
 - Clear, [639](#)
 - ComputeLength, [639](#)
 - ConstIterator, [639](#)
 - End, [639](#)
 - FindDataElement, [640](#)
 - GetItem, [640](#)
 - GetLength, [640](#)
 - GetNumberOfItems, [640](#)
 - IsUndefinedLength, [640](#)
 - ItemVector, [639](#)
 - Items, [641](#)
 - Iterator, [639](#)
 - New, [640](#)
 - operator=, [640](#)
 - operator==, [640](#)
 - Print, [640](#)
 - Read, [640](#)
 - SequenceLengthField, [641](#)
 - SequenceOfItems, [639](#)
 - SetLength, [641](#)
 - SetLengthToUndefined, [641](#)
 - SetNumberOfItems, [641](#)
 - SizeType, [639](#)
 - Write, [641](#)
- gdcmm::SerieHelper, [642](#)
 - ~SerieHelper, [643](#)
 - AddFile, [644](#)
 - AddFileName, [644](#)
 - AddRestriction, [644](#)
 - Clear, [644](#)
 - CreateDefaultUniqueSeriesIdentifier, [644](#)
 - CreateUniqueSeriesIdentifier, [644](#)
 - FileNameOrdering, [644](#)
 - GetFirstSingleSerieUIDFileSet, [644](#)
 - GetNextSingleSerieUIDFileSet, [644](#)
 - ImagePositionPatientOrdering, [644](#)
 - ItFileSetHt, [644](#)
 - OrderFileList, [644](#)
 - SerieHelper, [643](#)
 - SerieRestrictions, [643](#)
 - SetDirectory, [644](#)
 - SetLoadMode, [644](#)
 - SetUseSeriesDetails, [644](#)
 - SingleSerieUIDFileSetHT, [644](#)
 - SingleSerieUIDFileSetmap, [643](#)
 - UserOrdering, [644](#)
- gdcmm::SerieHelper::Rule, [613](#)
 - elem, [614](#)
 - group, [614](#)
 - op, [614](#)
 - value, [614](#)
- gdcmm::Series, [644](#)
 - Series, [645](#)
- gdcmm::ServiceClassUser, [646](#)
 - ~ServiceClassUser, [648](#)
 - GetAETitle, [648](#)
 - GetCalledAETitle, [648](#)
 - GetTimeout, [648](#)
 - InitializeConnection, [648](#)
 - IsPresentationContextAccepted, [649](#)
 - SendEcho, [649](#)
 - SendFind, [649](#)
 - SendMove, [649](#)
 - SendStore, [649](#)
 - ServiceClassUser, [648](#)
 - SetAETitle, [649](#)
 - SetCalledAETitle, [649](#)
 - SetHostname, [650](#)
 - SetPort, [650](#)
 - SetPortSCP, [650](#)
 - SetPresentationContexts, [650](#)
 - SetTimeout, [650](#)
 - StartAssociation, [650](#)
 - StopAssociation, [650](#)
- gdcmm::SimpleMemberCommand
 - ~SimpleMemberCommand, [654](#)
 - Execute, [654](#)
 - m_MemberFunction, [655](#)
 - m_This, [655](#)
 - New, [655](#)
 - Self, [654](#)
 - SetCallbackFunction, [655](#)
 - SimpleMemberCommand, [654](#)
 - TMemberFunctionPointer, [654](#)
- gdcmm::SimpleMemberCommand< T >, [652](#)
- gdcmm::SimpleSubjectWatcher, [655](#)
 - ~SimpleSubjectWatcher, [656](#)
 - EndFilter, [656](#)
 - ShowAbort, [656](#)
 - ShowAnonymization, [656](#)
 - ShowData, [656](#)
 - ShowDataSet, [656](#)
 - ShowIteration, [656](#)
 - ShowProgress, [656](#)
 - SimpleSubjectWatcher, [656](#)
 - StartFilter, [656](#)
 - TestAbortOff, [656](#)

- TestAbortOn, [656](#)
- gdcmm::SmartPointer
 - ~SmartPointer, [658](#)
 - GetPointer, [659](#)
 - operator ObjectType *, [659](#)
 - operator*, [659](#)
 - operator->, [659](#)
 - operator=, [659](#)
 - SmartPointer, [658](#)
- gdcmm::SmartPointer< ObjectType >, [657](#)
- gdcmm::Sorter, [661](#)
 - ~Sorter, [664](#)
 - AddSelect, [664](#)
 - FileNames, [665](#)
 - GetFileNames, [664](#)
 - operator<<, [665](#)
 - Print, [664](#)
 - Selection, [665](#)
 - SelectionMap, [663](#)
 - SetSortFunction, [664](#)
 - Sort, [664](#)
 - SortFunc, [665](#)
 - SortFunction, [663](#)
 - Sorter, [664](#)
 - StableSort, [664](#)
- gdcmm::Spacing, [665](#)
 - ~Spacing, [666](#)
 - ComputePixelAspectRatioFromPixelSpacing, [666](#)
 - Spacing, [666](#)
 - SpacingType, [666](#)
- gdcmm::Spectroscopy, [667](#)
 - Spectroscopy, [667](#)
- gdcmm::SplitMosaicFilter, [667](#)
 - ~SplitMosaicFilter, [668](#)
 - ComputeMOSAICDimensions, [668](#)
 - GetFile, [668](#)
 - GetImage, [668](#)
 - SetFile, [668](#)
 - SetImage, [668](#)
 - Split, [668](#)
 - SplitMosaicFilter, [668](#)
- gdcmm::StartEvent, [668](#)
- gdcmm::StreamImageReader, [670](#)
 - ~StreamImageReader, [671](#)
 - CanReadImage, [671](#)
 - DefinePixelExtent, [671](#)
 - DefineProperBufferLength, [672](#)
 - GetDimensionsValueForResolution, [672](#)
 - GetFile, [672](#)
 - Read, [672](#)
 - ReadImageInformation, [672](#)
 - SetFileName, [672](#)
 - SetStream, [673](#)
 - StreamImageReader, [671](#)
- gdcmm::StreamImageWriter, [673](#)
 - ~StreamImageWriter, [676](#)
 - CanWriteFile, [676](#)
 - DefinePixelExtent, [676](#)
 - DefineProperBufferLength, [676](#)
 - mElementOffsets, [678](#)
 - mElementOffsets1, [678](#)
 - mWriter, [678](#)
 - mXMax, [678](#)
 - mXMin, [678](#)
 - mYMax, [678](#)
 - mYMin, [678](#)
 - mZMax, [678](#)
 - mZMin, [678](#)
 - mspFile, [678](#)
 - SetFile, [676](#)
 - SetFileName, [676](#)
 - SetStream, [676](#)
 - StreamImageWriter, [676](#)
 - Write, [677](#)
 - WriteImageInformation, [677](#)
 - WriteImageSubregionRAW, [677](#)
 - WriteRawHeader, [677](#)
- gdcmm::String
 - const_iterator, [680](#)
 - const_reference, [680](#)
 - const_reverse_iterator, [681](#)
 - difference_type, [681](#)
 - IsValid, [681](#)
 - iterator, [681](#)
 - operator const char *, [681](#)
 - pointer, [681](#)
 - reference, [681](#)
 - reverse_iterator, [681](#)
 - size_type, [681](#)
 - String, [681](#)
 - Trim, [682](#)
 - Truncate, [682](#)
 - value_type, [681](#)
- gdcmm::String< TDelimiter, TMaxLength, TPadChar >, [678](#)
- gdcmm::StringFilter, [682](#)
 - ~StringFilter, [683](#)
 - ExecuteQuery, [683](#)
 - FromString, [683](#)
 - GetFile, [683](#)
 - SetDicts, [683](#)
 - SetFile, [683](#)
 - StringFilter, [683](#)
 - ToString, [684](#)
 - ToStringPair, [684](#)
 - UseDictAlways, [684](#)
- gdcmm::Study, [684](#)
 - Study, [685](#)
- gdcmm::Subject, [685](#)

- ~Subject, 686
- AddObserver, 686
- GetCommand, 686
- HasObserver, 686
- InvokeEvent, 686, 687
- RemoveAllObservers, 687
- RemoveObserver, 687
- Subject, 686
- gdcmm::Surface, 687
 - ~Surface, 690
 - GetAlgorithmFamily, 691
 - GetAlgorithmName, 691
 - GetAlgorithmVersion, 691
 - GetAxisOfRotation, 691
 - GetCenterOfRotation, 691
 - GetFiniteVolume, 691
 - GetManifold, 691
 - GetMaximumPointDistance, 691
 - GetMeanPointDistance, 691
 - GetMeshPrimitive, 691
 - GetNumberOfSurfacePoints, 691
 - GetNumberOfVectors, 691
 - GetPointCoordinatesData, 691
 - GetPointPositionAccuracy, 691
 - GetPointsBoundingBoxCoordinates, 691
 - GetProcessingAlgorithm, 692
 - GetRecommendedDisplayCIELabValue, 692
 - GetRecommendedDisplayGrayscaleValue, 692
 - GetRecommendedPresentationOpacity, 692
 - GetRecommendedPresentationType, 692
 - GetSTATES, 692
 - GetSTATESString, 692
 - GetSurfaceComments, 692
 - GetSurfaceNumber, 692
 - GetSurfaceProcessing, 692
 - GetSurfaceProcessingDescription, 692
 - GetSurfaceProcessingRatio, 692
 - GetVIEWType, 692
 - GetVIEWTypeString, 692
 - GetVectorAccuracy, 692
 - GetVectorCoordinateData, 692
 - GetVectorDimensionality, 692
 - STATES, 690
 - SetAlgorithmFamily, 692
 - SetAlgorithmName, 692
 - SetAlgorithmVersion, 692
 - SetAxisOfRotation, 692
 - SetCenterOfRotation, 692
 - SetFiniteVolume, 693
 - SetManifold, 693
 - SetMaximumPointDistance, 693
 - SetMeanPointDistance, 693
 - SetMeshPrimitive, 693
 - SetNumberOfSurfacePoints, 693
 - SetNumberOfVectors, 693
 - SetPointCoordinatesData, 693
 - SetPointPositionAccuracy, 693
 - SetPointsBoundingBoxCoordinates, 693
 - SetProcessingAlgorithm, 693
 - SetRecommendedDisplayCIELabValue, 693
 - SetRecommendedDisplayGrayscaleValue, 693
 - SetRecommendedPresentationOpacity, 693
 - SetRecommendedPresentationType, 693
 - SetSurfaceComments, 693
 - SetSurfaceNumber, 693
 - SetSurfaceProcessing, 693
 - SetSurfaceProcessingDescription, 693
 - SetSurfaceProcessingRatio, 693
 - SetVectorAccuracy, 693
 - SetVectorCoordinateData, 693
 - SetVectorDimensionality, 693
 - Surface, 690
 - VIEWType, 690
- gdcmm::SurfaceHelper, 694
 - ColorArray, 694
 - RGBToRecommendedDisplayCIELab, 695
 - RGBToRecommendedDisplayGrayscale, 696
 - RecommendedDisplayCIELabToRGB, 694, 695
- gdcmm::SurfaceReader, 696
 - ~SurfaceReader, 698
 - GetNumberOfSurfaces, 698
 - Read, 698
 - ReadPointMacro, 698
 - ReadSurface, 698
 - ReadSurfaces, 698
 - SurfaceReader, 698
- gdcmm::SurfaceWriter, 698
 - ~SurfaceWriter, 700
 - ComputeNumberOfSurfaces, 700
 - GetNumberOfSurfaces, 700
 - NumberOfSurfaces, 700
 - PrepareWrite, 700
 - PrepareWritePointMacro, 700
 - SetNumberOfSurfaces, 700
 - SurfaceWriter, 700
 - Write, 700
- gdcmm::SwapCode, 700
 - GetIndex, 702
 - GetSwapCodeString, 702
 - operator SwapCode::SwapCodeType, 702
 - operator<<, 702
 - SwapCode, 702
 - SwapCodeType, 701
- gdcmm::SwapperDoOp, 702
 - Swap, 702
 - SwapArray, 702
- gdcmm::SwapperNoOp, 703
 - Swap, 703

- SwapArray, [703](#)
- gdcmm::System, [703](#)
 - DeleteDirectory, [704](#)
 - EncodeBytes, [704](#)
 - FileExists, [704](#)
 - FileIsDirectory, [705](#)
 - FileIsSymlink, [705](#)
 - FileSize, [705](#)
 - FileTime, [705](#)
 - FormatDateTime, [705](#)
 - GetCWD, [706](#)
 - GetCurrentDateTime, [705](#)
 - GetCurrentModuleFileName, [705](#)
 - GetCurrentProcessFileName, [706](#)
 - GetCurrentResourcesDirectory, [706](#)
 - GetHostName, [706](#)
 - GetLastSystemError, [706](#)
 - GetLocaleCharSet, [706](#)
 - GetPermissions, [706](#)
 - GetTimezoneOffsetFromUTC, [706](#)
 - MakeDirectory, [706](#)
 - ParseDateTime, [706](#), [707](#)
 - RemoveFile, [707](#)
 - SetPermissions, [707](#)
 - StrCaseCmp, [707](#)
 - StrNCaseCmp, [707](#)
 - StrTokR, [707](#)
- gdcmm::Table, [707](#)
 - ~Table, [708](#)
 - GetTableEntry, [708](#)
 - InsertEntry, [708](#)
 - MapTableEntry, [708](#)
 - operator<<, [708](#)
 - Table, [708](#)
- gdcmm::TableEntry, [708](#)
 - ~TableEntry, [709](#)
 - TableEntry, [709](#)
- gdcmm::TableReader, [709](#)
 - ~TableReader, [710](#)
 - CharacterDataHandler, [710](#)
 - EndElement, [710](#)
 - GetDefs, [710](#)
 - GetFilename, [710](#)
 - HandleIOD, [710](#)
 - HandleIODEntry, [710](#)
 - HandleMacro, [710](#)
 - HandleMacroEntry, [710](#)
 - HandleMacroEntryDescription, [710](#)
 - HandleModule, [710](#)
 - HandleModuleEntry, [710](#)
 - HandleModuleEntryDescription, [711](#)
 - HandleModuleInclude, [711](#)
 - Read, [711](#)
 - SetFilename, [711](#)
 - StartElement, [711](#)
 - TableReader, [710](#)
- gdcmm::Tag, [712](#)
 - bytes, [718](#)
 - GetElement, [714](#)
 - GetElementTag, [715](#)
 - GetGroup, [715](#)
 - GetLength, [715](#)
 - GetPrivateCreator, [715](#)
 - IsGroupLength, [715](#)
 - IsGroupXX, [715](#)
 - IsIllegal, [715](#)
 - IsPrivate, [715](#)
 - IsPrivateCreator, [716](#)
 - IsPublic, [716](#)
 - operator<, [716](#)
 - operator<<, [718](#)
 - operator<=, [716](#)
 - operator>>, [718](#)
 - operator=, [716](#)
 - operator==, [716](#)
 - PrintAsPipeSeparatedString, [717](#)
 - Read, [717](#)
 - ReadFromCommaSeparatedString, [717](#)
 - ReadFromPipeSeparatedString, [717](#)
 - SetElement, [717](#)
 - SetElementTag, [717](#)
 - SetGroup, [717](#)
 - SetPrivateCreator, [718](#)
 - Tag, [714](#)
 - tag, [718](#)
 - tags, [718](#)
 - Write, [718](#)
- gdcmm::TagPath, [718](#)
 - ~TagPath, [719](#)
 - ConstructFromString, [719](#)
 - ConstructFromTagList, [719](#)
 - IsValid, [719](#)
 - Print, [719](#)
 - Push, [720](#)
 - TagPath, [719](#)
- gdcmm::Testing, [720](#)
 - ~Testing, [721](#)
 - ComputeFileMD5, [721](#)
 - ComputeMD5, [721](#)
 - GetDataExtraRoot, [722](#)
 - GetDataRoot, [722](#)
 - GetFilename, [722](#)
 - GetFileNames, [722](#)
 - GetLossyFlagFromFile, [722](#)
 - GetMD5DataImage, [722](#)
 - GetMD5DataImages, [722](#)
 - GetMD5FromBrokenFile, [722](#)
 - GetMD5FromFile, [722](#)

- GetMediaStorageDataFile, [723](#)
- GetMediaStorageDataFiles, [723](#)
- GetMediaStorageFromFile, [723](#)
- GetNumberOfFileNames, [723](#)
- GetNumberOfMD5DataImages, [723](#)
- GetNumberOfMediaStorageDataFiles, [723](#)
- GetPixelSpacingDataRoot, [723](#)
- GetSelectedTagsOffsetFromFile, [723](#)
- GetSourceDirectory, [723](#)
- GetStreamOffsetFromFile, [723](#)
- GetTempDirectory, [723](#)
- GetTempDirectoryW, [723](#)
- GetTempFilename, [723](#)
- GetTempFilenameW, [723](#)
- MD5DataImagesType, [721](#)
- MediaStorageDataFilesType, [721](#)
- Print, [723](#)
- Testing, [721](#)
- gdcm::Trace, [724](#)
 - ~Trace, [725](#)
 - DebugOff, [725](#)
 - DebugOn, [725](#)
 - ErrorOff, [725](#)
 - ErrorOn, [725](#)
 - GetDebugFlag, [725](#)
 - GetDebugStream, [725](#)
 - GetErrorFlag, [725](#)
 - GetErrorStream, [725](#)
 - GetStream, [725](#)
 - GetWarningFlag, [725](#)
 - GetWarningStream, [726](#)
 - SetDebug, [726](#)
 - SetDebugStream, [726](#)
 - SetError, [726](#)
 - SetErrorStream, [726](#)
 - SetStream, [726](#)
 - SetStreamToFile, [726](#)
 - SetWarning, [726](#)
 - SetWarningStream, [726](#)
 - Trace, [725](#)
 - WarningOff, [726](#)
 - WarningOn, [727](#)
- gdcm::TransferSyntax, [727](#)
 - CanStoreLossy, [729](#)
 - GetNegociatedType, [729](#)
 - GetString, [730](#)
 - GetSwapCode, [730](#)
 - GetTSString, [730](#)
 - GetTSType, [730](#)
 - IsEncapsulated, [730](#)
 - IsEncoded, [730](#)
 - IsExplicit, [730](#)
 - IsImplicit, [730](#)
 - IsLossless, [730](#)
 - IsLossy, [730](#)
 - IsValid, [730](#)
 - NegociatedType, [729](#)
 - operator TSType, [730](#)
 - operator<<, [730](#)
 - TSType, [729](#)
 - TransferSyntax, [729](#)
- gdcm::Type, [733](#)
 - GetTypeString, [735](#)
 - GetTypeType, [735](#)
 - operator TypeType, [735](#)
 - operator<<, [735](#)
 - Type, [735](#)
 - TypeType, [734](#)
- gdcm::UI, [735](#)
 - Internal, [735](#)
 - operator<<, [735](#)
- gdcm::UIDGenerator, [736](#)
 - Generate, [737](#)
 - GenerateUUID, [737](#)
 - GetGDCMUID, [737](#)
 - GetRoot, [737](#)
 - IsValid, [737](#)
 - SetRoot, [737](#)
 - UIDGenerator, [736](#)
- gdcm::UIDs, [737](#)
 - GetName, [755](#)
 - GetNumberOfTransferSyntaxStrings, [755](#)
 - GetString, [756](#)
 - GetTransferSyntaxString, [756](#)
 - GetTransferSyntaxStrings, [756](#)
 - GetUIDName, [756](#)
 - GetUIDString, [756](#)
 - operator TSType, [756](#)
 - SetFromUID, [756](#)
 - TSName, [742](#)
 - TSType, [749](#)
 - TransferSyntaxStringsType, [742](#)
- gdcm::UNExplicitDataElement, [804](#)
 - GetLength, [805](#)
 - Read, [805](#)
 - ReadPreValue, [806](#)
 - ReadValue, [806](#)
 - ReadWithLength, [806](#)
- gdcm::UNExplicitImplicitDataElement, [806](#)
 - GetLength, [807](#)
 - Read, [808](#)
 - ReadPreValue, [808](#)
 - ReadValue, [808](#)
- gdcm::Unpacker12Bits, [808](#)
 - Pack, [808](#)
 - Unpack, [808](#)
- gdcm::Usage, [809](#)
 - GetUsageString, [810](#)

- GetUsageType, 810
- operator UsageType, 810
- operator<<, 810
- Usage, 810
- UsageType, 810
- gdcmm::UserEvent, 810
- gdcmm::VL, 818
 - GetLength, 819
 - GetVL16Max, 819
 - GetVL32Max, 819
 - IsOdd, 819
 - IsUndefined, 819
 - operator uint32_t, 819
 - operator<<, 820
 - operator++, 819
 - operator+==, 819
 - Read, 819
 - Read16, 820
 - SetToUndefined, 820
 - Type, 819
 - VL, 819
 - Write, 820
 - Write16, 820
- gdcmm::VM, 820
 - Compatible, 823
 - GetIndex, 823
 - GetLength, 823
 - GetNumberOfElementsFromArray, 823
 - GetVMString, 823
 - GetVMType, 823
 - GetVMTypeFromLength, 824
 - IsValid, 824
 - operator VMType, 824
 - operator<<, 824
 - VM, 823
 - VMType, 822
- gdcmm::VMToLength< T >, 824
- gdcmm::VR, 824
 - CanDisplay, 827
 - Compatible, 827
 - GetLength, 827, 828
 - GetSize, 828
 - GetSizeof, 828
 - GetVRString, 828
 - GetVRStringFromFile, 828
 - GetVRType, 828
 - GetVRTypeFromFile, 828
 - IsASCII, 828
 - IsASCII2, 828
 - IsBinary, 828
 - IsBinary2, 828
 - IsDual, 828
 - IsSwap, 828
 - IsVRFile, 828
 - IsValid, 828
 - operator VRTYPE, 828
 - Read, 828
 - VR, 827
 - VRTYPE, 826
 - Write, 828
- gdcmm::VR16ExplicitDataElement, 829
 - GetLength, 830
 - Read, 830
 - ReadPreValue, 831
 - ReadValue, 831
 - ReadWithLength, 831
- gdcmm::VRToEncoding< T >, 831
- gdcmm::VRToType< T >, 831
- gdcmm::VRVLSIZE< 0 >, 832
 - Read, 832
 - Write, 832
- gdcmm::VRVLSIZE< 1 >, 832
 - Read, 832
 - Write, 832
- gdcmm::VRVLSIZE< T >, 832
- gdcmm::Validate, 813
 - ~Validate, 814
 - F, 814
 - GetValidatedFile, 814
 - SetFile, 814
 - V, 814
 - Validate, 814
 - Validation, 814
- gdcmm::Value, 814
 - ~Value, 815
 - Clear, 815
 - GetLength, 815
 - operator==, 816
 - SetLength, 816
 - Value, 815
- gdcmm::ValueIO
 - Read, 816
 - Write, 816
- gdcmm::ValueIO< TDE, TSwap, TType >, 816
- gdcmm::Version, 817
 - ~Version, 817
 - GetBuildVersion, 817
 - GetMajorVersion, 817
 - GetMinorVersion, 817
 - GetVersion, 817
 - operator<<, 818
 - Print, 817
 - Version, 817
- gdcmm::Waveform, 883
 - Waveform, 883
- gdcmm::Writer, 883
 - ~Writer, 887

- CheckFileMetaInformationOff, 887
- CheckFileMetaInformationOn, 887
- GetFile, 887
- GetStreamPtr, 887
- Ofstream, 888
- SetCheckFileMetaInformation, 887
- SetFile, 887
- SetFileName, 888
- SetStream, 888
- SetWriteDataSetOnly, 888
- Stream, 888
- StreamImageWriter, 888
- Write, 888
- Writer, 887
- gdcmm::XMLDictReader, 889
 - ~XMLDictReader, 890
 - CharacterDataHandler, 890
 - EndElement, 890
 - GetDict, 890
 - HandleDescription, 890
 - HandleEntry, 890
 - StartElement, 890
 - XMLDictReader, 890
- gdcmm::XMLPrivateDictReader, 890
 - ~XMLPrivateDictReader, 892
 - CharacterDataHandler, 892
 - EndElement, 892
 - GetPrivateDict, 892
 - HandleDescription, 892
 - HandleEntry, 892
 - StartElement, 892
 - XMLPrivateDictReader, 892
- gdcmm::ignore_char, 395
 - ignore_char, 396
 - m_char, 396
- gdcmm::network, 124
 - cMaxEventID, 129
 - cMaxStateID, 129
 - EEventID, 128
 - EStateID, 129
 - GetStateIndex, 129
- gdcmm::network::AAAbortPDU, 133
 - AAAbortPDU, 134
 - IsLastFragment, 134
 - Print, 134
 - Read, 134
 - SetReason, 134
 - SetSource, 135
 - Size, 135
 - Write, 135
- gdcmm::network::AAssociateACPDU, 135
 - AAssociateACPDU, 137
 - AAssociateRQPDU, 137
 - AddPresentationContextAC, 137
 - GetNumberOfPresentationContextAC, 137
 - GetPresentationContextAC, 137
 - GetUserInformation, 137
 - InitFromRQ, 137
 - IsLastFragment, 137
 - Print, 137
 - Read, 137
 - SetCalledAETitle, 137
 - SetCallingAETitle, 137
 - Size, 137
 - SizeType, 137
 - Write, 137
- gdcmm::network::AAssociateRJPDU, 138
 - AAssociateRJPDU, 139
 - IsLastFragment, 139
 - Print, 139
 - Read, 139
 - Size, 139
 - Write, 139
- gdcmm::network::AAssociateRQPDU, 139
 - AAssociateACPDU, 143
 - AAssociateRQPDU, 141
 - AddPresentationContext, 142
 - GetCalledAETitle, 142
 - GetCallingAETitle, 142
 - GetNumberOfPresentationContext, 142
 - GetPresentationContext, 142
 - GetPresentationContextByAbstractSyntax, 142
 - GetPresentationContextByID, 142
 - GetPresentationContexts, 142
 - GetReserved43_74, 142
 - GetUserInformation, 142
 - IsAETitleValid, 142
 - IsLastFragment, 142
 - PresentationContextArrayType, 141
 - Print, 142
 - Read, 142
 - SetCalledAETitle, 142
 - SetCallingAETitle, 142
 - SetUserInformation, 142
 - Size, 143
 - SizeType, 141
 - Write, 143
- gdcmm::network::ARTIMTimer, 160
 - ARTIMTimer, 161
 - GetElapsedTime, 161
 - GetHasExpired, 161
 - GetTimeout, 161
 - SetTimeout, 161
 - Start, 161
 - Stop, 161
- gdcmm::network::AReleaseRPPDU, 157
 - AReleaseRPPDU, 158
 - IsLastFragment, 158

- Print, 158
- Read, 158
- Size, 158
- Write, 158
- gdcmm::network::AReleaseRQPDU, 158
 - AReleaseRQPDU, 159
 - IsLastFragment, 160
 - Print, 160
 - Read, 160
 - Size, 160
 - Write, 160
- gdcmm::network::AbstractSyntax, 144
 - AbstractSyntax, 145
 - GetAsDataElement, 145
 - GetName, 145
 - operator==, 145
 - Print, 145
 - Read, 145
 - SetName, 145
 - SetNameFromUID, 145
 - Size, 145
 - Write, 145
- gdcmm::network::ApplicationContext, 154
 - ApplicationContext, 154
 - GetName, 155
 - Print, 155
 - Read, 155
 - SetName, 155
 - Size, 155
 - Write, 155
- gdcmm::network::AsynchronousOperationsWindowSub, 162
 - AsynchronousOperationsWindowSub, 163
 - Print, 163
 - Read, 163
 - Size, 163
 - Write, 163
- gdcmm::network::BaseCompositeMessage, 192
 - ConstructPDV, 194
- gdcmm::network::BasePDU, 194
 - ~BasePDU, 196
 - IsLastFragment, 196
 - Print, 196
 - Read, 196
 - Size, 196
 - Write, 196
- gdcmm::network::CEchoRQ, 225
 - AffectedSOPClassUID, 227
 - ConstructPDV, 227
 - MessageID, 227
- gdcmm::network::CEchoRSP, 227
 - ConstructPDVByDataSet, 228
- gdcmm::network::CFind, 228
- gdcmm::network::CFindCancelRQ, 229
 - ConstructPDVByDataSet, 230
- gdcmm::network::CFindRQ, 230
 - ConstructPDV, 231
- gdcmm::network::CFindRSP, 231
 - ConstructPDVByDataSet, 232
- gdcmm::network::CMoveCancelRq, 232
 - ConstructPDVByDataSet, 233
- gdcmm::network::CMoveRQ, 234
 - ConstructPDV, 235
- gdcmm::network::CMoveRSP, 235
 - ConstructPDVByDataSet, 236
- gdcmm::network::CStoreRQ, 267
 - ConstructPDV, 268
- gdcmm::network::CStoreRSP, 268
 - ConstructPDV, 269
- gdcmm::network::CompositeMessageFactory, 245
 - ConstructCEchoRQ, 246
 - ConstructCFindRQ, 246
 - ConstructCMoveRQ, 246
 - ConstructCStoreRQ, 246
 - ConstructCStoreRSP, 246
- gdcmm::network::DIMSE, 314
 - CommandTypes, 315
- gdcmm::network::ImplementationClassUIDSub, 436
 - ImplementationClassUIDSub, 436
 - Print, 436
 - Read, 436
 - Size, 436
 - Write, 436
- gdcmm::network::ImplementationUIDSub, 436
 - ImplementationUIDSub, 437
 - Write, 437
- gdcmm::network::ImplementationVersionNameSub, 437
 - ImplementationVersionNameSub, 437
 - Print, 437
 - Read, 437
 - Size, 437
 - Write, 437
- gdcmm::network::MaximumLengthSub, 483
 - GetMaximumLength, 484
 - MaximumLengthSub, 484
 - Print, 484
 - Read, 484
 - SetMaximumLength, 484
 - Size, 484
 - Write, 484
- gdcmm::network::PDUFactory, 537
 - ConstructAbortPDU, 538
 - ConstructPDU, 538
 - ConstructReleasePDU, 538
 - CreateCEchoPDU, 538
 - CreateCFindPDU, 538
 - CreateCMovePDU, 538
 - CreateCStoreRQPDU, 538

- CreateCStoreRSPDU, 538
- DetermineEventByPDU, 538
- GetPDVs, 538
- gdcmm::network::PDataTFPDU, 529
 - AddPresentationDataValue, 531
 - GetNumberOfPresentationDataValues, 531
 - GetPresentationDataValue, 531
 - IsLastFragment, 531
 - PDataTFPDU, 531
 - Print, 531
 - Read, 531
 - ReadInto, 531
 - Size, 531
 - SizeType, 531
 - Write, 531
- gdcmm::network::PresentationContextAC, 567
 - GetPresentationContextID, 568
 - GetReason, 568
 - GetTransferSyntax, 568
 - PresentationContextAC, 568
 - Print, 568
 - Read, 568
 - SetPresentationContextID, 568
 - SetReason, 568
 - SetTransferSyntax, 568
 - Size, 568
 - Write, 568
- gdcmm::network::PresentationContextRQ, 570
 - AddTransferSyntax, 571
 - GetAbstractSyntax, 571, 572
 - GetNumberOfTransferSyntaxes, 572
 - GetPresentationContextID, 572
 - GetTransferSyntax, 572
 - GetTransferSyntaxes, 572
 - operator==, 572
 - PresentationContextRQ, 571
 - Print, 572
 - Read, 572
 - SetAbstractSyntax, 572
 - SetPresentationContextID, 572
 - Size, 572
 - SizeType, 571
 - Write, 572
- gdcmm::network::PresentationDataValue, 572
 - ConcatenatePDVBlobs, 573
 - GetBlob, 573
 - GetIsCommand, 573
 - GetIsLastFragment, 573
 - GetMessageHeader, 573
 - GetPresentationContextID, 573
 - PresentationDataValue, 573
 - Print, 573
 - Read, 573
 - ReadInto, 573
 - SetBlob, 573
 - SetCommand, 573
 - SetDataSet, 574
 - SetLastFragment, 574
 - SetMessageHeader, 574
 - SetPresentationContextID, 574
 - Size, 574
 - Write, 574
- gdcmm::network::RoleSelectionSub, 612
 - Print, 613
 - Read, 613
 - RoleSelectionSub, 613
 - SetTuple, 613
 - Size, 613
 - Write, 613
- gdcmm::network::SOPClassExtendedNegotiationSub, 659
 - Print, 660
 - Read, 660
 - SOPClassExtendedNegotiationSub, 660
 - SetTuple, 660
 - Size, 660
 - Write, 660
- gdcmm::network::ServiceClassApplicationInformation, 645
 - Print, 645
 - Read, 645
 - ServiceClassApplicationInformation, 645
 - SetTuple, 645
 - Size, 645
 - Write, 646
- gdcmm::network::TableRow, 711
 - ~TableRow, 712
 - TableRow, 712
 - transitions, 712
- gdcmm::network::TransferSyntaxSub, 731
 - GetName, 731
 - operator==, 731
 - Print, 731
 - Read, 731
 - SetName, 731
 - SetNameFromUID, 731
 - Size, 731
 - TransferSyntaxSub, 731
 - Write, 732
- gdcmm::network::Transition, 732
 - ~Transition, 733
 - mAction, 733
 - mEnd, 733
 - MakeNew, 733
 - Transition, 732, 733
- gdcmm::network::ULAction, 756
 - ~ULAction, 758
 - PerformAction, 758
 - ULAction, 758
- gdcmm::network::ULActionAA1, 759

PerformAction, 759
 gdcmm::network::ULActionAA2, 760
 PerformAction, 760
 gdcmm::network::ULActionAA3, 761
 PerformAction, 762
 gdcmm::network::ULActionAA4, 762
 PerformAction, 763
 gdcmm::network::ULActionAA5, 763
 PerformAction, 764
 gdcmm::network::ULActionAA6, 764
 PerformAction, 765
 gdcmm::network::ULActionAA7, 766
 PerformAction, 766
 gdcmm::network::ULActionAA8, 767
 PerformAction, 767
 gdcmm::network::ULActionAE1, 768
 PerformAction, 769
 gdcmm::network::ULActionAE2, 769
 PerformAction, 770
 gdcmm::network::ULActionAE3, 770
 PerformAction, 771
 gdcmm::network::ULActionAE4, 771
 PerformAction, 772
 gdcmm::network::ULActionAE5, 773
 PerformAction, 773
 gdcmm::network::ULActionAE6, 774
 PerformAction, 774
 gdcmm::network::ULActionAE7, 775
 PerformAction, 776
 gdcmm::network::ULActionAE8, 776
 PerformAction, 777
 gdcmm::network::ULActionAR1, 777
 PerformAction, 778
 gdcmm::network::ULActionAR10, 778
 PerformAction, 779
 gdcmm::network::ULActionAR2, 780
 PerformAction, 780
 gdcmm::network::ULActionAR3, 781
 PerformAction, 781
 gdcmm::network::ULActionAR4, 782
 PerformAction, 783
 gdcmm::network::ULActionAR5, 783
 PerformAction, 784
 gdcmm::network::ULActionAR6, 784
 PerformAction, 785
 gdcmm::network::ULActionAR7, 785
 PerformAction, 786
 gdcmm::network::ULActionAR8, 787
 PerformAction, 787
 gdcmm::network::ULActionAR9, 788
 PerformAction, 788
 gdcmm::network::ULActionDT1, 789
 PerformAction, 790
 gdcmm::network::ULActionDT2, 790

PerformAction, 791
 gdcmm::network::ULBasicCallback, 791
 ~ULBasicCallback, 792
 GetDataSets, 792
 GetResponses, 792
 HandleDataSet, 792
 HandleResponse, 792
 ULBasicCallback, 792
 gdcmm::network::ULConnection, 793
 ~ULConnection, 794
 AddAcceptedPresentationContext, 794
 FindContext, 794
 GetAcceptedPresentationContexts, 794
 GetConnectionInfo, 794
 GetMaxPDUSize, 794
 GetPresentationContextACByID, 794
 GetPresentationContextIDFromPresentationContext, 794
 GetPresentationContextRQByID, 794
 GetPresentationContexts, 794
 GetProtocol, 794
 GetState, 794
 GetTimer, 794
 InitializeConnection, 795
 InitializeIncomingConnection, 795
 SetMaxPDUSize, 795
 SetPresentationContexts, 795
 SetState, 795
 StopProtocol, 795
 ULConnection, 794
 gdcmm::network::ULConnectionCallback, 795
 ~ULConnectionCallback, 796
 DataSetHandled, 796
 DataSetHandles, 796
 HandleDataSet, 796
 HandleResponse, 796
 ResetHandledDataSet, 796
 ULConnectionCallback, 796
 gdcmm::network::ULConnectionInfo, 797
 GetCalledAETitle, 797
 GetCalledComputerName, 797
 GetCalledIPAddress, 797
 GetCalledIPPort, 797
 GetCallingAETitle, 797
 GetMaxPDULength, 797
 Initialize, 797
 SetMaxPDULength, 797
 ULConnectionInfo, 797
 gdcmm::network::ULConnectionManager, 798
 ~ULConnectionManager, 800
 BreakConnection, 800
 BreakConnectionNow, 800
 EstablishConnection, 800
 EstablishConnectionMove, 800

- SendEcho, [800](#)
- SendFind, [800](#)
- SendMove, [800](#)
- SendStore, [800](#)
- ULConnectionManager, [800](#)
- gdcmm::network::ULEvent, [801](#)
 - ~ULEvent, [801](#)
 - GetEvent, [801](#)
 - GetPDUs, [801](#)
 - SetEvent, [801](#)
 - SetPDU, [801](#)
 - ULEvent, [801](#)
- gdcmm::network::ULTransitionTable, [802](#)
 - HandleEvent, [802](#)
 - PrintTable, [802](#)
 - ULTransitionTable, [802](#)
- gdcmm::network::ULWritingCallback, [802](#)
 - ~ULWritingCallback, [804](#)
 - HandleDataSet, [804](#)
 - HandleResponse, [804](#)
 - SetDirectory, [804](#)
 - ULWritingCallback, [803](#)
- gdcmm::network::UserInformation, [812](#)
 - ~UserInformation, [812](#)
 - AddRoleSelectionSub, [812](#)
 - AddSOPClassExtendedNegotiationSub, [812](#)
 - GetMaximumLengthSub, [812](#)
 - operator=, [812](#)
 - Print, [812](#)
 - Read, [812](#)
 - Size, [812](#)
 - UserInformation, [812](#)
 - Write, [813](#)
- gdcmm::static_assert_test< x >, [670](#)
- gdcmm::terminal, [130](#)
 - Attribute, [131](#)
 - Color, [131](#)
 - Mode, [131](#)
 - setAttribute, [131](#)
 - setbgcolor, [131](#)
 - setfgcolor, [131](#)
 - setmode, [131](#)
- gdcmAAbortPDU.h, [893](#)
- gdcmAAssociateACPDU.h, [894](#)
- gdcmAAssociateRJPDU.h, [894](#)
- gdcmAAssociateRQPDU.h, [895](#)
- gdcmARTIMTimer.h, [903](#)
- gdcmAReleaseRPPDU.h, [901](#)
- gdcmAReleaseRQPDU.h, [902](#)
- gdcmASN1.h, [904](#)
- gdcmAbstractSyntax.h, [896](#)
- gdcmAnonymizeEvent.h, [897](#)
- gdcmAnonymizer.h, [899](#)
- gdcmApplicationContext.h, [899](#)
- gdcmApplicationEntity.h, [900](#)
- gdcmAssertAlwaysMacro
 - gdcmTrace.h, [1118](#)
- gdcmAssertMacro
 - gdcmTrace.h, [1118](#)
- gdcmAsynchronousOperationsWindowSub.h, [905](#)
- gdcmAttribute.h, [905](#)
- gdcmAudioCodec.h, [907](#)
- gdcmBase64.h, [907](#)
- gdcmBaseCompositeMessage.h, [908](#)
- gdcmBasePDU.h, [909](#)
- gdcmBaseRootQuery.h, [910](#)
- gdcmBasicOffsetTable.h, [912](#)
- gdcmBitmap.h, [913](#)
- gdcmBitmapToBitmapFilter.h, [914](#)
- gdcmBoxRegion.h, [915](#)
- gdcmByteBuffer.h, [915](#)
- gdcmByteSwap.h, [916](#)
- gdcmByteSwapFilter.h, [917](#)
- gdcmByteValue.h, [918](#)
- gdcmCEchoMessages.h, [919](#)
- gdcmCFindMessages.h, [919](#)
- gdcmCMoveMessages.h, [920](#)
- gdcmCP246ExplicitDataElement.h, [929](#)
- gdcmCSAElement.h, [930](#)
- gdcmCSAHeader.h, [931](#)
- gdcmCSAHeaderDict.h, [932](#)
- gdcmCSAHeaderDictEntry.h, [933](#)
- gdcmCStoreMessages.h, [934](#)
- gdcmCodeString.h, [923](#)
- gdcmCodec.h, [921](#)
- gdcmCoder.h, [922](#)
- gdcmCommand.h, [924](#)
- gdcmCommandDataSet.h, [926](#)
- gdcmCompositeMessageFactory.h, [926](#)
- gdcmCompositeNetworkFunctions.h, [927](#)
- gdcmConstCharWrapper.h, [928](#)
- gdcmCryptographicMessageSyntax.h, [929](#)
- gdcmCurve.h, [935](#)
- gdcmDICOMDIR.h, [945](#)
- gdcmDICOMDIRGenerator.h, [946](#)
- gdcmDIMSE.h, [953](#)
- gdcmDataElement.h, [936](#)
- gdcmDataEvent.h, [938](#)
- gdcmDataSet.h, [939](#)
- gdcmDataSetEvent.h, [940](#)
- gdcmDataSetHelper.h, [940](#)
- gdcmDebugMacro
 - gdcmTrace.h, [1119](#)
- gdcmDecoder.h, [941](#)
- gdcmDefinedTerms.h, [942](#)
- gdcmDeflateStream.h, [943](#)
- gdcmDefs.h, [943](#)
- gdcmDeltaEncodingCodec.h, [945](#)

gdcmDict.h, 947
gdcmDictConverter.h, 949
gdcmDictEntry.h, 949
gdcmDictPrinter.h, 951
gdcmDicts.h, 951
gdcmDirectionCosines.h, 953
gdcmDirectory.h, 954
gdcmDirectoryHelper.h, 955
gdcmDummyValueGenerator.h, 956
gdcmDumper.h, 957
gdcmElement.h, 957
gdcmEncapsulatedDocument.h, 959
gdcmEnumeratedValues.h, 959
gdcmErrorMacro
 gdcmTrace.h, 1119
gdcmEvent.h, 960
 gdcmEventMacro, 961
gdcmEventMacro
 gdcmEvent.h, 961
gdcmException.h, 962
gdcmExplicitDataElement.h, 963
gdcmExplicitImplicitDataElement.h, 963
gdcmFiducials.h, 964
gdcmFile.h, 965
gdcmFileAnonymizer.h, 966
gdcmFileDerivation.h, 966
gdcmFileExplicitFilter.h, 967
gdcmFileMetaInformation.h, 968
gdcmFileSet.h, 970
gdcmFilename.h, 969
gdcmFilenameGenerator.h, 969
gdcmFindPatientRootQuery.h, 972
gdcmFindStudyRootQuery.h, 973
gdcmFragment.h, 973
gdcmGlobal.h, 975
gdcmGroupDict.h, 976
gdcmIOD.h, 996
gdcmIODEntry.h, 998
gdcmIODs.h, 1001
gdcmIPPSorter.h, 1002
gdcmIconImage.h, 976
gdcmIconImageFilter.h, 977
gdcmIconImageGenerator.h, 978
gdcmImage.h, 979
gdcmImageApplyLookupTable.h, 980
gdcmImageChangePhotometricInterpretation.h, 981
gdcmImageChangePlanarConfiguration.h, 982
gdcmImageChangeTransferSyntax.h, 983
gdcmImageCodec.h, 984
gdcmImageConverter.h, 985
gdcmImageFragmentSplitter.h, 986
gdcmImageHelper.h, 987
gdcmImageReader.h, 988
gdcmImageRegionReader.h, 989
gdcmImageToImageFilter.h, 990
gdcmImageWriter.h, 991
gdcmImplementationClassUIDSub.h, 992
gdcmImplementationUIDSub.h, 993
gdcmImplementationVersionNameSub.h, 994
gdcmImplicitDataElement.h, 996
gdcmItem.h, 1003
gdcmJPEG12Codec.h, 1005
gdcmJPEG16Codec.h, 1005
gdcmJPEG2000Codec.h, 1006
gdcmJPEG8Codec.h, 1007
gdcmJPEGCodec.h, 1008
gdcmJPEGLSCodec.h, 1009
gdcmKAKADUCodec.h, 1010
gdcmLO.h, 1012
gdcmLegacyMacro.h, 1011
 GDCM_LEGACY, 1012
 GDCM_LEGACY_BODY, 1012
gdcmLookupTable.h, 1013
gdcmMD5.h, 1021
gdcmMacro.h, 1014
gdcmMacroEntry.h, 1016
 GDCMMACROENTRY_H, 1018
gdcmMacros.h, 1018
gdcmMaximumLengthSub.h, 1020
gdcmMediaStorage.h, 1022
gdcmMeshPrimitive.h, 1023
gdcmModule.h, 1025
gdcmModuleEntry.h, 1026
gdcmModules.h, 1028
gdcmMovePatientRootQuery.h, 1029
gdcmMoveStudyRootQuery.h, 1030
gdcmNestedModuleEntries.h, 1031
gdcmNetworkEvents.h, 1033
gdcmNetworkStateID.h, 1034
gdcmObject.h, 1035
gdcmOrientation.h, 1036
gdcmOverlay.h, 1036
gdcmPDBelement.h, 1041
gdcmPDBHeader.h, 1043
gdcmPDFCodec.h, 1043
gdcmPDUFactory.h, 1044
gdcmPDataTFPDU.h, 1040
gdcmPGXCodec.h, 1046
gdcmPNMCodec.h, 1052
gdcmPVRGCodec.h, 1062
gdcmParseException.h, 1037
gdcmParser.h, 1039
gdcmPatient.h, 1039
gdcmPersonName.h, 1045
gdcmPhotometricInterpretation.h, 1046
gdcmPixelFormat.h, 1047
gdcmPixmap.h, 1048
gdcmPixmapReader.h, 1049

- gdcmPixmapToPixmapFilter.h, 1051
- gdcmPixmapWriter.h, 1051
- gdcmPreamble.h, 1053
- gdcmPresentationContext.h, 1054
- gdcmPresentationContextAC.h, 1055
- gdcmPresentationContextGenerator.h, 1057
- gdcmPresentationContextRQ.h, 1057
- gdcmPresentationDataValue.h, 1058
- gdcmPrinter.h, 1059
- gdcmPrivateTag.h, 1060
- gdcmProgressEvent.h, 1062
- gdcmPythonFilter.h, 1063
- gdcmQueryBase.h, 1064
- gdcmQueryFactory.h, 1065
- gdcmQueryImage.h, 1066
- gdcmQueryPatient.h, 1067
- gdcmQuerySeries.h, 1068
- gdcmQueryStudy.h, 1069
- gdcmRAWCodec.h, 1070
- gdcmRLECodec.h, 1074
- gdcmReader.h, 1071
- gdcmRegion.h, 1072
- gdcmRescaler.h, 1073
- gdcmRoleSelectionSub.h, 1074
- gdcmSHA1.h, 1087
- gdcmSOPClassExtendedNegotiationSub.h, 1090
- gdcmSOPClassUIDToIOD.h, 1091
- gdcmScanner.h, 1075
- gdcmSegment.h, 1076
- gdcmSegmentHelper.h, 1078
- gdcmSegmentReader.h, 1080
- gdcmSegmentWriter.h, 1081
- gdcmSegmentedPaletteColorLookupTable.h, 1078
- gdcmSequenceOfFragments.h, 1082
- gdcmSequenceOfItems.h, 1083
- gdcmSerieHelper.h, 1083
- gdcmSeries.h, 1085
- gdcmServiceClassApplicationInformation.h, 1086
- gdcmServiceClassUser.h, 1087
- gdcmSimpleSubjectWatcher.h, 1088
- gdcmSmartPointer.h, 1089
- gdcmSorter.h, 1092
- gdcmSpacing.h, 1094
- gdcmSpectroscopy.h, 1094
- gdcmSplitMosaicFilter.h, 1095
- gdcmStaticAssert.h, 1096
 - GDCM_DO_JOIN, 1096
 - GDCM_DO_JOIN2, 1096
 - GDCM_JOIN, 1096
- gdcmStreamImageReader.h, 1097
- gdcmStreamImageWriter.h, 1097
- gdcmString.h, 1098
- gdcmStringFilter.h, 1099
- gdcmStudy.h, 1100
- gdcmSubject.h, 1101
- gdcmSurface.h, 1102
- gdcmSurfaceHelper.h, 1103
- gdcmSurfaceReader.h, 1104
- gdcmSurfaceWriter.h, 1105
- gdcmSwapCode.h, 1106
- gdcmSwapper.h, 1107
- gdcmSystem.h, 1108
- gdcmTable.h, 1109
- gdcmTableEntry.h, 1110
- gdcmTableReader.h, 1111
- gdcmTag.h, 1113
- gdcmTagPath.h, 1114
- gdcmTagToVR.h, 1114
- gdcmTerminal.h, 1115
- gdcmTestDriver.h, 1116
- gdcmTesting.h, 1116
- gdcmTrace.h, 1117
 - GDCM_FUNCTION, 1118
 - gdcmAssertAlwaysMacro, 1118
 - gdcmAssertMacro, 1118
 - gdcmDebugMacro, 1119
 - gdcmErrorMacro, 1119
 - gdcmWarningMacro, 1119
- gdcmTransferSyntax.h, 1120
- gdcmTransferSyntaxSub.h, 1121
- gdcmType.h, 1122
- gdcmTypes.h, 1124
- gdcmUIDGenerator.h, 1124
- gdcmUIDs.h, 1125
- gdcmULAction.h, 1126
- gdcmULActionAA.h, 1127
- gdcmULActionAE.h, 1128
- gdcmULActionAR.h, 1129
- gdcmULActionDT.h, 1129
- gdcmULBasicCallback.h, 1130
- gdcmULConnection.h, 1131
- gdcmULConnectionCallback.h, 1132
- gdcmULConnectionInfo.h, 1133
- gdcmULConnectionManager.h, 1134
- gdcmULEvent.h, 1135
- gdcmULTransitionTable.h, 1136
- gdcmULWritingCallback.h, 1137
- gdcmUNExplicitDataElement.h, 1137
- gdcmUNExplicitImplicitDataElement.h, 1138
- gdcmUnpacker12Bits.h, 1139
- gdcmUsage.h, 1139
- gdcmUserInformation.h, 1142
- gdcmVL.h, 1146
- gdcmVM.h, 1147
 - TYPETOLENGTH, 1148
- gdcmVR.h, 1148
 - TYPETOENCODING, 1150
 - VRTypeTemplateCase, 1150

- gdcmVR16ExplicitDataElement.h, 1151
- gdcmValidate.h, 1143
- gdcmValue.h, 1143
- gdcmValueIO.h, 1144
- gdcmVersion.h, 1145
- gdcmWarningMacro
 - gdcmTrace.h, 1119
- gdcmWaveform.h, 1151
- gdcmWin32.h, 1152
 - GDCM_EXPORT, 1152
- gdcmWriter.h, 1152
- gdcmXMLDictReader.h, 1153
- gdcmXMLPrivateDictReader.h, 1154
- gdcmanon.man, 897
- gdcmconv.man, 928
- gdcmdiff.man, 952
- gdcmdump.man, 956
- gdcmgendir.man, 975
- gdcmimg.man, 992
- gdcminfo.man, 996
- gdcmpdf.man, 1043
- gdcmraw.man, 1070
- gdcmscanner.man, 1076
- gdcm SCU.man, 1076
- gdcm tar.man, 1114
- gdcmviewer.man, 1146
- GeneralECGWaveformStorage
 - gdcm::MediaStorage, 489
 - gdcm::UIDs, 745
- GeneralElectricMagneticResonanceImageStorage
 - gdcm::MediaStorage, 489
- GeneralPurposePerformedProcedureStepSOPClass
 - gdcm::UIDs, 747
- GeneralPurposeScheduledProcedureStepSOPClass
 - gdcm::UIDs, 747
- GeneralPurposeWorklistInformationModelFIND
 - gdcm::UIDs, 747
- GeneralPurposeWorklistManagementMetaSOPClass
 - gdcm::UIDs, 747
- GeneralRelevantPatientInformationQuery
 - gdcm::UIDs, 747
- Generate
 - gdcm::DICOMDIRGenerator, 302
 - gdcm::DummyValueGenerator, 322
 - gdcm::FilenameGenerator, 376
 - gdcm::IconImageGenerator, 394
 - gdcm::UIDGenerator, 737
- GenerateFromFilenames
 - gdcm::PresentationContextGenerator, 570
- GenerateFromUID
 - gdcm::PresentationContextGenerator, 570
- GenerateUUID
 - gdcm::UIDGenerator, 737
- Get
 - gdcm::ByteBuffer, 219
- GetAETitle
 - gdcm::ServiceClassUser, 648
- GetALGOType
 - gdcm::Segment, 623
- GetALGOTypeString
 - gdcm::Segment, 623
- GetAbbreviation
 - gdcm::GroupDict, 390
- GetAbstractSyntax
 - gdcm::network::PresentationContextRQ, 571, 572
 - gdcm::PresentationContext, 566
- GetAbstractSyntaxUID
 - gdcm::BaseRootQuery, 199
 - gdcm::FindPatientRootQuery, 381
 - gdcm::FindStudyRootQuery, 384
 - gdcm::MovePatientRootQuery, 508
 - gdcm::MoveStudyRootQuery, 510
- GetAcceptedPresentationContexts
 - gdcm::network::ULConnection, 794
- GetAlgorithmFamily
 - gdcm::Surface, 691
- GetAlgorithmName
 - gdcm::Surface, 691
- GetAlgorithmVersion
 - gdcm::Surface, 691
- GetAllFilenamesFromTagToValue
 - gdcm::Scanner, 618
- GetAllRequiredTags
 - gdcm::QueryBase, 587
- GetAllTags
 - gdcm::QueryBase, 587
- GetAnatomicRegion
 - gdcm::Segment, 623
- GetAsDataElement
 - gdcm::Attribute, 165
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 172
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 179
 - gdcm::Element, 327
 - gdcm::Element< TVR, VM::VM1_n >, 331
 - gdcm::network::AbstractSyntax, 145
- GetAsPoints
 - gdcm::Curve, 271
- GetAsString
 - gdcm::CodeString, 241
- GetAxisOfRotation
 - gdcm::Surface, 691
- GetBasicApplicationLevelConfidentialityProfileAttributes
 - gdcm::Anonymizer, 151
- GetBitPosition
 - gdcm::Overlay, 522
- GetBitSample

- gdcm::LookupTable, [479](#)
- GetBitsAllocated
 - gdcm::Overlay, [522](#)
 - gdcm::PixelFormat, [546](#)
- GetBitsStored
 - gdcm::PixelFormat, [547](#)
- GetBlob
 - gdcm::network::PresentationDataValue, [573](#)
- GetBuffer
 - gdcm::Bitmap, [208](#)
 - gdcm::ByteValue, [223](#)
 - gdcm::Overlay, [522](#)
 - gdcm::Parser, [528](#)
 - gdcm::SequenceOfFragments, [634](#)
- GetBuffer2
 - gdcm::Bitmap, [208](#)
- GetBufferAsRGBA
 - gdcm::LookupTable, [479](#)
- GetBufferLength
 - gdcm::Bitmap, [208](#)
 - gdcm::JPEGLSCodec, [470](#)
 - gdcm::PNMCodec, [563](#)
 - gdcm::RLECodec, [612](#)
- GetBuildVersion
 - gdcm::Version, [817](#)
- GetByteValue
 - gdcm::CSAElement, [255](#)
 - gdcm::DataElement, [276](#)
- GetCSADataInfo
 - gdcm::CSAHeader, [260](#)
- GetCSAEEnd
 - gdcm::CSAHeader, [261](#)
- GetCSAElementByName
 - gdcm::CSAHeader, [261](#)
- GetCSAHeaderDict
 - gdcm::Dicts, [313](#)
- GetCSAHeaderDictEntry
 - gdcm::CSAHeaderDict, [263](#)
- GetCSAImageHeaderInfoTag
 - gdcm::CSAHeader, [261](#)
- GetCSASeriesHeaderInfoTag
 - gdcm::CSAHeader, [261](#)
- GetCTImageSeriesUIDs
 - gdcm::DirectoryHelper, [321](#)
- GetCWD
 - gdcm::System, [706](#)
- GetCalledAETitle
 - gdcm::network::AAssociateRQPDU, [142](#)
 - gdcm::network::ULConnectionInfo, [797](#)
 - gdcm::ServiceClassUser, [648](#)
- GetCalledComputerName
 - gdcm::network::ULConnectionInfo, [797](#)
- GetCalledIPAddress
 - gdcm::network::ULConnectionInfo, [797](#)
- GetCalledIPPort
 - gdcm::network::ULConnectionInfo, [797](#)
- GetCallingAETitle
 - gdcm::network::AAssociateRQPDU, [142](#)
 - gdcm::network::ULConnectionInfo, [797](#)
- GetCenterOfRotation
 - gdcm::Surface, [691](#)
- GetCharacterFromCurrentLocale
 - gdcm::QueryFactory, [588](#)
- GetCipherType
 - gdcm::CryptographicMessageSyntax, [253](#)
- GetColorLevel
 - vtkImageColorViewer, [862](#)
- GetColorWindow
 - vtkImageColorViewer, [862](#)
- GetColumns
 - gdcm::Bitmap, [209](#)
 - gdcm::Overlay, [522](#)
- GetCommand
 - gdcm::Subject, [686](#)
- GetConnectionInfo
 - gdcm::network::ULConnection, [794](#)
- GetConstructorString
 - gdcm::Dicts, [313](#)
- GetContourReferencedFrameOfReferenceClassUID
 - vtkRTStructSetProperties, [881](#)
- GetContourReferencedFrameOfReferenceInstanceUID
 - vtkRTStructSetProperties, [881](#)
- GetCryptographicMessageSyntax
 - gdcm::Anonymizer, [151](#)
- GetCurrentByteIndex
 - gdcm::Parser, [528](#)
- GetCurrentDateTime
 - gdcm::System, [705](#)
- GetCurrentModuleFileName
 - gdcm::System, [705](#)
- GetCurrentProcessFileName
 - gdcm::System, [706](#)
- GetCurrentResourcesDirectory
 - gdcm::System, [706](#)
- GetCurve
 - gdcm::Pixmap, [551](#), [552](#)
- GetCurveDataDescriptor
 - gdcm::Curve, [271](#)
- GetDEEnd
 - gdcm::DataSet, [289](#)
- GetDES
 - gdcm::DataSet, [289](#)
- GetData
 - gdcm::DataEvent, [284](#)
- GetDataElement
 - gdcm::Bitmap, [209](#)
 - gdcm::DataSet, [288](#), [289](#)
 - gdcm::Item, [452](#)

- GetDataExtraRoot
 - gdcm::Testing, [722](#)
- GetDataLength
 - gdcm::DataEvent, [284](#)
- GetDataRoot
 - gdcm::Testing, [722](#)
- GetDataSet
 - gdcm::CSAHeader, [261](#)
 - gdcm::DataSetEvent, [293](#)
 - gdcm::File, [359](#)
- GetDataSetTransferSyntax
 - gdcm::FileMetaInformation, [370](#)
- GetDataSets
 - gdcm::network::ULBasicCallback, [792](#)
- GetDataValueRepresentation
 - gdcm::Curve, [271](#)
- GetDebugFlag
 - gdcm::Trace, [725](#)
- GetDebugStream
 - gdcm::Trace, [725](#)
- GetDecodeLength
 - gdcm::Base64, [192](#)
- GetDefaultTransferSyntax
 - gdcm::PresentationContextGenerator, [570](#)
- GetDefs
 - gdcm::Global, [388](#)
 - gdcm::TableReader, [710](#)
- GetDescription
 - gdcm::CSAHeaderDictEntry, [265](#)
 - gdcm::Exception, [351](#)
 - gdcm::ModuleEntry, [505](#)
 - gdcm::Overlay, [522](#)
- GetDescriptiveName
 - vtkGDCMImageReader, [836](#)
 - vtkGDCMImageWriter, [841](#)
- GetDict
 - gdcm::XMLDictReader, [890](#)
- GetDictEntry
 - gdcm::Dict, [304](#)
 - gdcm::Dicts, [313](#), [314](#)
 - gdcm::PrivateDict, [578](#)
- GetDictEntryByKeyword
 - gdcm::Dict, [304](#)
- GetDictEntryByName
 - gdcm::Dict, [304](#)
- GetDictName
 - gdcm::DictConverter, [307](#)
- GetDictVM
 - gdcm::Attribute, [166](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [179](#)
- GetDictVR
 - gdcm::Attribute, [166](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [179](#)
- GetDicts
 - gdcm::Global, [388](#)
- GetDimension
 - gdcm::Bitmap, [209](#)
- GetDimensions
 - gdcm::Bitmap, [209](#)
 - gdcm::Curve, [271](#)
 - gdcm::ImageCodec, [416](#)
- GetDimensionsValue
 - gdcm::ImageHelper, [422](#)
- GetDimensionsValueForResolution
 - gdcm::StreamImageReader, [672](#)
- GetDirectionCosines
 - gdcm::Image, [398](#)
- GetDirectionCosinesFromDataSet
 - gdcm::ImageHelper, [423](#)
- GetDirectionCosinesTolerance
 - gdcm::IPPSorter, [448](#)
- GetDirectionCosinesValue
 - gdcm::ImageHelper, [423](#)
- GetDirectories
 - gdcm::Directory, [319](#)
- GetElapsedTime
 - gdcm::network::ARTIMTimer, [161](#)
- GetElement
 - gdcm::Tag, [714](#)
- GetElementTag
 - gdcm::Tag, [715](#)
- GetEncodeLength
 - gdcm::Base64, [192](#)
- GetErrorCode
 - gdcm::Parser, [528](#)
- GetErrorFlag
 - gdcm::Trace, [725](#)
- GetErrorStream
 - gdcm::Trace, [725](#)
- GetErrorString
 - gdcm::Parser, [528](#)
- GetEvent
 - gdcm::network::ULEvent, [801](#)
- GetEventName
 - gdcm::AnonymizeEvent, [147](#)
 - gdcm::DataEvent, [284](#)
 - gdcm::DataSetEvent, [293](#)
 - gdcm::Event, [349](#)
 - gdcm::ProgressEvent, [582](#)
- GetExtension
 - gdcm::Filename, [374](#)
- GetFile

- gdcm::Anonymizer, [151](#)
- gdcm::DICOMDIRGenerator, [302](#)
- gdcm::FileDerivation, [365](#)
- gdcm::FileExplicitFilter, [367](#)
- gdcm::IconImageFilter, [392](#)
- gdcm::PythonFilter, [585](#)
- gdcm::Reader, [602](#), [603](#)
- gdcm::SplitMosaicFilter, [668](#)
- gdcm::StreamImageReader, [672](#)
- gdcm::StringFilter, [683](#)
- gdcm::Writer, [887](#)
- vtkGDCMMedicalImageProperties, [845](#)
- GetFileExtensions
 - vtkGDCMImageReader, [836](#)
 - vtkGDCMImageWriter, [841](#)
- GetFileMetaInformationVersion
 - gdcm::FileMetaInformation, [370](#)
- GetFileName
 - gdcm::Filename, [374](#)
 - gdcm::Testing, [722](#)
 - vtkGDCMImageWriter, [841](#)
 - vtkGDCMThreadedImageReader2, [857](#)
- GetFileNames
 - gdcm::Testing, [722](#)
- GetFilename
 - gdcm::FilenameGenerator, [376](#)
 - gdcm::TableReader, [710](#)
- GetFilenameFromTagToValue
 - gdcm::Scanner, [618](#)
- GetFilenames
 - gdcm::Directory, [319](#)
 - gdcm::FilenameGenerator, [376](#)
 - gdcm::Scanner, [618](#)
 - gdcm::Sorter, [664](#)
- GetFilenamesFromSeriesUIDs
 - gdcm::DirectoryHelper, [321](#)
- GetFiles
 - gdcm::FileSet, [378](#)
- GetFiniteVolume
 - gdcm::Surface, [691](#)
- GetFirstSingleSerieUIDFileSet
 - gdcm::SerieHelper, [644](#)
- GetForcePixelSpacing
 - gdcm::ImageHelper, [423](#)
- GetForceRescaleInterceptSlope
 - gdcm::ImageHelper, [423](#)
- GetFormat
 - gdcm::CSAHeader, [261](#)
- GetFragBuffer
 - gdcm::SequenceOfFragments, [634](#)
- GetFragment
 - gdcm::SequenceOfFragments, [634](#)
- GetFragmentSizeMax
 - gdcm::ImageFragmentSplitter, [421](#)
- GetFrameOfReference
 - gdcm::DirectoryHelper, [321](#)
- GetFullLength
 - gdcm::FileMetaInformation, [371](#)
- GetGDCMDataRoot
 - vtkGDCMTesting, [853](#)
- GetGDCMImplementationClassUID
 - gdcm::FileMetaInformation, [371](#)
- GetGDCMImplementationVersionName
 - gdcm::FileMetaInformation, [371](#)
- GetGDCMSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, [371](#)
- GetGDCMUID
 - gdcm::UIDGenerator, [737](#)
- GetGroup
 - gdcm::Curve, [271](#)
 - gdcm::Overlay, [522](#)
 - gdcm::Tag, [715](#)
- GetHasExpired
 - gdcm::network::ARTIMTimer, [161](#)
- GetHeader
 - gdcm::File, [359](#)
- GetHeaderInfo
 - gdcm::ImageCodec, [416](#)
 - gdcm::JPEG12Codec, [456](#)
 - gdcm::JPEG16Codec, [458](#)
 - gdcm::JPEG2000Codec, [461](#)
 - gdcm::JPEG8Codec, [463](#)
 - gdcm::JPEGCodec, [467](#)
 - gdcm::JPEGLSCodec, [470](#)
 - gdcm::PGXCodec, [541](#)
 - gdcm::PNMCodec, [563](#)
 - gdcm::RAWCodec, [599](#)
 - gdcm::RLECodec, [612](#)
- GetHierarchicalSearchTags
 - gdcm::QueryBase, [587](#)
 - gdcm::QueryImage, [590](#)
 - gdcm::QueryPatient, [592](#)
 - gdcm::QuerySeries, [594](#)
 - gdcm::QueryStudy, [596](#)
- GetHighBit
 - gdcm::PixelFormat, [547](#)
- GetHostName
 - gdcm::System, [706](#)
- GetIE
 - gdcm::IODEntry, [444](#)
- GetIOD
 - gdcm::IODs, [446](#)
 - gdcm::SOPClassUIDToIOD, [661](#)
- GetIODEntry
 - gdcm::IOD, [442](#)
- GetIODFromFile
 - gdcm::Defs, [297](#)
- GetIODFromSOPClassUID

- gdcm::SOPClassUIDToIOD, 661
- GetIODNameFromMediaStorage
 - gdcm::Defs, 297
- GetIODs
 - gdcm::Defs, 297
- GetIconImage
 - gdcm::IconImageFilter, 392
 - gdcm::IconImageGenerator, 394
 - gdcm::Pixmap, 552
 - vtkGDCMImageReader, 836
- GetImage
 - gdcm::ImageReader, 427
 - gdcm::ImageWriter, 435
 - gdcm::PixmapWriter, 560
 - gdcm::SplitMosaicFilter, 668
- GetImplementationClassUID
 - gdcm::FileMetaInformation, 371
- GetImplementationVersionName
 - gdcm::FileMetaInformation, 371
- GetIndex
 - gdcm::SwapCode, 702
 - gdcm::VM, 823
- GetInput
 - gdcm::ImageToImageFilter, 433
 - gdcm::PixmapToPixmapFilter, 558
 - vtkImageColorViewer, 862
- GetInputFilename
 - gdcm::DictConverter, 307
- GetInstance
 - gdcm::Global, 388
- GetIntercept
 - gdcm::Image, 398
 - gdcm::Rescaler, 608
- GetInterfile
 - gdcm::CSAHeader, 261
- GetInternal
 - gdcm::Preamble, 565
- GetIsCommand
 - gdcm::network::PresentationDataValue, 573
- GetIsLastFragment
 - gdcm::network::PresentationDataValue, 573
- GetItem
 - gdcm::SequenceOfItems, 640
- GetKey
 - gdcm::CSAElement, 256
- GetKeys
 - gdcm::Scanner, 618
- GetKeyword
 - gdcm::DictEntry, 308
- GetKeywordFromTag
 - gdcm::Dict, 304
- GetLUT
 - gdcm::Bitmap, 209
 - gdcm::ImageCodec, 416
 - gdcm::ImageHelper, 423
 - gdcm::LookupTable, 479
- GetLUTDescriptor
 - gdcm::LookupTable, 479
- GetLUTLength
 - gdcm::LookupTable, 479
- GetLabel
 - gdcm::Orientation, 518
- GetLastElement
 - gdcm::ParseException, 526
- GetLastSystemError
 - gdcm::System, 706
- GetLength
 - gdcm::ByteValue, 223
 - gdcm::CP246ExplicitDataElement, 251
 - gdcm::DataElement, 276
 - gdcm::DataSet, 289
 - gdcm::Element, 327
 - gdcm::Element< TVR, VM::VM1_n >, 331
 - gdcm::Element< VR::AS, VM::VM5 >, 339
 - gdcm::ExplicitDataElement, 354
 - gdcm::ExplicitImplicitDataElement, 356
 - gdcm::Fragment, 386
 - gdcm::ImplicitDataElement, 440
 - gdcm::Item, 452
 - gdcm::Preamble, 565
 - gdcm::SequenceOfFragments, 635
 - gdcm::SequenceOfItems, 640
 - gdcm::Tag, 715
 - gdcm::UNExplicitDataElement, 805
 - gdcm::UNExplicitImplicitDataElement, 807
 - gdcm::Value, 815
 - gdcm::VL, 819
 - gdcm::VM, 823
 - gdcm::VR, 827, 828
 - gdcm::VR16ExplicitDataElement, 830
- GetLocaleCharset
 - gdcm::System, 706
- GetLossless
 - gdcm::JPEGCodec, 467
 - gdcm::JPEGLSCodec, 471
- GetLossyFlag
 - gdcm::ImageCodec, 416
- GetLossyFlagFromFile
 - gdcm::Testing, 722
- GetMD5DataImage
 - gdcm::Testing, 722
- GetMD5DataImages
 - gdcm::Testing, 722
- GetMD5FromBrokenFile
 - gdcm::Testing, 722
- GetMD5FromFile
 - gdcm::Testing, 722
- GetMD5MetaImage

- vtkGDCMTesting, [853](#)
- GetMHDMD5FromFile
 - vtkGDCMTesting, [853](#)
- GetMPTType
 - gdcm::MeshPrimitive, [499](#)
- GetMPTTypeString
 - gdcm::MeshPrimitive, [499](#)
- GetMRIImageSeriesUIDs
 - gdcm::DirectoryHelper, [321](#)
- GetMSString
 - gdcm::MediaStorage, [491](#)
- GetMSType
 - gdcm::MediaStorage, [491](#)
- GetMTime
 - vtkImageMapToColors16, [867](#)
- GetMacro
 - gdcm::Macros, [483](#)
- GetMacroEntry
 - gdcm::Macro, [482](#)
- GetMacros
 - gdcm::Defs, [297](#)
- GetMajorAxisFromPatientRelativeDirectionCosine
 - gdcm::Orientation, [518](#)
- GetMajorVersion
 - gdcm::Version, [817](#)
- GetManifold
 - gdcm::Surface, [691](#)
- GetMapping
 - gdcm::Scanner, [619](#)
- GetMappingFromTagToValue
 - gdcm::Scanner, [619](#)
- GetMappings
 - gdcm::Scanner, [619](#)
- GetMax
 - gdcm::PixelFormat, [547](#)
- GetMaxLength
 - gdcm::PersonName, [539](#)
- GetMaxPDULength
 - gdcm::network::ULConnectionInfo, [797](#)
- GetMaxPDUSize
 - gdcm::network::ULConnection, [794](#)
- GetMaximumLength
 - gdcm::network::MaximumLengthSub, [484](#)
- GetMaximumLengthSub
 - gdcm::network::UserInformation, [812](#)
- GetMaximumPointDistance
 - gdcm::Surface, [691](#)
- GetMeanPointDistance
 - gdcm::Surface, [691](#)
- GetMediaStorage
 - gdcm::DataSet, [289](#)
 - gdcm::FileMetaInformation, [371](#)
- GetMediaStorageDataFile
 - gdcm::Testing, [723](#)
- GetMediaStorageDataFiles
 - gdcm::Testing, [723](#)
- GetMediaStorageFromFile
 - gdcm::Testing, [723](#)
- GetMeshPrimitive
 - gdcm::Surface, [691](#)
- GetMessageHeader
 - gdcm::network::PresentationDataValue, [573](#)
- GetMetaInformationTS
 - gdcm::FileMetaInformation, [371](#)
- GetMin
 - gdcm::PixelFormat, [547](#)
- GetMinorVersion
 - gdcm::Version, [817](#)
- GetModality
 - gdcm::MediaStorage, [491](#)
- GetModalityDimension
 - gdcm::MediaStorage, [491](#)
- GetModule
 - gdcm::Modules, [506](#)
- GetModuleEntry
 - gdcm::NestedModuleEntries, [513](#)
- GetModuleEntryInMacros
 - gdcm::Module, [502](#)
- GetModules
 - gdcm::Defs, [297](#)
- GetName
 - gdcm::CSAElement, [256](#)
 - gdcm::CSAHeaderDictEntry, [265](#)
 - gdcm::DictEntry, [308](#)
 - gdcm::Filename, [374](#)
 - gdcm::GroupDict, [390](#)
 - gdcm::IODEntry, [444](#)
 - gdcm::Macro, [482](#)
 - gdcm::Module, [502](#)
 - gdcm::ModuleEntry, [505](#)
 - gdcm::network::AbstractSyntax, [145](#)
 - gdcm::network::ApplicationContext, [155](#)
 - gdcm::network::TransferSyntaxSub, [731](#)
 - gdcm::PDBElement, [533](#)
 - gdcm::QueryBase, [587](#)
 - gdcm::QueryImage, [590](#)
 - gdcm::QueryPatient, [592](#)
 - gdcm::QuerySeries, [594](#)
 - gdcm::QueryStudy, [596](#)
 - gdcm::UIDs, [755](#)
- GetNeedByteSwap
 - gdcm::Bitmap, [209](#)
 - gdcm::ImageCodec, [416](#)
- GetNegotiatedType
 - gdcm::TransferSyntax, [729](#)
- GetNestedDataSet
 - gdcm::Item, [452](#), [453](#)
- GetNextSingleSerieUIDFileSet

- gdcm::SerieHelper, [644](#)
- GetNoOfItems
 - gdcm::CSAElement, [256](#)
- GetNumberOfComponents
 - gdcm::PersonName, [539](#)
- GetNumberOfContourReferencedFrameOfReferences
 - vtkRTStructSetProperties, [881](#)
- GetNumberOfCurves
 - gdcm::Curve, [271](#)
 - gdcm::Pixmap, [552](#)
- GetNumberOfDimensions
 - gdcm::Bitmap, [209](#)
 - gdcm::ImageCodec, [416](#)
- GetNumberOfElementsFromArray
 - gdcm::VM, [823](#)
- GetNumberOfFileNames
 - gdcm::Testing, [723](#)
- GetNumberOfFilenames
 - gdcm::FilenameGenerator, [376](#)
- GetNumberOfFragments
 - gdcm::SequenceOfFragments, [635](#)
- GetNumberOfIODs
 - gdcm::IOD, [442](#)
- GetNumberOfIconImages
 - gdcm::IconImageFilter, [393](#)
- GetNumberOfItems
 - gdcm::SequenceOfItems, [640](#)
- GetNumberOfMD5DataImages
 - gdcm::Testing, [723](#)
- GetNumberOfMD5MetaImages
 - vtkGDCMTesting, [853](#)
- GetNumberOfMSSString
 - gdcm::MediaStorage, [491](#)
- GetNumberOfMSType
 - gdcm::MediaStorage, [491](#)
- GetNumberOfMediaStorageDataFiles
 - gdcm::Testing, [723](#)
- GetNumberOfModality
 - gdcm::MediaStorage, [491](#)
- GetNumberOfModuleEntries
 - gdcm::NestedModuleEntries, [513](#)
- GetNumberOfOverlays
 - gdcm::Pixmap, [552](#)
- GetNumberOfPoints
 - gdcm::Curve, [271](#)
- GetNumberOfPresentationContext
 - gdcm::network::AAssociateRQPDU, [142](#)
- GetNumberOfPresentationContextAC
 - gdcm::network::AAssociateACPDU, [137](#)
- GetNumberOfPresentationDataValues
 - gdcm::network::PDataTFPDU, [531](#)
- GetNumberOfPrimitivesData
 - gdcm::MeshPrimitive, [499](#)
- GetNumberOfReferencedFrameOfReferences
 - vtkRTStructSetProperties, [881](#)
- GetNumberOfSOPClassToIOD
 - gdcm::SOPClassUIDToIOD, [661](#)
- GetNumberOfSegments
 - gdcm::SegmentWriter, [631](#)
- GetNumberOfStructureSetROIs
 - vtkRTStructSetProperties, [881](#)
- GetNumberOfSurfacePoints
 - gdcm::Surface, [691](#)
- GetNumberOfSurfaces
 - gdcm::SurfaceReader, [698](#)
 - gdcm::SurfaceWriter, [700](#)
- GetNumberOfTransferSyntaxStrings
 - gdcm::UIDs, [755](#)
- GetNumberOfTransferSyntaxes
 - gdcm::network::PresentationContextRQ, [572](#)
 - gdcm::PresentationContext, [567](#)
- GetNumberOfValues
 - gdcm::Attribute, [166](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [179](#)
- GetNumberOfVectors
 - gdcm::Surface, [691](#)
- GetObliquityThresholdCosineValue
 - gdcm::Orientation, [518](#)
- GetOffScreenRendering
 - vtkImageColorViewer, [862](#)
- GetOptionalTags
 - gdcm::QueryBase, [587](#)
 - gdcm::QueryImage, [590](#)
 - gdcm::QueryPatient, [592](#)
 - gdcm::QuerySeries, [594](#)
 - gdcm::QueryStudy, [596](#)
- GetOrderedValues
 - gdcm::Scanner, [619](#)
- GetOrigin
 - gdcm::Image, [398](#)
 - gdcm::Overlay, [522](#)
- GetOriginValue
 - gdcm::ImageHelper, [423](#)
- GetOutput
 - gdcm::ImageConverter, [419](#)
- GetOutput
 - gdcm::BitmapToBitmapFilter, [215](#)
 - gdcm::ImageToImageFilter, [433](#)
 - gdcm::PixmapToPixmapFilter, [558](#)
- GetOutputAsBitmap
 - gdcm::BitmapToBitmapFilter, [215](#)
- GetOutputAsPixmap
 - gdcm::PixmapToPixmapFilter, [558](#)
- GetOutputFilename
 - gdcm::DictConverter, [307](#)

GetOutputType
 gdcm::DictConverter, 307

GetOverlay
 gdcm::Pixmap, 552
 vtkGDCMImageReader, 836

GetOverlayData
 gdcm::Overlay, 522

GetOverlayTypeAsString
 gdcm::Overlay, 523

GetOverlayTypeFromString
 gdcm::Overlay, 523

GetOverlayVisibility
 vtkImageColorViewer, 862

GetOwner
 gdcm::PrivateTag, 580

GetPDBEEnd
 gdcm::PDBHeader, 535

GetPDBElementByName
 gdcm::PDBHeader, 535

GetPDBInfoTag
 gdcm::PDBHeader, 535

GetPDUs
 gdcm::network::ULEvent, 801

GetPDVs
 gdcm::network::PDUFactory, 538

GetPIString
 gdcm::PhotometricInterpretation, 543

GetPIType
 gdcm::PhotometricInterpretation, 543

GetPath
 gdcm::Filename, 374

GetPattern
 gdcm::FilenameGenerator, 376

GetPermissions
 gdcm::System, 706

GetPhotometricInterpretation
 gdcm::Bitmap, 209
 gdcm::ImageChangePhotometricInterpretation, 405
 gdcm::ImageCodec, 416

GetPhotometricInterpretationValue
 gdcm::ImageHelper, 423

GetPixelFormat
 gdcm::Bitmap, 209, 210
 gdcm::ImageCodec, 416

GetPixelFormatValue
 gdcm::ImageHelper, 423

GetPixelRepresentation
 gdcm::PixelFormat, 547

GetPixelSize
 gdcm::PixelFormat, 547

GetPixelSpacingDataRoot
 gdcm::Testing, 723

GetPixmap
 gdcm::IconImageGenerator, 395
 gdcm::PixmapReader, 555
 gdcm::PixmapWriter, 560, 561

GetPlanarConfiguration
 gdcm::Bitmap, 210
 gdcm::ImageChangePlanarConfiguration, 408
 gdcm::ImageCodec, 416

GetPlanarConfigurationValue
 gdcm::ImageHelper, 423

GetPointCoordinatesData
 gdcm::Surface, 691

GetPointPositionAccuracy
 gdcm::Surface, 691

GetPointer
 gdcm::ByteValue, 224
 gdcm::LookupTable, 479
 gdcm::SmartPointer, 659
 vtkLookupTable16, 878

GetPointerFromElement
 gdcm::ImageHelper, 423

GetPointsBoundingBoxCoordinates
 gdcm::Surface, 691

GetPosition
 vtkImageColorViewer, 862

GetPreamble
 gdcm::FileMetaInformation, 371

GetPrefix
 gdcm::FilenameGenerator, 377

GetPresentationContext
 gdcm::network::AAssociateRQPDU, 142

GetPresentationContextAC
 gdcm::network::AAssociateACPDU, 137

GetPresentationContextACByID
 gdcm::network::ULConnection, 794

GetPresentationContextByAbstractSyntax
 gdcm::network::AAssociateRQPDU, 142

GetPresentationContextByID
 gdcm::network::AAssociateRQPDU, 142

GetPresentationContextID
 gdcm::network::PresentationContextAC, 568
 gdcm::network::PresentationContextRQ, 572
 gdcm::network::PresentationDataValue, 573
 gdcm::PresentationContext, 567

GetPresentationContextIDFromPresentationContext
 gdcm::network::ULConnection, 794

GetPresentationContextRQByID
 gdcm::network::ULConnection, 794

GetPresentationContexts
 gdcm::network::AAssociateRQPDU, 142
 gdcm::network::ULConnection, 794
 gdcm::PresentationContextGenerator, 570

GetPresentationDataValue
 gdcm::network::PDataTFPDU, 531

GetPrimitiveData
 gdcm::MeshPrimitive, 499

- GetPrimitiveType
 - gdcm::MeshPrimitive, [499](#)
- GetPrimitivesData
 - gdcm::MeshPrimitive, [499](#)
- GetPrintStyle
 - gdcm::Printer, [576](#)
- GetPrivateCreator
 - gdcm::DataSet, [289](#)
 - gdcm::Tag, [715](#)
- GetPrivateDict
 - gdcm::Dicts, [314](#)
 - gdcm::XMLPrivateDictReader, [892](#)
- GetProcessingAlgorithm
 - gdcm::Surface, [692](#)
- GetProgress
 - gdcm::ProgressEvent, [582](#)
- GetPropertyCategory
 - gdcm::Segment, [623](#)
- GetPropertyType
 - gdcm::Segment, [623](#)
- GetProtocol
 - gdcm::network::ULConnection, [794](#)
- GetPublicDict
 - gdcm::Dicts, [314](#)
- GetQuality
 - gdcm::JPEG2000Codec, [461](#)
 - gdcm::JPEGCodec, [467](#)
- GetQueryDataSet
 - gdcm::BaseRootQuery, [199](#)
- GetQueryLevel
 - gdcm::QueryBase, [587](#)
 - gdcm::QueryImage, [590](#)
 - gdcm::QueryPatient, [592](#)
 - gdcm::QuerySeries, [594](#)
 - gdcm::QueryStudy, [596](#)
- GetQueryLevelFromQueryRoot
 - gdcm::BaseRootQuery, [199](#)
- GetQueryLevelFromString
 - gdcm::BaseRootQuery, [199](#)
- GetQueryLevelString
 - gdcm::BaseRootQuery, [199](#)
- GetRAWMD5FromFile
 - vtkGDCMTesting, [853](#)
- GetRTStructSeriesUIDs
 - gdcm::DirectoryHelper, [321](#)
- GetRate
 - gdcm::JPEG2000Codec, [461](#)
- GetReason
 - gdcm::network::PresentationContextAC, [568](#)
- GetRecommendedDisplayCIELabValue
 - gdcm::Surface, [692](#)
- GetRecommendedDisplayGrayscaleValue
 - gdcm::Surface, [692](#)
- GetRecommendedPresentationOpacity
 - gdcm::Surface, [692](#)
- GetRecommendedPresentationType
 - gdcm::Surface, [692](#)
- GetRef
 - gdcm::IODEntry, [444](#)
- GetReferencedFrameOfReferenceClassUID
 - vtkRTStructSetProperties, [881](#)
- GetReferencedFrameOfReferenceInstanceUID
 - vtkRTStructSetProperties, [881](#)
- GetRegion
 - gdcm::ImageRegionReader, [430](#)
- GetRequiredTags
 - gdcm::QueryBase, [587](#)
 - gdcm::QueryImage, [591](#)
 - gdcm::QueryPatient, [593](#)
 - gdcm::QuerySeries, [595](#)
 - gdcm::QueryStudy, [597](#)
- GetRescaleInterceptSlopeValue
 - gdcm::ImageHelper, [423](#)
- GetReserved43_74
 - gdcm::network::AAssociateRQPDU, [142](#)
- GetResponses
 - gdcm::network::ULBasicCallback, [792](#)
- GetRetired
 - gdcm::DictEntry, [308](#)
- GetRoot
 - gdcm::UIDGenerator, [737](#)
- GetRows
 - gdcm::Bitmap, [210](#)
 - gdcm::Overlay, [523](#)
- GetSOPClassUID
 - gdcm::DirectoryHelper, [322](#)
- GetSOPClassUIDFromIOD
 - gdcm::SOPClassUIDToIOD, [661](#)
- GetSOPClassUIDToIOD
 - gdcm::SOPClassUIDToIOD, [661](#)
- GetSOPClassUIDToIODs
 - gdcm::SOPClassUIDToIOD, [661](#)
- GetSTATES
 - gdcm::Surface, [692](#)
- GetSTATESString
 - gdcm::Surface, [692](#)
- GetSamplesPerPixel
 - gdcm::PhotometricInterpretation, [544](#)
 - gdcm::PixelFormat, [547](#)
- GetScalarType
 - gdcm::PixelFormat, [548](#)
- GetScalarTypeAsString
 - gdcm::PixelFormat, [548](#)
- GetScanner
 - gdcm::DICOMDIRGenerator, [302](#)
- GetSegment
 - gdcm::SegmentWriter, [631](#)
- GetSegmentAlgorithmName

- gdcm::Segment, [623](#)
- GetSegmentAlgorithmType
 - gdcm::Segment, [623](#)
- GetSegmentDescription
 - gdcm::Segment, [623](#)
- GetSegmentLabel
 - gdcm::Segment, [623](#)
- GetSegmentNumber
 - gdcm::Segment, [623](#)
- GetSegments
 - gdcm::SegmentReader, [629](#)
 - gdcm::SegmentWriter, [631](#)
- GetSelectedTagsOffsetFromFile
 - gdcm::Testing, [723](#)
- GetSequenceOfFragments
 - gdcm::DataElement, [276](#)
- GetSequenceOfItems
 - gdcm::DataElement, [276](#), [277](#)
- GetSeriesUIDsBySOPClassUID
 - gdcm::DirectoryHelper, [322](#)
- GetSize
 - gdcm::VR, [828](#)
 - vtkImageColorViewer, [863](#)
- GetSizeof
 - gdcm::VR, [828](#)
- GetSliceMax
 - vtkImageColorViewer, [863](#)
- GetSliceMin
 - vtkImageColorViewer, [863](#)
- GetSliceRange
 - vtkImageColorViewer, [863](#)
- GetSlope
 - gdcm::Image, [399](#)
 - gdcm::Rescaler, [608](#)
- GetSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, [371](#)
- GetSourceDirectory
 - gdcm::Testing, [723](#)
- GetSpacing
 - gdcm::Image, [399](#)
- GetSpacingTagFromMediaStorage
 - gdcm::ImageHelper, [423](#)
- GetSpacingValue
 - gdcm::ImageHelper, [424](#)
- GetStart
 - gdcm::ByteBuffer, [219](#)
- GetState
 - gdcm::network::ULConnection, [794](#)
- GetStateIndex
 - gdcm::network, [129](#)
- GetStream
 - gdcm::Trace, [725](#)
- GetStreamOffsetFromFile
 - gdcm::Testing, [723](#)
- GetStreamPtr
 - gdcm::Reader, [603](#)
 - gdcm::Writer, [887](#)
- GetString
 - gdcm::MediaStorage, [491](#)
 - gdcm::PhotometricInterpretation, [544](#)
 - gdcm::TransferSyntax, [730](#)
 - gdcm::UIDs, [756](#)
- GetStringValueFromTag
 - gdcm::DirectoryHelper, [322](#)
- GetStructureSetObservationNumber
 - vtkRTStructSetProperties, [881](#)
- GetStructureSetROIDescription
 - vtkRTStructSetProperties, [881](#)
- GetStructureSetROIGenerationAlgorithm
 - vtkRTStructSetProperties, [881](#)
- GetStructureSetROIName
 - vtkRTStructSetProperties, [881](#)
- GetStructureSetROINumber
 - vtkRTStructSetProperties, [881](#)
- GetStructureSetROIObservationLabel
 - vtkRTStructSetProperties, [881](#)
- GetStructureSetROIRefFrameRefUID
 - vtkRTStructSetProperties, [881](#)
- GetStructureSetRTROIInterpretedType
 - vtkRTStructSetProperties, [881](#)
- GetSurface
 - gdcm::Segment, [623](#)
- GetSurfaceComments
 - gdcm::Surface, [692](#)
- GetSurfaceCount
 - gdcm::Segment, [623](#)
- GetSurfaceNumber
 - gdcm::Surface, [692](#)
- GetSurfaceProcessing
 - gdcm::Surface, [692](#)
- GetSurfaceProcessingDescription
 - gdcm::Surface, [692](#)
- GetSurfaceProcessingRatio
 - gdcm::Surface, [692](#)
- GetSurfaces
 - gdcm::Segment, [624](#)
- GetSwapCode
 - gdcm::TransferSyntax, [730](#)
- GetSwapCodeString
 - gdcm::SwapCode, [702](#)
- GetSyngoDT
 - gdcm::CSAElement, [256](#)
- GetTSString
 - gdcm::TransferSyntax, [730](#)
- GetTSType
 - gdcm::TransferSyntax, [730](#)
- GetTable
 - gdcm::SequenceOfFragments, [635](#)

- GetTableEntry
 - gdcm::Table, 708
- GetTag
 - gdcm::AnonymizeEvent, 147
 - gdcm::Attribute, 166
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 180
 - gdcm::DataElement, 277
- GetTagListByLevel
 - gdcm::BaseRootQuery, 199
 - gdcm::FindPatientRootQuery, 381
 - gdcm::FindStudyRootQuery, 384
 - gdcm::MovePatientRootQuery, 508
 - gdcm::MoveStudyRootQuery, 510
- GetTempDirectory
 - gdcm::Testing, 723
- GetTempDirectoryW
 - gdcm::Testing, 723
- GetTempFilename
 - gdcm::Testing, 723
- GetTempFilenameW
 - gdcm::Testing, 723
- GetTimeout
 - gdcm::network::ARTIMTimer, 161
 - gdcm::ServiceClassUser, 648
- GetTimer
 - gdcm::network::ULConnection, 794
- GetTimezoneOffsetFromUTC
 - gdcm::System, 706
- GetToplevel
 - gdcm::Directory, 320
- GetTransferSyntax
 - gdcm::Bitmap, 210
 - gdcm::ImageChangeTransferSyntax, 411
 - gdcm::network::PresentationContextAC, 568
 - gdcm::network::PresentationContextRQ, 572
 - gdcm::PresentationContext, 567
- GetTransferSyntaxString
 - gdcm::UIDs, 756
- GetTransferSyntaxStrings
 - gdcm::UIDs, 756
- GetTransferSyntaxes
 - gdcm::network::PresentationContextRQ, 572
- GetType
 - gdcm::ModuleEntry, 505
 - gdcm::Orientation, 518
 - gdcm::Overlay, 523
 - gdcm::PhotometricInterpretation, 544
- GetTypeAsEnum
 - gdcm::Overlay, 523
- GetTypeFromTag
 - gdcm::Defs, 297
 - gdcm::IOD, 442
- GetTypeOfData
 - gdcm::Curve, 271
- GetTypeOfDataDescription
 - gdcm::Curve, 271
- GetTypeString
 - gdcm::Type, 735
- GetTypeType
 - gdcm::Type, 735
- GetUIDName
 - gdcm::UIDs, 756
- GetUIDString
 - gdcm::UIDs, 756
- GetUniqueTags
 - gdcm::QueryBase, 587
 - gdcm::QueryImage, 591
 - gdcm::QueryPatient, 593
 - gdcm::QuerySeries, 595
 - gdcm::QueryStudy, 597
- GetUnpackBuffer
 - gdcm::Overlay, 523
- GetUnpackBufferLength
 - gdcm::Overlay, 523
- GetUsage
 - gdcm::IODEntry, 444
- GetUsageString
 - gdcm::Usage, 810
- GetUsageType
 - gdcm::IODEntry, 444
 - gdcm::Usage, 810
- GetUserData
 - gdcm::Parser, 528
- GetUserInformation
 - gdcm::network::AAssociateACPDU, 137
 - gdcm::network::AAssociateRQPDU, 142
- GetVIEWType
 - gdcm::Surface, 692
- GetVIEWTypeString
 - gdcm::Surface, 692
- GetVL
 - gdcm::DataElement, 278
- GetVL16Max
 - gdcm::VL, 819
- GetVL32Max
 - gdcm::VL, 819
- GetVM
 - gdcm::Attribute, 167
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 173
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >, 176
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >, 177

- gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [180](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM2_-2n >, [183](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM2_n >, [184](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM3_-3n >, [186](#)
- gdcm::Attribute< Group, Element, TVR, VM::VM3_n >, [187](#)
- gdcm::CSAElement, [256](#)
- gdcm::CSAHeaderDictEntry, [265](#)
- gdcm::DictEntry, [309](#)
- gdcm::Element, [327](#)
- gdcm::Element< TVR, VM::VM1_n >, [331](#)
- GetVMString
 - gdcm::VM, [823](#)
- GetVMType
 - gdcm::VM, [823](#)
- GetVMTypeFromLength
 - gdcm::VM, [824](#)
- GetVR
 - gdcm::Attribute, [167](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [180](#)
 - gdcm::CSAElement, [256](#)
 - gdcm::CSAHeaderDictEntry, [265](#)
 - gdcm::DataElement, [278](#)
 - gdcm::DictEntry, [309](#)
 - gdcm::Element, [327](#)
 - gdcm::Element< TVR, VM::VM1_n >, [331](#)
- GetVRFromTag
 - gdcm, [119](#)
- GetVRString
 - gdcm::VR, [828](#)
- GetVRStringFromFile
 - gdcm::VR, [828](#)
- GetVRType
 - gdcm::VR, [828](#)
- GetVRTypeFromFile
 - gdcm::VR, [828](#)
- GetVTKDataRoot
 - vtkGDCMTesting, [853](#)
- GetValidatedFile
 - gdcm::Validate, [814](#)
- GetValue
 - gdcm::Attribute, [166](#), [167](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [180](#)
 - gdcm::CSAElement, [256](#)
 - gdcm::DataElement, [277](#)
 - gdcm::Element, [327](#)
 - gdcm::Element< TVR, VM::VM1_n >, [331](#)
 - gdcm::PDBelement, [533](#)
 - gdcm::Scanner, [619](#)
- GetValueAsSQ
 - gdcm::DataElement, [277](#)
- GetValues
 - gdcm::Attribute, [167](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [173](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [180](#)
 - gdcm::Element, [327](#)
 - gdcm::Scanner, [619](#)
- GetVectorAccuracy
 - gdcm::Surface, [692](#)
- GetVectorCoordinateData
 - gdcm::Surface, [692](#)
- GetVectorDimensionality
 - gdcm::Surface, [692](#)
- GetVersion
 - gdcm::Version, [817](#)
- GetWarningFlag
 - gdcm::Trace, [725](#)
- GetWarningStream
 - gdcm::Trace, [726](#)
- GetWindowName
 - vtkImageColorViewer, [863](#)
- GetXMax
 - gdcm::BoxRegion, [217](#)
- GetXMin
 - gdcm::BoxRegion, [217](#)
- GetYMax
 - gdcm::BoxRegion, [217](#)
- GetYMin
 - gdcm::BoxRegion, [218](#)
- GetZMax
 - gdcm::BoxRegion, [218](#)
- GetZMin
 - gdcm::BoxRegion, [218](#)
- GetZSpacing
 - gdcm::IPPSorter, [448](#)
- GetZSpacingTagFromMediaStorage
 - gdcm::ImageHelper, [424](#)
- GetZSpacingTolerance
 - gdcm::IPPSorter, [448](#)
- Global
 - gdcm::Defs, [298](#)
 - gdcm::Dicts, [314](#)
 - gdcm::Global, [388](#)
- GlobalInstance
 - gdcm, [124](#)
- GrabOverlayFromPixelData

- gdcm::Overlay, [523](#)
- Graphics
 - gdcm::Overlay, [521](#)
- GrayscaleSoftcopyPresentationStateStorageSOPClass
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [746](#)
- green
 - gdcm::terminal, [131](#)
- group
 - gdcm::SerieHelper::Rule, [614](#)
- GroupDict
 - gdcm::GroupDict, [390](#)
- GroupStringVector
 - gdcm::GroupDict, [390](#)
- GuessFromModality
 - gdcm::MediaStorage, [491](#)
- HSV
 - gdcm::PhotometricInterpretation, [543](#)
- HandleDataSet
 - gdcm::network::ULBasicCallback, [792](#)
 - gdcm::network::ULConnectionCallback, [796](#)
 - gdcm::network::ULWritingCallback, [804](#)
- HandleDescription
 - gdcm::XMLDictReader, [890](#)
 - gdcm::XMLPrivateDictReader, [892](#)
- HandleEntry
 - gdcm::XMLDictReader, [890](#)
 - gdcm::XMLPrivateDictReader, [892](#)
- HandleEvent
 - gdcm::network::ULTransitionTable, [802](#)
- HandleIOD
 - gdcm::TableReader, [710](#)
- HandleIODEntry
 - gdcm::TableReader, [710](#)
- HandleMacro
 - gdcm::TableReader, [710](#)
- HandleMacroEntry
 - gdcm::TableReader, [710](#)
- HandleMacroEntryDescription
 - gdcm::TableReader, [710](#)
- HandleModule
 - gdcm::TableReader, [710](#)
- HandleModuleEntry
 - gdcm::TableReader, [710](#)
- HandleModuleEntryDescription
 - gdcm::TableReader, [711](#)
- HandleModuleInclude
 - gdcm::TableReader, [711](#)
- HandleResponse
 - gdcm::network::ULBasicCallback, [792](#)
 - gdcm::network::ULConnectionCallback, [796](#)
 - gdcm::network::ULWritingCallback, [804](#)
- HangingProtocolInformationModelFIND
 - gdcm::UIDs, [748](#)
- HangingProtocolInformationModelMOVE
 - gdcm::UIDs, [748](#)
- HangingProtocolStorage
 - gdcm::MediaStorage, [490](#)
 - gdcm::UIDs, [748](#)
- HardcopyColorImageStorageSOPClassRetired
 - gdcm::UIDs, [745](#)
- HardcopyGrayscaleImageStorage
 - gdcm::MediaStorage, [489](#)
- HardcopyGrayscaleImageStorageSOPClassRetired
 - gdcm::UIDs, [745](#)
- HasObserver
 - gdcm::Subject, [686](#)
- HemodynamicWaveformStorage
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [745](#)
- hidden
 - gdcm::terminal, [131](#)
- ICBM452T1FrameofReference
 - gdcm::UIDs, [744](#)
- ICBMSingleSubjectMRIFrameofReference
 - gdcm::UIDs, [744](#)
- INT12
 - gdcm::PixelFormat, [546](#)
- INT16
 - gdcm::PixelFormat, [546](#)
- INT32
 - gdcm::PixelFormat, [546](#)
- INT8
 - gdcm::PixelFormat, [546](#)
- INTERFILE
 - gdcm::CSAHeader, [260](#)
- INVALID
 - gdcm::VR, [826](#)
- IS
 - gdcm::VR, [827](#)
- IOD
 - gdcm::IOD, [442](#)
- IODEntry
 - gdcm::IODEntry, [444](#)
- IODMapType
 - gdcm::IODs, [445](#)
- IODMapTypeConstIterator
 - gdcm::IODs, [445](#)
- IODName
 - gdcm::IODs, [445](#)
- IODs
 - gdcm::IODs, [446](#)
- IPPSorter
 - gdcm::IPPSorter, [448](#)
- Icon
 - gdcm::Pixmap, [552](#)

- IconDataScalarType
 - vtkGDCMImageReader, 838
- IconImage
 - gdcm, 117
- IconImageDataExtent
 - vtkGDCMImageReader, 838
- IconImageFilter
 - gdcm::IconImageFilter, 392
- IconImageGenerator
 - gdcm::IconImageGenerator, 394
- IconNumberOfScalarComponents
 - vtkGDCMImageReader, 838
- ignore_char
 - gdcm::ignore_char, 396
- Image
 - gdcm::Image, 398
- ImageOverlayBoxSOPClassRetired
 - gdcm::UIDs, 745
- ImageActor
 - vtkImageColorViewer, 865
- ImageApplyLookupTable
 - gdcm::ImageApplyLookupTable, 402
- ImageChangePhotometricInterpretation
 - gdcm::ImageChangePhotometricInterpretation, 405
 - gdcm::ImageCodec, 417
- ImageChangePlanarConfiguration
 - gdcm::ImageChangePlanarConfiguration, 408
- ImageChangeTransferSyntax
 - gdcm::Bitmap, 212
 - gdcm::ImageChangeTransferSyntax, 411
- ImageCodec
 - gdcm::ImageCodec, 415
- ImageConverter
 - gdcm::ImageConverter, 419
- ImageFormat
 - vtkGDCMImageReader, 838
- ImageFragmentSplitter
 - gdcm::ImageFragmentSplitter, 421
- ImageOrientationPatient
 - vtkGDCMImageReader, 838
- ImagePositionPatient
 - vtkGDCMImageReader, 838
- ImagePositionPatientOrdering
 - gdcm::SerieHelper, 644
- ImageReader
 - gdcm::ImageReader, 427
- ImageRegionReader
 - gdcm::ImageRegionReader, 430
 - gdcm::JPEG2000Codec, 461
 - gdcm::JPEGCodec, 467
 - gdcm::JPEGLSCCodec, 471
 - gdcm::RLECodec, 612
- ImageToImageFilter
 - gdcm::ImageToImageFilter, 433
- ImageWriter
 - gdcm::ImageWriter, 435
- ImplementationClassUIDSub
 - gdcm::network::ImplementationClassUIDSub, 436
- ImplementationUIDSub
 - gdcm::network::ImplementationUIDSub, 437
- ImplementationVersionNameSub
 - gdcm::network::ImplementationVersionNameSub, 437
- Implicit
 - gdcm::TransferSyntax, 729
- ImplicitVRBigEndianACRNEMA
 - gdcm::TransferSyntax, 729
- ImplicitVRBigEndianPrivateGE
 - gdcm::TransferSyntax, 729
- ImplicitVRLittleEndian
 - gdcm::TransferSyntax, 729
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
 - gdcm::UIDs, 742
- IncompleteLUT
 - gdcm::LookupTable, 480
- InitFromRQ
 - gdcm::network::AAssociateACPDU, 137
- Initialize
 - gdcm::network::ULConnectionInfo, 797
- InitializeBlueLUT
 - gdcm::LookupTable, 479
- InitializeConnection
 - gdcm::network::ULConnection, 795
 - gdcm::ServiceClassUser, 648
- InitializeDataSet
 - gdcm::BaseRootQuery, 199
 - gdcm::FindPatientRootQuery, 382
 - gdcm::FindStudyRootQuery, 384
 - gdcm::MovePatientRootQuery, 508
 - gdcm::MoveStudyRootQuery, 511
- InitializeGreenLUT
 - gdcm::LookupTable, 479
- InitializeIncomingConnection
 - gdcm::network::ULConnection, 795
- InitializeLUT
 - gdcm::LookupTable, 479
- InitializeRTStructSet
 - vtkGDCMPolyDataWriter, 850
- InitializeRedLUT
 - gdcm::LookupTable, 479
- Initialized
 - gdcm::LookupTable, 479
- Input
 - gdcm::BitmapToBitmapFilter, 215
- Insert
 - gdcm::CommandDataSet, 245
 - gdcm::DataSet, 289
 - gdcm::FileMetaInformation, 371

- gdcmm::GroupDict, 390
- InsertDataElement
 - gdcmm::DataSet, 289
 - gdcmm::Item, 453
- InsertEntry
 - gdcmm::Table, 708
- InstallPipeline
 - vtkImageColorViewer, 863
- InstanceAvailabilityNotificationSOPClass
 - gdcmm::UIDs, 747
- Interactor
 - vtkImageColorViewer, 865
- InteractorStyle
 - vtkImageColorViewer, 865
- Internal
 - gdcmm::ApplicationEntity, 156
 - gdcmm::Attribute, 170
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 175
 - gdcmm::Element, 328
 - gdcmm::Element< VR::AS, VM::VM5 >, 339
 - gdcmm::LookupTable, 480
 - gdcmm::UI, 735
- InternalCode
 - gdcmm::Coder, 239
 - gdcmm::JPEG12Codec, 456
 - gdcmm::JPEG16Codec, 458
 - gdcmm::JPEG8Codec, 463
- Internals
 - vtkRTStructSetProperties, 882
- Invalid
 - gdcmm::Overlay, 521
 - gdcmm::Usage, 810
- InverseRescale
 - gdcmm::Rescaler, 608
- InverseRescaleFunctionIntoBestFit
 - gdcmm::Rescaler, 608
- InvokeEvent
 - gdcmm::Subject, 686, 687
- IsAETitleValid
 - gdcmm::network::AAAssociateRQPDU, 142
- IsASCII
 - gdcmm::VR, 828
- IsASCII2
 - gdcmm::VR, 828
- IsBinary
 - gdcmm::VR, 828
- IsBinary2
 - gdcmm::VR, 828
- IsDual
 - gdcmm::VR, 828
- IsEmpty
 - gdcmm::Bitmap, 210
 - gdcmm::ByteValue, 224
- gdcmm::CSAElement, 256
- gdcmm::CSAHeaderDict, 263
- gdcmm::Curve, 271
- gdcmm::DataElement, 278
- gdcmm::DataSet, 289
- gdcmm::Defs, 297
- gdcmm::Dict, 305
- gdcmm::Dicts, 314
- gdcmm::Filename, 374
- gdcmm::Macros, 483
- gdcmm::Modules, 506
- gdcmm::Overlay, 523
- gdcmm::Preamble, 565
- gdcmm::PrivateDict, 578
- gdcmm::SegmentHelper::BasicCodedEntry, 202
- IsEncapsulated
 - gdcmm::TransferSyntax, 730
- IsEncoded
 - gdcmm::TransferSyntax, 730
- IsExplicit
 - gdcmm::TransferSyntax, 730
- IsGroupLength
 - gdcmm::Tag, 715
- IsGroupXX
 - gdcmm::Tag, 715
- IsIdentical
 - gdcmm::Filename, 374
- IsIllegal
 - gdcmm::Tag, 715
- IsImage
 - gdcmm::MediaStorage, 491
- IsImplicit
 - gdcmm::TransferSyntax, 730
- IsInPixelData
 - gdcmm::Overlay, 523
- IsKey
 - gdcmm::Scanner, 619
- IsLastFragment
 - gdcmm::network::AAAbortPDU, 134
 - gdcmm::network::AAAssociateACPDU, 137
 - gdcmm::network::AAAssociateRJPD, 139
 - gdcmm::network::AAAssociateRQPDU, 142
 - gdcmm::network::AResponseRPPDU, 158
 - gdcmm::network::AResponseRQPDU, 160
 - gdcmm::network::BasePDU, 196
 - gdcmm::network::PDataTFPDU, 531
- IsLossless
 - gdcmm::PhotometricInterpretation, 544
 - gdcmm::TransferSyntax, 730
- IsLossy
 - gdcmm::Bitmap, 210
 - gdcmm::ImageCodec, 416
 - gdcmm::PhotometricInterpretation, 544
 - gdcmm::TransferSyntax, 730

- IsOdd
 - gdcm::VL, [819](#)
- IsPresentationContextAccepted
 - gdcm::ServiceClassUser, [649](#)
- IsPrintable
 - gdcm::ByteValue, [224](#)
- IsPrivate
 - gdcm::Tag, [715](#)
- IsPrivateCreator
 - gdcm::Tag, [716](#)
- IsPublic
 - gdcm::Tag, [716](#)
- IsRetired
 - gdcm::PhotometricInterpretation, [544](#)
- IsSameColorSpace
 - gdcm::PhotometricInterpretation, [544](#)
- IsStateSuspension
 - gdcm::JPEG12Codec, [456](#)
 - gdcm::JPEG16Codec, [458](#)
 - gdcm::JPEG8Codec, [463](#)
 - gdcm::JPEGCodec, [467](#)
- IsSwap
 - gdcm::VR, [828](#)
- IsTransferSyntaxCompatible
 - gdcm::Bitmap, [210](#)
- IsUndefined
 - gdcm::MediaStorage, [491](#)
 - gdcm::VL, [819](#)
- IsUndefinedLength
 - gdcm::DataElement, [278](#)
 - gdcm::SequenceOfItems, [640](#)
- IsUnique
 - gdcm::DictEntry, [309](#)
- IsVRFile
 - gdcm::VR, [828](#)
- IsValid
 - gdcm::ApplicationEntity, [156](#)
 - gdcm::BoxRegion, [218](#)
 - gdcm::CodeString, [241](#)
 - gdcm::DirectionCosines, [317](#)
 - gdcm::FileMetaInformation, [371](#)
 - gdcm::ImageCodec, [416](#)
 - gdcm::JPEGCodec, [467](#)
 - gdcm::LO, [475](#)
 - gdcm::PixelFormat, [548](#)
 - gdcm::Preamble, [565](#)
 - gdcm::Region, [606](#)
 - gdcm::String, [681](#)
 - gdcm::TagPath, [719](#)
 - gdcm::TransferSyntax, [730](#)
 - gdcm::UIDGenerator, [737](#)
 - gdcm::VM, [824](#)
 - gdcm::VR, [828](#)
- IsZero
 - gdcm::Overlay, [523](#)
- ItFileSetHt
 - gdcm::SerieHelper, [644](#)
- Item
 - gdcm::Item, [452](#)
- ItemVector
 - gdcm::SequenceOfItems, [639](#)
- Items
 - gdcm::SequenceOfItems, [641](#)
- Iterator
 - gdcm::CSAHeaderDict, [263](#)
 - gdcm::DataSet, [287](#)
 - gdcm::Dict, [304](#)
 - gdcm::SequenceOfFragments, [634](#)
 - gdcm::SequenceOfItems, [639](#)
- iterator
 - gdcm::CodeString, [240](#)
 - gdcm::LO, [475](#)
 - gdcm::String, [681](#)
- JPEG2000
 - gdcm::TransferSyntax, [729](#)
- JPEG2000_COMPRESSION
 - vtkGDCMImageWriter, [841](#)
- JPEG2000ImageCompression
 - gdcm::UIDs, [743](#)
- JPEG2000ImageCompressionLosslessOnly
 - gdcm::UIDs, [743](#)
- JPEG2000Lossless
 - gdcm::TransferSyntax, [729](#)
- JPEG2000Part2
 - gdcm::TransferSyntax, [729](#)
- JPEG2000Part2Lossless
 - gdcm::TransferSyntax, [729](#)
- JPEG2000Part2MulticomponentImageCompression
 - gdcm::UIDs, [743](#)
- JPEG2000Part2MulticomponentImageCompression-LosslessOnly
 - gdcm::UIDs, [743](#)
- JPEG_COMPRESSION
 - vtkGDCMImageWriter, [841](#)
- JPEGBaselineProcess1
 - gdcm::TransferSyntax, [729](#)
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEGBitImageCompression
 - gdcm::UIDs, [742](#)
- JPEGExtendedHierarchicalProcess1618Retired
 - gdcm::UIDs, [743](#)
- JPEGExtendedHierarchicalProcess1719Retired
 - gdcm::UIDs, [743](#)
- JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEGBitImageCompressionProcess4only
 - gdcm::UIDs, [742](#)
- JPEGExtendedProcess2_4

- gdcm::TransferSyntax, [729](#)
- JPEGExtendedProcess35Retired
 - gdcm::UIDs, [742](#)
- JPEGExtendedProcess3_5
 - gdcm::TransferSyntax, [729](#)
- JPEGFullProgressionHierarchicalProcess2426Retired
 - gdcm::UIDs, [743](#)
- JPEGFullProgressionHierarchicalProcess2527Retired
 - gdcm::UIDs, [743](#)
- JPEGFullProgressionNonHierarchicalProcess1012-Retired
 - gdcm::UIDs, [742](#)
- JPEGFullProgressionNonHierarchicalProcess1113-Retired
 - gdcm::UIDs, [742](#)
- JPEGFullProgressionProcess10_12
 - gdcm::TransferSyntax, [729](#)
- JPEGLS_COMPRESSION
 - vtkGDCMImageWriter, [841](#)
- JPEGLSLossless
 - gdcm::TransferSyntax, [729](#)
- JPEGLSLosslessImageCompression
 - gdcm::UIDs, [743](#)
- JPEGLSLossyNearLosslessImageCompression
 - gdcm::UIDs, [743](#)
- JPEGLSNearLossless
 - gdcm::TransferSyntax, [729](#)
- JPEGLosslessHierarchicalProcess28Retired
 - gdcm::UIDs, [743](#)
- JPEGLosslessHierarchicalProcess29Retired
 - gdcm::UIDs, [743](#)
- JPEGLosslessNonHierarchicalFirstOrderPrediction-Process14SelectionValue1DefaultTransfer-SyntaxforLosslessJPEGImageCompression
 - gdcm::UIDs, [743](#)
- JPEGLosslessNonHierarchicalProcess14
 - gdcm::UIDs, [742](#)
- JPEGLosslessNonHierarchicalProcess15Retired
 - gdcm::UIDs, [743](#)
- JPEGLosslessProcess14
 - gdcm::TransferSyntax, [729](#)
- JPEGLosslessProcess14_1
 - gdcm::TransferSyntax, [729](#)
- JPEGSpectralSelectionHierarchicalProcess2022Retired
 - gdcm::UIDs, [743](#)
- JPEGSpectralSelectionHierarchicalProcess2123Retired
 - gdcm::UIDs, [743](#)
- JPEGSpectralSelectionNonHierarchicalProcess68Retired
 - gdcm::UIDs, [742](#)
- JPEGSpectralSelectionNonHierarchicalProcess79Retired
 - gdcm::UIDs, [742](#)
- JPEGSpectralSelectionProcess6_8
 - gdcm::TransferSyntax, [729](#)
- JPIPReferenced
 - gdcm::TransferSyntax, [729](#)
 - gdcm::UIDs, [743](#)
- JPIPReferencedDeflate
 - gdcm::UIDs, [743](#)
- JPEG12Codec
 - gdcm::JPEG12Codec, [456](#)
- JPEG16Codec
 - gdcm::JPEG16Codec, [458](#)
- JPEG2000Codec
 - gdcm::JPEG2000Codec, [460](#)
- JPEG8Codec
 - gdcm::JPEG8Codec, [463](#)
- JPEGCodec
 - gdcm::JPEGCodec, [466](#)
- JPEGLSCodec
 - gdcm::JPEGLSCodec, [470](#)
- Join
 - gdcm::Filename, [374](#)
- JunkAfterDocElementError
 - gdcm::Parser, [528](#)
- KAKADUCodec
 - gdcm::KAKADUCodec, [473](#)
- KeyObjectSelectionDocument
 - gdcm::MediaStorage, [489](#)
- KeyObjectSelectionDocumentStorage
 - gdcm::UIDs, [747](#)
- KeyField
 - gdcm::CSAElement, [257](#)
- KeyValuePairArrayType
 - gdcm::CompositeNetworkFunctions, [247](#)
- KeyValuePairType
 - gdcm::CompositeNetworkFunctions, [247](#)
- LD_ALL
 - gdcm, [119](#)
- LD_NOSEQ
 - gdcm, [119](#)
- LD_NOSHADOW
 - gdcm, [119](#)
- LD_NOSHADOWSEQ
 - gdcm, [119](#)
- LINE
 - gdcm::MeshPrimitive, [498](#)
- LO
 - gdcm::VR, [827](#)
- LT
 - gdcm::VR, [827](#)
- LO
 - gdcm::LO, [475](#)
- LOComp
 - gdcm, [118](#)
- LTComp
 - gdcm, [118](#)
- LUT

- gdcm::Bitmap, [212](#)
- gdcm::ImageCodec, [418](#)
- LUTPtr
 - gdcm::Bitmap, [208](#)
 - gdcm::ImageCodec, [415](#)
- LeadECGWaveformStorage
 - gdcm::MediaStorage, [489](#)
- Level
 - vtkImageMapToWindowLevelColors2, [871](#)
- ListCharSets
 - gdcm::QueryFactory, [588](#)
- LittleEndian
 - gdcm::SwapCode, [701](#)
- Load
 - gdcm::Directory, [320](#)
- LoadDefault
 - gdcm::CSAHeaderDict, [264](#)
 - gdcm::Dict, [305](#)
 - gdcm::PrivateDict, [578](#)
- LoadDefaults
 - gdcm::Defs, [297](#)
 - gdcm::Dicts, [314](#)
- LoadFromDataElement
 - gdcm::CSAHeader, [261](#)
 - gdcm::PDBHeader, [535](#)
- LoadFromFile
 - gdcm::Defs, [297](#)
- LoadIconImage
 - vtkGDCMImageReader, [838](#)
- LoadImageFromFiles
 - gdcm::DirectoryHelper, [322](#)
- LoadOverlays
 - vtkGDCMImageReader, [838](#)
- LoadResourcesFiles
 - gdcm::Global, [389](#)
- LoadSingleFile
 - vtkGDCMImageReader, [836](#)
- Locate
 - gdcm::Global, [389](#)
- LodModeType
 - gdcm, [119](#)
- LookupTable
 - gdcm::LookupTable, [478](#)
 - vtkImageMapToColors16, [868](#)
- LookupTableType
 - gdcm::LookupTable, [478](#)
- Lossless
 - gdcm::JPEGCodec, [468](#)
- LossyFlag
 - gdcm::Bitmap, [212](#)
 - gdcm::ImageCodec, [418](#)
 - vtkGDCMImageReader, [838](#)
- MAGNIFIED
 - gdcm::Spacing, [666](#)
- MANUAL
 - gdcm::Segment, [623](#)
- MONOCHROME1
 - gdcm::PhotometricInterpretation, [543](#)
- MONOCHROME2
 - gdcm::PhotometricInterpretation, [543](#)
- MPEG2MainProfile
 - gdcm::TransferSyntax, [729](#)
- MPEG2MainProfileMainLevel
 - gdcm::UIDs, [743](#)
- MPTType_END
 - gdcm::MeshPrimitive, [498](#)
- MRImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- MRSpectroscopyStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- MS_END
 - gdcm::MediaStorage, [490](#)
- m_ConstMemberFunction
 - gdcm::MemberCommand, [496](#)
- m_MemberFunction
 - gdcm::MemberCommand, [496](#)
 - gdcm::SimpleMemberCommand, [655](#)
- m_This
 - gdcm::MemberCommand, [496](#)
 - gdcm::SimpleMemberCommand, [655](#)
- m_char
 - gdcm::ignore_char, [396](#)
- mAction
 - gdcm::network::Transition, [733](#)
- MD5
 - gdcm::MD5, [485](#)
- MD5DataImagesType
 - gdcm::Testing, [721](#)
- MD5MetalImagesType
 - vtkGDCMTesting, [853](#)
- mDataSet
 - gdcm::BaseRootQuery, [200](#)
- mElementOffsets
 - gdcm::StreamImageWriter, [678](#)
- mElementOffsets1
 - gdcm::StreamImageWriter, [678](#)
- mEnd
 - gdcm::network::Transition, [733](#)
- mHelpDescription
 - gdcm::BaseRootQuery, [200](#)
- mImage
 - gdcm::BaseRootQuery, [200](#)
- MPTType
 - gdcm::MeshPrimitive, [498](#)
- mPatient

- gdcmm::BaseRootQuery, 200
- mRootType
 - gdcmm::BaseRootQuery, 200
- MSType
 - gdcmm::MediaStorage, 488
- mSeries
 - gdcmm::BaseRootQuery, 200
- mStudy
 - gdcmm::BaseRootQuery, 200
- mWriter
 - gdcmm::StreamImageWriter, 678
- mXMax
 - gdcmm::StreamImageWriter, 678
- mXMin
 - gdcmm::StreamImageWriter, 678
- mYMax
 - gdcmm::StreamImageWriter, 678
- mYMin
 - gdcmm::StreamImageWriter, 678
- mZMax
 - gdcmm::StreamImageWriter, 678
- mZMin
 - gdcmm::StreamImageWriter, 678
- Macro
 - gdcmm::Macro, 481
- MacroEntry
 - gdcmm, 118
- Macros
 - gdcmm::Macros, 483
- magenta
 - gdcmm::terminal, 131
- MakeDirectory
 - gdcmm::System, 706
- MakeNew
 - gdcmm::network::Transition, 733
- MakeObject
 - gdcmm::AnonymizeEvent, 147
 - gdcmm::DataEvent, 284
 - gdcmm::DataSetEvent, 293
 - gdcmm::Event, 349
 - gdcmm::ProgressEvent, 582
- MammographyCADSR
 - gdcmm::MediaStorage, 489
- MammographyCADSRStorage
 - gdcmm::UIDs, 746
- Mandatory
 - gdcmm::Usage, 810
- MapCSAHeaderDictEntry
 - gdcmm::CSAHeaderDict, 263
- MapDictEntry
 - gdcmm::Dict, 304
- MapIODEntry
 - gdcmm::IOD, 442
- MapModuleEntry
 - gdcmm::Macro, 481
 - gdcmm::Module, 502
- MapScalarsThroughTable2
 - vtkLookupTable16, 878
- MapTableEntry
 - gdcmm::Table, 708
- MappingType
 - gdcmm::Scanner, 617
- MaxLength
 - gdcmm::ApplicationEntity, 156
 - gdcmm::PersonName, 540
- MaxNumberOfComponents
 - gdcmm::ApplicationEntity, 156
 - gdcmm::PersonName, 540
- MaxPrintLength
 - gdcmm::Printer, 577
- MaximumLengthSub
 - gdcmm::network::MaximumLengthSub, 484
- MediaCreationManagementSOPClassUID
 - gdcmm::UIDs, 745
- MediaStorageDirectoryStorage
 - gdcmm::MediaStorage, 488
 - gdcmm::UIDs, 743
- MediaStorage
 - gdcmm::MediaStorage, 491
- MediaStorageDataFilesType
 - gdcmm::Testing, 721
- MedicalImageProperties
 - vtkGDCMImageReader, 838
 - vtkGDCMPolyDataReader, 848
 - vtkGDCMPolyDataWriter, 851
- MemberCommand
 - gdcmm::MemberCommand, 495
- MeshPrimitive
 - gdcmm::MeshPrimitive, 499
- MessageID
 - gdcmm::network::CEchoRQ, 227
- MetaInformationTS
 - gdcmm::FileMetaInformation, 372
- ModalityPerformedProcedureStepNotificationSOPClass
 - gdcmm::UIDs, 744
- ModalityPerformedProcedureStepRetrieveSOPClass
 - gdcmm::UIDs, 744
- ModalityPerformedProcedureStepSOPClass
 - gdcmm::MediaStorage, 490
 - gdcmm::UIDs, 744
- ModalityWorklistInformationModelIFIND
 - gdcmm::UIDs, 747
- Mode
 - gdcmm::terminal, 131
- Module
 - gdcmm::Module, 502
- ModuleEntry
 - gdcmm::ModuleEntry, 504

- ModuleMapType
 - gdcm::Macros, [483](#)
 - gdcm::Modules, [506](#)
- Modules
 - gdcm::Modules, [506](#)
- MovePatientRootQuery
 - gdcm::MovePatientRootQuery, [508](#)
- MoveStudyRootQuery
 - gdcm::MoveStudyRootQuery, [510](#)
- mSPFile
 - gdcm::StreamImageWriter, [678](#)
- MultiframeGrayscaleByteSecondaryCaptureImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- MultiframeGrayscaleWordSecondaryCaptureImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- MultiframeSingleBitSecondaryCaptureImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- MultiframeTrueColorSecondaryCaptureImageStorage
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [745](#)
- N_ACTION_RQ
 - gdcm::network::DIMSE, [316](#)
- N_ACTION_RSP
 - gdcm::network::DIMSE, [316](#)
- N_CREATE_RQ
 - gdcm::network::DIMSE, [316](#)
- N_CREATE_RSP
 - gdcm::network::DIMSE, [316](#)
- N_DELETE_RQ
 - gdcm::network::DIMSE, [316](#)
- N_DELETE_RSP
 - gdcm::network::DIMSE, [316](#)
- N_EVENT_REPORT_RQ
 - gdcm::network::DIMSE, [315](#)
- N_EVENT_REPORT_RSP
 - gdcm::network::DIMSE, [315](#)
- N_GET_RQ
 - gdcm::network::DIMSE, [315](#)
- N_GET_RSP
 - gdcm::network::DIMSE, [316](#)
- N_SET_RQ
 - gdcm::network::DIMSE, [316](#)
- N_SET_RSP
 - gdcm::network::DIMSE, [316](#)
- NO
 - gdcm::Surface, [690](#)
- NO_COMPRESSION
 - vtkGDCMImageWriter, [841](#)
- NOMAGIC
 - gdcm::CSAHeader, [260](#)
- Name
 - gdcm::ModuleEntry, [505](#)
- NameField
 - gdcm::CSAElement, [258](#)
 - gdcm::PDBelement, [533](#)
- NeedByteSwap
 - gdcm::Bitmap, [212](#)
 - gdcm::ImageCodec, [418](#)
- NeedOverlayCleanup
 - gdcm::ImageCodec, [418](#)
- NegotiatedType
 - gdcm::TransferSyntax, [729](#)
- NestedMacroEntries
 - gdcm, [118](#)
- NestedModuleEntries
 - gdcm::NestedModuleEntries, [513](#)
- New
 - gdcm::Anonymizer, [151](#)
 - gdcm::MemberCommand, [495](#)
 - gdcm::Scanner, [620](#)
 - gdcm::SequenceOfFragments, [635](#)
 - gdcm::SequenceOfItems, [640](#)
 - gdcm::SimpleMemberCommand, [655](#)
 - vtkGDCMImageReader, [836](#)
 - vtkGDCMImageWriter, [841](#)
 - vtkGDCMMedicalImageProperties, [845](#)
 - vtkGDCMPolyDataReader, [847](#)
 - vtkGDCMPolyDataWriter, [850](#)
 - vtkGDCMTesting, [853](#)
 - vtkGDCMThreadedImageReader, [855](#)
 - vtkGDCMThreadedImageReader2, [857](#)
 - vtkImageColorViewer, [863](#)
 - vtkImageMapToColors16, [867](#)
 - vtkImageMapToWindowLevelColors2, [870](#)
 - vtkImagePlanarComponentsToComponents, [872](#)
 - vtkImageRGBToYBR, [874](#)
 - vtkImageYBRToRGB, [876](#)
 - vtkLookupTable16, [878](#)
 - vtkRTStructSetProperties, [881](#)
- NoElementsError
 - gdcm::Parser, [528](#)
- NoError
 - gdcm::Parser, [528](#)
- NoMemoryError
 - gdcm::Parser, [528](#)
- NoObject
 - gdcm::MediaStorage, [490](#)
- NoOfItemsField
 - gdcm::CSAElement, [258](#)
- Normalize
 - gdcm::DirectionCosines, [317](#)
- NuclearMedicineImageStorage

- gdcmm::MediaStorage, 489
- gdcmm::UIDs, 746
- NuclearMedicineImageStorageRetired
 - gdcmm::MediaStorage, 488
 - gdcmm::UIDs, 745
- NumberOfDimensions
 - gdcmm::Bitmap, 212
 - gdcmm::ImageCodec, 418
- NumberOfIconImages
 - vtkGDCMImageReader, 838
- NumberOfOverlays
 - vtkGDCMImageReader, 838
- NumberOfSurfaces
 - gdcmm::SurfaceWriter, 700
- OB
 - gdcmm::VR, 827
- OB_OW
 - gdcmm::VR, 827
- OBLIQUE
 - gdcmm::Orientation, 518
- OF
 - gdcmm::VR, 827
- OW
 - gdcmm::VR, 827
- Object
 - gdcmm::Object, 516
- ObjectEnd
 - gdcmm::MediaStorage, 490
- ObjectType
 - gdcmm::MediaStorage, 490
- Ofstream
 - gdcmm::Writer, 888
- op
 - gdcmm::SerieHelper::Rule, 614
- operator const char *
 - gdcmm::ConstCharWrapper, 250
 - gdcmm::Filename, 374
 - gdcmm::String, 681
- operator const double *
 - gdcmm::DirectionCosines, 317
- operator const std::vector< char > &
 - gdcmm::ByteValue, 224
- operator MStype
 - gdcmm::MediaStorage, 492
- operator ObjectType *
 - gdcmm::SmartPointer, 659
- operator PType
 - gdcmm::PhotometricInterpretation, 544
- operator ScalarType
 - gdcmm::PixelFormat, 548
- operator SwapCode::SwapCodeType
 - gdcmm::SwapCode, 702
- operator TStype
 - gdcmm::TransferSyntax, 730
 - gdcmm::UIDs, 756
- operator TypeType
 - gdcmm::Type, 735
- operator uint32_t
 - gdcmm::VL, 819
- operator UsageType
 - gdcmm::Usage, 810
- operator VMType
 - gdcmm::VM, 824
- operator VRType
 - gdcmm::VR, 828
- operator<
 - gdcmm::Attribute, 167
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 173
 - gdcmm::CSAElement, 257
 - gdcmm::CSAHeaderDictEntry, 265
 - gdcmm::DataElement, 278
 - gdcmm::PrivateTag, 580
 - gdcmm::Tag, 716
- operator<<
 - gdcmm, 120–123
 - gdcmm::BasicOffsetTable, 205
 - gdcmm::CodeString, 241
 - gdcmm::CommandDataSet, 245
 - gdcmm::CSAElement, 257
 - gdcmm::CSAHeader, 262
 - gdcmm::CSAHeaderDict, 264
 - gdcmm::CSAHeaderDictEntry, 266
 - gdcmm::DataElement, 281
 - gdcmm::DataSet, 291
 - gdcmm::Dict, 305
 - gdcmm::DictEntry, 310
 - gdcmm::Dicts, 314
 - gdcmm::Directory, 320
 - gdcmm::File, 360
 - gdcmm::FileMetaInformation, 372
 - gdcmm::FileSet, 378
 - gdcmm::Fragment, 387
 - gdcmm::Global, 389
 - gdcmm::GroupDict, 391
 - gdcmm::IOD, 443
 - gdcmm::IODEntry, 444
 - gdcmm::IODs, 446
 - gdcmm::Item, 453
 - gdcmm::Macro, 482
 - gdcmm::Macros, 483
 - gdcmm::MediaStorage, 492
 - gdcmm::Module, 502
 - gdcmm::ModuleEntry, 505
 - gdcmm::Modules, 507
 - gdcmm::NestedModuleEntries, 513
 - gdcmm::Object, 516

- gdcmm::Orientation, 518
- gdcmm::PDBelement, 533
- gdcmm::PDBHeader, 535
- gdcmm::PhotometricInterpretation, 544
- gdcmm::PixelFormat, 549
- gdcmm::Preamble, 565
- gdcmm::PrivateDict, 578
- gdcmm::PrivateTag, 580
- gdcmm::Scanner, 620
- gdcmm::Sorter, 665
- gdcmm::SwapCode, 702
- gdcmm::Table, 708
- gdcmm::Tag, 718
- gdcmm::TransferSyntax, 730
- gdcmm::Type, 735
- gdcmm::UI, 735
- gdcmm::Usage, 810
- gdcmm::Version, 818
- gdcmm::VL, 820
- gdcmm::VM, 824
- gdcmm::VR, 829
- operator<=
 - gdcmm::Tag, 716
- operator>>
 - gdcmm, 124
 - gdcmm::Tag, 718
- operator*
 - gdcmm::SmartPointer, 659
- operator()
 - gdcmm::DataSet, 290
 - gdcmm::Scanner::Itstr, 480
- operator++
 - gdcmm::VL, 819
- operator+=
 - gdcmm::VL, 819
- operator->
 - gdcmm::SmartPointer, 659
- operator=
 - gdcmm::BoxRegion, 218
 - gdcmm::ByteValue, 224
 - gdcmm::CSAElement, 257
 - gdcmm::DataElement, 278
 - gdcmm::DataSet, 290
 - gdcmm::Element< TVR, VM::VM1_n >, 331
 - gdcmm::network::UserInfo, 812
 - gdcmm::Object, 516
 - gdcmm::ParseException, 526
 - gdcmm::Preamble, 565
 - gdcmm::SequenceOfItems, 640
 - gdcmm::SmartPointer, 659
 - gdcmm::Tag, 716
- operator==
 - gdcmm, 123
 - gdcmm::Attribute, 168
 - gdcmm::Attribute< Group, Element, TVR, VM::VM1 >, 173
 - gdcmm::ByteValue, 224
 - gdcmm::CodeString, 241
 - gdcmm::CSAElement, 257
 - gdcmm::DataElement, 279
 - gdcmm::network::AbstractSyntax, 145
 - gdcmm::network::PresentationContextRQ, 572
 - gdcmm::network::TransferSyntaxSub, 731
 - gdcmm::PDBelement, 533
 - gdcmm::PixelFormat, 548
 - gdcmm::PresentationContext, 567
 - gdcmm::SequenceOfFragments, 635
 - gdcmm::SequenceOfItems, 640
 - gdcmm::Tag, 716
 - gdcmm::Value, 816
- OphthalmicPhotography16BitImageStorage
 - gdcmm::UIDs, 746
- OphthalmicPhotography8BitImageStorage
 - gdcmm::MediaStorage, 490
 - gdcmm::UIDs, 746
- OphthalmicTomographyImageStorage
 - gdcmm::MediaStorage, 490
 - gdcmm::UIDs, 746
- OrderFileList
 - gdcmm::SerieHelper, 644
- Orientation
 - gdcmm::Orientation, 518
- OrientationType
 - gdcmm::Orientation, 518
- Output
 - gdcmm::BitmapToBitmapFilter, 215
- OutputFormat
 - vtkImageMapToColors16, 868
- OutputTypes
 - gdcmm::DictConverter, 306
- Overlay
 - gdcmm::Overlay, 522
- OverlayImageActor
 - vtkImageColorViewer, 865
- OverlayType
 - gdcmm::Overlay, 521
- Overlays
 - gdcmm::Pixmap, 552
- PALETTE_COLOR
 - gdcmm::PhotometricInterpretation, 543
- PDF
 - gdcmm::MediaStorage, 490
- PETImageStorage
 - gdcmm::MediaStorage, 489
- PHILIPS
 - gdcmm::Dicts, 313
- PI_END

- gdcmm::PhotometricInterpretation, 543
- PN
 - gdcmm::VR, 827
- POINTS
 - gdcmm::Surface, 690
- PDBElement
 - gdcmm::PDBElement, 533
- PDBHeader
 - gdcmm::PDBHeader, 535
- PDFCodec
 - gdcmm::PDFCodec, 537
- PDataTFPDU
 - gdcmm::network::PDataTFPDU, 531
- PF
 - gdcmm::Bitmap, 212
 - gdcmm::ImageCodec, 418
- PGXCodec
 - gdcmm::PGXCodec, 541
- PI
 - gdcmm::Bitmap, 212
 - gdcmm::ImageCodec, 418
- PType
 - gdcmm::PhotometricInterpretation, 543
- PNComp
 - gdcmm, 118
- PNMCodec
 - gdcmm::PNMCodec, 563
- PVRGCodec
 - gdcmm::PVRGCodec, 584
- Pack
 - gdcmm::Unpacker12Bits, 808
- Padding
 - gdcmm::ApplicationEntity, 156
 - gdcmm::PersonName, 540
- Parent
 - gdcmm::Element< TVR, VM::VM1_2 >, 329
 - gdcmm::Element< TVR, VM::VM2_2n >, 334
 - gdcmm::Element< TVR, VM::VM2_n >, 335
 - gdcmm::Element< TVR, VM::VM3_3n >, 337
 - gdcmm::Element< TVR, VM::VM3_n >, 338
- Parse
 - gdcmm::Parser, 528
- ParseBuffer
 - gdcmm::Parser, 529
- ParseCertificateFile
 - gdcmm::CryptographicMessageSyntax, 253
- ParseDateTime
 - gdcmm::System, 706, 707
- ParseDump
 - gdcmm::ASN1, 162
- ParseDumpFile
 - gdcmm::ASN1, 162
- ParseException
 - gdcmm::ParseException, 526
- ParseKeyFile
 - gdcmm::CryptographicMessageSyntax, 253
- Parser
 - gdcmm::Parser, 528
- PassAlphaToOutput
 - vtkImageMapToColors16, 868
- Patient
 - gdcmm::Patient, 529
- PatientRootQueryRetrieveInformationModelFIND
 - gdcmm::UIDs, 747
- PatientRootQueryRetrieveInformationModelGET
 - gdcmm::UIDs, 747
- PatientRootQueryRetrieveInformationModelMOVE
 - gdcmm::UIDs, 747
- PatientStudyOnlyQueryRetrieveInformationModelFIND-Retired
 - gdcmm::UIDs, 747
- PatientStudyOnlyQueryRetrieveInformationModelGET-Retired
 - gdcmm::UIDs, 747
- PatientStudyOnlyQueryRetrieveInformationModelMOVE-Retired
 - gdcmm::UIDs, 747
- PerformAction
 - gdcmm::network::ULAction, 758
 - gdcmm::network::ULActionAA1, 759
 - gdcmm::network::ULActionAA2, 760
 - gdcmm::network::ULActionAA3, 762
 - gdcmm::network::ULActionAA4, 763
 - gdcmm::network::ULActionAA5, 764
 - gdcmm::network::ULActionAA6, 765
 - gdcmm::network::ULActionAA7, 766
 - gdcmm::network::ULActionAA8, 767
 - gdcmm::network::ULActionAE1, 769
 - gdcmm::network::ULActionAE2, 770
 - gdcmm::network::ULActionAE3, 771
 - gdcmm::network::ULActionAE4, 772
 - gdcmm::network::ULActionAE5, 773
 - gdcmm::network::ULActionAE6, 774
 - gdcmm::network::ULActionAE7, 776
 - gdcmm::network::ULActionAE8, 777
 - gdcmm::network::ULActionAR1, 778
 - gdcmm::network::ULActionAR10, 779
 - gdcmm::network::ULActionAR2, 780
 - gdcmm::network::ULActionAR3, 781
 - gdcmm::network::ULActionAR4, 783
 - gdcmm::network::ULActionAR5, 784
 - gdcmm::network::ULActionAR6, 785
 - gdcmm::network::ULActionAR7, 786
 - gdcmm::network::ULActionAR8, 787
 - gdcmm::network::ULActionAR9, 788
 - gdcmm::network::ULActionDT1, 790
 - gdcmm::network::ULActionDT2, 791
- Philips3D

- gdcm::MediaStorage, [489](#)
- PhilipsPrivateMRSyntheticImageStorage
 - gdcm::MediaStorage, [490](#)
- PhotometricInterpretation
 - gdcm::PhotometricInterpretation, [543](#)
- PixelData
 - gdcm::Bitmap, [212](#)
 - gdcm::PixmapReader, [556](#)
 - gdcm::PixmapWriter, [561](#)
- PixelFormat
 - gdcm::PixelFormat, [546](#)
- Pixmap
 - gdcm::Pixmap, [551](#)
- PixmapReader
 - gdcm::Bitmap, [212](#)
 - gdcm::PixmapReader, [555](#)
- PixmapToPixmapFilter
 - gdcm::PixmapToPixmapFilter, [557](#)
- PixmapWriter
 - gdcm::PixmapWriter, [560](#)
- PlanarConfiguration
 - gdcm::Bitmap, [212](#)
 - gdcm::ImageCodec, [418](#)
 - vtkGDCMImageReader, [838](#)
- pointer
 - gdcm::CodeString, [240](#)
 - gdcm::LO, [475](#)
 - gdcm::String, [681](#)
- PositronEmissionTomographyImageStorage
 - gdcm::UIDs, [747](#)
- Preamble
 - gdcm::Preamble, [565](#)
- PrepareWrite
 - gdcm::PixmapWriter, [561](#)
 - gdcm::SegmentWriter, [631](#)
 - gdcm::SurfaceWriter, [700](#)
- PrepareWritePointMacro
 - gdcm::SurfaceWriter, [700](#)
- Prepend
 - gdcm::Global, [389](#)
- PresentationLUTSOPClass
 - gdcm::UIDs, [745](#)
- PresentationContext
 - gdcm::PresentationContext, [566](#)
- PresentationContextAC
 - gdcm::network::PresentationContextAC, [568](#)
- PresentationContextArrayType
 - gdcm::network::AAssociateRQPDU, [141](#)
 - gdcm::PresentationContextGenerator, [569](#)
- PresentationContextGenerator
 - gdcm::PresentationContextGenerator, [569](#)
- PresentationContextRQ
 - gdcm::network::PresentationContextRQ, [571](#)
- PresentationDataValue
 - gdcm::network::PresentationDataValue, [573](#)
- PrimitiveData
 - gdcm::MeshPrimitive, [499](#)
- PrimitiveType
 - gdcm::MeshPrimitive, [499](#)
- PrimitivesData
 - gdcm::MeshPrimitive, [498](#)
- Print
 - gdcm::ApplicationEntity, [156](#)
 - gdcm::Attribute, [168](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [174](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [180](#)
 - gdcm::BaseRootQuery, [199](#)
 - gdcm::Bitmap, [210](#)
 - gdcm::BoxRegion, [218](#)
 - gdcm::ByteValue, [224](#)
 - gdcm::CSAHeader, [262](#)
 - gdcm::Curve, [271](#)
 - gdcm::DataSet, [290](#)
 - gdcm::DictPrinter, [312](#)
 - gdcm::DirectionCosines, [317](#)
 - gdcm::Directory, [320](#)
 - gdcm::Element, [327](#)
 - gdcm::Element< TVR, VM::VM1_n >, [331](#)
 - gdcm::Element< VR::AS, VM::VM5 >, [339](#)
 - gdcm::Event, [349](#)
 - gdcm::Image, [399](#)
 - gdcm::LookupTable, [479](#)
 - gdcm::network::AAAbortPDU, [134](#)
 - gdcm::network::AAssociateACPDU, [137](#)
 - gdcm::network::AAssociateRJPDPU, [139](#)
 - gdcm::network::AAssociateRQPDU, [142](#)
 - gdcm::network::AbstractSyntax, [145](#)
 - gdcm::network::ApplicationContext, [155](#)
 - gdcm::network::AReleaseRPPDU, [158](#)
 - gdcm::network::AReleaseRQPDU, [160](#)
 - gdcm::network::AsynchronousOperationsWindow-Sub, [163](#)
 - gdcm::network::BasePDU, [196](#)
 - gdcm::network::ImplementationClassUIDSub, [436](#)
 - gdcm::network::ImplementationVersionNameSub, [437](#)
 - gdcm::network::MaximumLengthSub, [484](#)
 - gdcm::network::PDataTFPDU, [531](#)
 - gdcm::network::PresentationContextAC, [568](#)
 - gdcm::network::PresentationContextRQ, [572](#)
 - gdcm::network::PresentationDataValue, [573](#)
 - gdcm::network::RoleSelectionSub, [613](#)
 - gdcm::network::ServiceClassApplicationInformation, [645](#)
 - gdcm::network::SOPClassExtendedNegociationSub, [660](#)

- gdcm::network::TransferSyntaxSub, 731
- gdcm::network::UserInformation, 812
- gdcm::Object, 516
- gdcm::Orientation, 518
- gdcm::Overlay, 523
- gdcm::PDBHeader, 535
- gdcm::PersonName, 539
- gdcm::PixelFormat, 548
- gdcm::Pixmap, 552
- gdcm::Preamble, 565
- gdcm::PresentationContext, 567
- gdcm::Printer, 576
- gdcm::Region, 606
- gdcm::Scanner, 620
- gdcm::SegmentedPaletteColorLookupTable, 626
- gdcm::SequenceOfFragments, 635
- gdcm::SequenceOfItems, 640
- gdcm::Sorter, 664
- gdcm::TagPath, 719
- gdcm::Testing, 723
- gdcm::Version, 817
- PrintJobSOPClass
 - gdcm::UIDs, 744
- PrintQueueManagementSOPClassRetired
 - gdcm::UIDs, 745
- PrintQueueSOPInstanceRetired
 - gdcm::UIDs, 745
- PrintASCII
 - gdcm::ByteValue, 224
- PrintAsPipeSeparatedString
 - gdcm::Tag, 717
- PrintDataElement
 - gdcm::Printer, 576
- PrintDataElement2
 - gdcm::DictPrinter, 312
- PrintDataSet
 - gdcm::Printer, 576
- PrintDataSet2
 - gdcm::DictPrinter, 312
- PrintGroupLength
 - gdcm::ByteValue, 225
- PrintHex
 - gdcm::ByteValue, 225
- PrintSQ
 - gdcm::Printer, 576
- PrintSelf
 - vtkGDCMImageReader, 836
 - vtkGDCMImageWriter, 841
 - vtkGDCMMedicalImageProperties, 845
 - vtkGDCMPolyDataReader, 847
 - vtkGDCMPolyDataWriter, 850
 - vtkGDCMTesting, 853
 - vtkGDCMThreadedImageReader, 855
 - vtkGDCMThreadedImageReader2, 857
 - vtkImageColorViewer, 863
 - vtkImageMapToColors16, 867
 - vtkImageMapToWindowLevelColors2, 870
 - vtkImagePlanarComponentsToComponents, 872
 - vtkImageRGBToYBR, 874
 - vtkImageYBRToRGB, 876
 - vtkLookupTable16, 878
 - vtkRTStructSetProperties, 882
- PrintStyle
 - gdcm::Printer, 577
- PrintStyles
 - gdcm::Printer, 576
- PrintTable
 - gdcm::network::ULTransitionTable, 802
- PrintXML
 - gdcm::PrivateDict, 578
- Printer
 - gdcm::Printer, 576
- PrinterConfigurationRetrievalSOPClass
 - gdcm::UIDs, 744
- PrinterConfigurationRetrievalSOPInstance
 - gdcm::UIDs, 744
- PrinterSOPClass
 - gdcm::UIDs, 744
- PrinterSOPInstance
 - gdcm::UIDs, 744
- PrivateDict
 - gdcm::PrivateDict, 578
- PrivateTag
 - gdcm::PrivateTag, 580
- ProceduralEventLoggingSOPClass
 - gdcm::UIDs, 744
- ProceduralEventLoggingSOPInstance
 - gdcm::UIDs, 744
- ProcedureLogStorage
 - gdcm::UIDs, 746
- Process
 - gdcm::Parser, 529
- ProcessDataSet
 - gdcm::FileExplicitFilter, 367
- ProcessPublicTag
 - gdcm::Scanner, 620
- ProduceCharacterSetDataElement
 - gdcm::QueryFactory, 588
- ProduceQuery
 - gdcm::QueryFactory, 589
- ProductCharacteristicsQuerySOPClass
 - gdcm::UIDs, 748
- ProgressEvent
 - gdcm::ProgressEvent, 582
- PropertyCategory
 - gdcm::Segment, 624
- PropertyType
 - gdcm::Segment, 624

- PseudoColorSoftcopyPresentationStateStorageSOP-
Class
gdcmm::UIDs, [746](#)
- PullPrintRequestSOPClassRetired
gdcmm::UIDs, [745](#)
- PullStoredPrintManagementMetaSOPClassRetired
gdcmm::UIDs, [745](#)
- Push
gdcmm::TagPath, [720](#)
- PushBackFile
vtkGDCMMedicalImageProperties, [845](#)
- PythonFilter
gdcmm::PythonFilter, [585](#)
- Quality
gdcmm::JPEGCodec, [468](#)
- QueryFactory
gdcmm::BaseRootQuery, [200](#)
gdcmm::FindPatientRootQuery, [382](#)
gdcmm::FindStudyRootQuery, [384](#)
gdcmm::MovePatientRootQuery, [509](#)
gdcmm::MoveStudyRootQuery, [511](#)
- RED
gdcmm::LookupTable, [478](#)
- RFC2557MIMEencapsulation
gdcmm::UIDs, [743](#)
- RGB
gdcmm::PhotometricInterpretation, [543](#)
- RLE_COMPRESSION
vtkGDCMImageWriter, [841](#)
- RLELossless
gdcmm::TransferSyntax, [729](#)
gdcmm::UIDs, [743](#)
- ROI
gdcmm::Overlay, [521](#)
- RTBeamsDeliveryInstructionStorageSupplement74-
FrozenDraft
gdcmm::UIDs, [747](#)
- RTBeamsTreatmentRecordStorage
gdcmm::UIDs, [747](#)
- RTBrachyTreatmentRecordStorage
gdcmm::UIDs, [747](#)
- RTConventionalMachineVerificationSupplement74Frozen-
Draft
gdcmm::UIDs, [747](#)
- RTDoseStorage
gdcmm::MediaStorage, [489](#)
gdcmm::UIDs, [747](#)
- RTImageStorage
gdcmm::MediaStorage, [489](#)
gdcmm::UIDs, [747](#)
- RTIonBeamsTreatmentRecordStorage
gdcmm::MediaStorage, [490](#)
gdcmm::UIDs, [747](#)
- RTIonMachineVerificationSupplement74FrozenDraft
gdcmm::UIDs, [747](#)
- RTIonPlanStorage
gdcmm::MediaStorage, [490](#)
gdcmm::UIDs, [747](#)
- RTPlanStorage
gdcmm::MediaStorage, [489](#)
gdcmm::UIDs, [747](#)
- RTStructureSetStorage
gdcmm::MediaStorage, [489](#)
gdcmm::UIDs, [747](#)
- RTTreatmentSummaryRecordStorage
gdcmm::MediaStorage, [490](#)
gdcmm::UIDs, [747](#)
- RAWCodec
gdcmm::RAWCodec, [598](#)
- README.txt, [1154](#)
- RGB2YBR
gdcmm::ImageChangePhotometricInterpretation, [405](#)
- RGBPixelsToRGBPlanes
gdcmm::ImageChangePlanarConfiguration, [408](#)
- RGBPlanesToRGBPixels
gdcmm::ImageChangePlanarConfiguration, [408](#)
- RGBToRecommendedDisplayCIELab
gdcmm::SurfaceHelper, [695](#)
- RGBToRecommendedDisplayGrayscale
gdcmm::SurfaceHelper, [696](#)
- RLECodec
gdcmm::RLECodec, [611](#)
- RTStructSetProperties
vtkGDCMPolyDataReader, [848](#)
vtkGDCMPolyDataWriter, [851](#)
- RawDataStorage
gdcmm::MediaStorage, [489](#)
gdcmm::UIDs, [746](#)
- Read
gdcmm::BasicOffsetTable, [205](#)
gdcmm::ByteValue, [225](#)
gdcmm::CommandDataSet, [245](#)
gdcmm::CP246ExplicitDataElement, [251](#)
gdcmm::CSAHeader, [262](#)
gdcmm::DataElement, [279](#)
gdcmm::DataSet, [290](#)
gdcmm::Element, [327](#)
gdcmm::Element< TVR, VM::VM1_n >, [331](#)
gdcmm::EncodingImplementation< VR::VRASCII >, [344](#)
gdcmm::EncodingImplementation< VR::VRBINARY >, [345](#)
gdcmm::ExplicitDataElement, [354](#)
gdcmm::ExplicitImplicitDataElement, [356](#)
gdcmm::File, [360](#)
gdcmm::FileMetaInformation, [371](#)
gdcmm::Fragment, [386](#)

- gdcm::ImageReader, 427
- gdcm::ImageRegionReader, 430
- gdcm::ImplicitDataElement, 440
- gdcm::Item, 453
- gdcm::network::AAAbortPDU, 134
- gdcm::network::AAAssociateACPDU, 137
- gdcm::network::AAAssociateRJPDU, 139
- gdcm::network::AAAssociateRQPDU, 142
- gdcm::network::AbstractSyntax, 145
- gdcm::network::ApplicationContext, 155
- gdcm::network::AReleaseRPPDU, 158
- gdcm::network::AReleaseRQPDU, 160
- gdcm::network::AsynchronousOperationsWindow-Sub, 163
- gdcm::network::BasePDU, 196
- gdcm::network::ImplementationClassUIDSub, 436
- gdcm::network::ImplementationVersionNameSub, 437
- gdcm::network::MaximumLengthSub, 484
- gdcm::network::PDataTFPDU, 531
- gdcm::network::PresentationContextAC, 568
- gdcm::network::PresentationContextRQ, 572
- gdcm::network::PresentationDataValue, 573
- gdcm::network::RoleSelectionSub, 613
- gdcm::network::ServiceClassApplicationInformation, 645
- gdcm::network::SOPClassExtendedNegociationSub, 660
- gdcm::network::TransferSyntaxSub, 731
- gdcm::network::UserInformation, 812
- gdcm::PGXCodec, 542
- gdcm::PixmapReader, 555
- gdcm::PNMCodec, 563
- gdcm::Preamble, 565
- gdcm::Reader, 603
- gdcm::SegmentReader, 629
- gdcm::SequenceOfFragments, 635
- gdcm::SequenceOfItems, 640
- gdcm::StreamImageReader, 672
- gdcm::SurfaceReader, 698
- gdcm::TableReader, 711
- gdcm::Tag, 717
- gdcm::UNExplicitDataElement, 805
- gdcm::UNExplicitImplicitDataElement, 808
- gdcm::ValueIO, 816
- gdcm::VL, 819
- gdcm::VR, 828
- gdcm::VR16ExplicitDataElement, 830
- gdcm::VRVLSize< 0 >, 832
- gdcm::VRVLSize< 1 >, 832
- Read16
 - gdcm::VL, 820
- ReadACRNEAImage
 - gdcm::ImageReader, 428
 - gdcm::PixmapReader, 555
- ReadBacktrack
 - gdcm::Fragment, 386
- ReadCompat
 - gdcm::FileMetaInformation, 371
- ReadCompatInternal
 - gdcm::FileMetaInformation, 371
- ReadComputeLength
 - gdcm::EncodingImplementation< VR::VRASCII >, 344
 - gdcm::EncodingImplementation< VR::VRBINARY >, 345
- ReadDataSet
 - gdcm::Reader, 603
- ReadFiles
 - vtkGDCMThreadedImageReader, 855
- ReadFromCommaSeparatedString
 - gdcm::PrivateTag, 580
 - gdcm::Tag, 717
- ReadFromPipeSeparatedString
 - gdcm::Tag, 717
- ReadImage
 - gdcm::ImageReader, 428
 - gdcm::PixmapReader, 555
- ReadImageInformation
 - gdcm::StreamImageReader, 672
- ReadImageInternal
 - gdcm::PixmapReader, 555
- ReadInformation
 - gdcm::ImageRegionReader, 430
- ReadInto
 - gdcm::network::PDataTFPDU, 531
 - gdcm::network::PresentationDataValue, 573
- ReadIntoBuffer
 - gdcm::ImageRegionReader, 430
- ReadMetaInformation
 - gdcm::Reader, 603
- ReadNested
 - gdcm::DataSet, 290
- ReadNoSwap
 - gdcm::EncodingImplementation< VR::VRASCII >, 345
 - gdcm::EncodingImplementation< VR::VRBINARY >, 346
- ReadOrSkip
 - gdcm::DataElement, 279
- ReadPointMacro
 - gdcm::SurfaceReader, 698
- ReadPreValue
 - gdcm::CP246ExplicitDataElement, 252
 - gdcm::DataElement, 279
 - gdcm::ExplicitDataElement, 354
 - gdcm::ExplicitImplicitDataElement, 356
 - gdcm::Fragment, 386

- gdcm::ImplicitDataElement, [440](#)
- gdcm::SequenceOfFragments, [635](#)
- gdcm::UNExplicitDataElement, [806](#)
- gdcm::UNExplicitImplicitDataElement, [808](#)
- gdcm::VR16ExplicitDataElement, [831](#)
- ReadPreamble
 - gdcm::Reader, [603](#)
- ReadSegment
 - gdcm::SegmentReader, [629](#)
- ReadSegments
 - gdcm::SegmentReader, [629](#)
- ReadSelectedTags
 - gdcm::DataSet, [290](#)
 - gdcm::Reader, [603](#)
- ReadSelectedTagsWithLength
 - gdcm::DataSet, [290](#)
- ReadSurface
 - gdcm::SurfaceReader, [698](#)
- ReadSurfaces
 - gdcm::SurfaceReader, [698](#)
- ReadUpToTag
 - gdcm::DataSet, [290](#)
 - gdcm::Reader, [603](#)
- ReadUpToTagWithLength
 - gdcm::DataSet, [290](#)
- ReadVM
 - gdcm::DictConverter, [307](#)
- ReadVR
 - gdcm::DictConverter, [307](#)
- ReadValue
 - gdcm::CP246ExplicitDataElement, [252](#)
 - gdcm::DataElement, [279](#)
 - gdcm::ExplicitDataElement, [354](#)
 - gdcm::ExplicitImplicitDataElement, [356](#)
 - gdcm::Fragment, [386](#)
 - gdcm::ImplicitDataElement, [440](#)
 - gdcm::SequenceOfFragments, [635](#)
 - gdcm::UNExplicitDataElement, [806](#)
 - gdcm::UNExplicitImplicitDataElement, [808](#)
 - gdcm::VR16ExplicitDataElement, [831](#)
- ReadWithLength
 - gdcm::CP246ExplicitDataElement, [252](#)
 - gdcm::DataElement, [279](#)
 - gdcm::DataSet, [290](#)
 - gdcm::ExplicitDataElement, [354](#)
 - gdcm::ExplicitImplicitDataElement, [356](#)
 - gdcm::ImplicitDataElement, [440](#)
 - gdcm::UNExplicitDataElement, [806](#)
 - gdcm::VR16ExplicitDataElement, [831](#)
- Reader
 - gdcm::Reader, [602](#)
- Readuint16
 - gdcm::DictConverter, [307](#)
- RealWorldValueMappingStorage
 - gdcm::UIDs, [746](#)
- RecommendedDisplayCIELabToRGB
 - gdcm::SurfaceHelper, [694](#), [695](#)
- RecurseDataSet
 - gdcm::Anonymizer, [151](#)
- red
 - gdcm::terminal, [131](#)
- reference
 - gdcm::CodeString, [240](#)
 - gdcm::LO, [475](#)
 - gdcm::String, [681](#)
- ReferenceFrameOfReferenceUID
 - vtkRTStructSetProperties, [882](#)
- ReferenceSeriesInstanceUID
 - vtkRTStructSetProperties, [882](#)
- ReferencedColorPrintManagementMetaSOPClassRetired
 - gdcm::UIDs, [744](#)
- ReferencedGrayscalePrintManagementMetaSOPClassRetired
 - gdcm::UIDs, [744](#)
- ReferencedImageBoxSOPClassRetired
 - gdcm::UIDs, [744](#)
- Region
 - gdcm::Region, [605](#)
- Register
 - gdcm::Object, [516](#)
- Remove
 - gdcm::Anonymizer, [151](#)
 - gdcm::DataSet, [290](#)
 - gdcm::FileAnonymizer, [362](#)
 - gdcm::Preamble, [565](#)
- RemoveAllObservers
 - gdcm::Subject, [687](#)
- RemoveDictEntry
 - gdcm::PrivateDict, [578](#)
- RemoveFile
 - gdcm::System, [707](#)
- RemoveGroupLength
 - gdcm::Anonymizer, [151](#)
- RemoveObserver
 - gdcm::Subject, [687](#)
- RemoveOverlay
 - gdcm::Pixmap, [552](#)
- RemovePrivateTags
 - gdcm::Anonymizer, [151](#)
- RemoveRetired
 - gdcm::Anonymizer, [152](#)
- Render
 - vtkImageColorViewer, [863](#)
- RenderWindow
 - vtkImageColorViewer, [865](#)
- Renderer
 - vtkImageColorViewer, [865](#)
- Replace

- gdcmm::Anonymizer, [152](#)
- gdcmm::CommandDataSet, [245](#)
- gdcmm::DataSet, [290](#)
- gdcmm::FileAnonymizer, [362](#)
- gdcmm::FileMetaInformation, [371](#)
- ReplaceEmpty
 - gdcmm::DataSet, [290](#)
- RequestData
 - vtkGDCMPolyDataReader, [847](#)
 - vtkImageMapToColors16, [868](#)
 - vtkImageMapToWindowLevelColors2, [870](#)
 - vtkImagePlanarComponentsToComponents, [872](#)
- RequestData_HemodynamicWaveformStorage
 - vtkGDCMPolyDataReader, [847](#)
- RequestData_RTStructureSetStorage
 - vtkGDCMPolyDataReader, [847](#)
- RequestDataCompat
 - vtkGDCMImageReader, [836](#)
 - vtkGDCMThreadedImageReader, [855](#)
- RequestInformation
 - vtkGDCMPolyDataReader, [847](#)
 - vtkGDCMThreadedImageReader2, [858](#)
 - vtkImageMapToColors16, [868](#)
 - vtkImageMapToWindowLevelColors2, [870](#)
- RequestInformation_HemodynamicWaveformStorage
 - vtkGDCMPolyDataReader, [847](#)
- RequestInformation_RTStructureSetStorage
 - vtkGDCMPolyDataReader, [847](#)
- RequestInformationCompat
 - vtkGDCMImageReader, [836](#)
- RequestPaddedCompositePixelCode
 - gdcmm::ImageCodec, [418](#)
- RequestPlanarConfiguration
 - gdcmm::ImageCodec, [418](#)
- Rescale
 - gdcmm::Rescaler, [608](#)
- RescaleFunctionIntoBestFit
 - gdcmm::Rescaler, [608](#)
- Rescaler
 - gdcmm::Rescaler, [608](#)
- reset
 - gdcmm::terminal, [131](#)
- ResetHandledDataSet
 - gdcmm::network::ULConnectionCallback, [796](#)
- RetrieveSOPInstanceUIDFromIndex
 - gdcmm::DirectoryHelper, [322](#)
- RetrieveSOPInstanceUIDFromZPosition
 - gdcmm::DirectoryHelper, [322](#)
- reverse
 - gdcmm::terminal, [131](#)
- reverse_iterator
 - gdcmm::CodeString, [240](#)
 - gdcmm::LO, [475](#)
 - gdcmm::String, [681](#)
- RoleSelectionSub
 - gdcmm::network::RoleSelectionSub, [613](#)
- SAGITTAL
 - gdcmm::Orientation, [518](#)
- SH
 - gdcmm::VR, [827](#)
- SIEMENS
 - gdcmm::Dicts, [313](#)
- SINGLEBIT
 - gdcmm::PixelFormat, [546](#)
- SL
 - gdcmm::VR, [827](#)
- SLICE_ORIENTATION_XY
 - vtkImageColorViewer, [862](#)
- SLICE_ORIENTATION_XZ
 - vtkImageColorViewer, [862](#)
- SLICE_ORIENTATION_YZ
 - vtkImageColorViewer, [862](#)
- SPM2AVG152PDFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2AVG152T1FrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2AVG152T2FrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2AVG305T1FrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2BRAINMASKFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2CSFFFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2EPIFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2FILT1FrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2GRAYFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2PDFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2PETFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2SINGLESUBJT1FrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2SPECTFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2T1FrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2T2FrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2TRANSMFrameofReference
 - gdcmm::UIDs, [743](#)
- SPM2WHITEFrameofReference
 - gdcmm::UIDs, [743](#)
- SQ

- gdcmm::VR, [827](#)
- SS
 - gdcmm::VR, [827](#)
- ST
 - gdcmm::VR, [827](#)
- STATES_END
 - gdcmm::Surface, [690](#)
- SURFACE
 - gdcmm::Surface, [690](#)
- SV10
 - gdcmm::CSAHeader, [260](#)
- SHA1
 - gdcmm::SHA1, [651](#)
- SHComp
 - gdcmm, [118](#)
- SOPClassExtendedNegociationSub
 - gdcmm::network::SOPClassExtendedNegociationSub, [660](#)
- SOPInstanceUID
 - vtkRTStructSetProperties, [883](#)
- STATES
 - gdcmm::Surface, [690](#)
- STComp
 - gdcmm, [118](#)
- ScalarType
 - gdcmm::PixelFormat, [546](#)
- Scale
 - vtkGDCMImageReader, [838](#)
- Scan
 - gdcmm::Scanner, [620](#)
- Scanner
 - gdcmm::Scanner, [618](#)
- SecondaryCaptureImageStorage
 - gdcmm::MediaStorage, [488](#)
 - gdcmm::UIDs, [745](#)
- Segment
 - gdcmm::Segment, [623](#)
- SegmentAlgorithmName
 - gdcmm::Segment, [624](#)
- SegmentAlgorithmType
 - gdcmm::Segment, [624](#)
- SegmentDescription
 - gdcmm::Segment, [624](#)
- SegmentLabel
 - gdcmm::Segment, [624](#)
- SegmentMap
 - gdcmm::SegmentReader, [628](#)
- SegmentNumber
 - gdcmm::Segment, [624](#)
- SegmentReader
 - gdcmm::SegmentReader, [628](#)
- SegmentVector
 - gdcmm::SegmentReader, [628](#)
 - gdcmm::SegmentWriter, [631](#)
- SegmentWriter
 - gdcmm::SegmentWriter, [631](#)
- Segmentation
 - gdcmm::MediaStorage, [490](#)
- SegmentationStorage
 - gdcmm::MediaStorage, [490](#)
 - gdcmm::UIDs, [746](#)
- SegmentedPaletteColorLookupTable
 - gdcmm::SegmentedPaletteColorLookupTable, [626](#)
- Segments
 - gdcmm::SegmentReader, [629](#)
 - gdcmm::SegmentWriter, [631](#)
- Selection
 - gdcmm::Sorter, [665](#)
- SelectionMap
 - gdcmm::Sorter, [663](#)
- Self
 - gdcmm::AnonymizeEvent, [147](#)
 - gdcmm::DataEvent, [283](#)
 - gdcmm::DataSetEvent, [292](#)
 - gdcmm::MemberCommand, [494](#)
 - gdcmm::ProgressEvent, [582](#)
 - gdcmm::SimpleMemberCommand, [654](#)
- SendEcho
 - gdcmm::network::ULConnectionManager, [800](#)
 - gdcmm::ServiceClassUser, [649](#)
- SendFind
 - gdcmm::network::ULConnectionManager, [800](#)
 - gdcmm::ServiceClassUser, [649](#)
- SendMove
 - gdcmm::network::ULConnectionManager, [800](#)
 - gdcmm::ServiceClassUser, [649](#)
- SendStore
 - gdcmm::network::ULConnectionManager, [800](#)
 - gdcmm::ServiceClassUser, [649](#)
- Separator
 - gdcmm::ApplicationEntity, [156](#)
 - gdcmm::PersonName, [540](#)
- SequenceLengthField
 - gdcmm::SequenceOfItems, [641](#)
- SequenceOfFragments
 - gdcmm::SequenceOfFragments, [634](#)
- SequenceOfItems
 - gdcmm::SequenceOfItems, [639](#)
- SerieHelper
 - gdcmm::SerieHelper, [643](#)
- SerieRestrictions
 - gdcmm::SerieHelper, [643](#)
- Series
 - gdcmm::Series, [645](#)
- SeriesInstanceUID
 - vtkRTStructSetProperties, [883](#)
- ServiceClassApplicationInformation

- gdcm::network::ServiceClassApplicationInformation, 645
- ServiceClassUser
 - gdcm::ServiceClassUser, 648
- Set
 - gdcm::Attribute, 168
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 180
 - gdcm::Element, 328
 - gdcm::Element< TVR, VM::VM1_n >, 331
- SetAETitle
 - gdcm::ServiceClassUser, 649
- SetAbstractSyntax
 - gdcm::network::PresentationContextRQ, 572
 - gdcm::PresentationContext, 567
- SetAlgorithmFamily
 - gdcm::Surface, 692
- SetAlgorithmName
 - gdcm::Surface, 692
- SetAlgorithmVersion
 - gdcm::Surface, 692
- SetAnatomicRegion
 - gdcm::Segment, 624
- SetArray
 - gdcm::Element< TVR, VM::VM1_n >, 331
- SetAxisOfRotation
 - gdcm::Surface, 692
- SetBitPosition
 - gdcm::Overlay, 524
- SetBitSample
 - gdcm::JPEGCodec, 467
- SetBitsAllocated
 - gdcm::Overlay, 524
 - gdcm::PixelFormat, 548
- SetBitsStored
 - gdcm::PixelFormat, 548
- SetBlob
 - gdcm::ApplicationEntity, 156
 - gdcm::network::PresentationDataValue, 573
 - gdcm::PersonName, 539
- SetBlueLUT
 - gdcm::LookupTable, 479
- SetBufferLength
 - gdcm::JPEGLSCCodec, 471
 - gdcm::PNMCodec, 563
 - gdcm::RLECodec, 612
- SetByteSwapTag
 - gdcm::ByteSwapFilter, 221
- SetByteValue
 - gdcm::Attribute, 168
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
- gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 181
- gdcm::CSAElement, 257
- gdcm::DataElement, 279
- SetByteValueNoSwap
 - gdcm::Attribute, 168
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
- SetCallbackFunction
 - gdcm::MemberCommand, 495
 - gdcm::SimpleMemberCommand, 655
- SetCalledAETitle
 - gdcm::network::AAssociateACPDU, 137
 - gdcm::network::AAssociateRQPDU, 142
 - gdcm::ServiceClassUser, 649
- SetCallingAETitle
 - gdcm::network::AAssociateACPDU, 137
 - gdcm::network::AAssociateRQPDU, 142
- SetCenterOfRotation
 - gdcm::Surface, 692
- SetChangePrivateTags
 - gdcm::FileExplicitFilter, 367
- SetCheckFileMetaInformation
 - gdcm::Writer, 887
- SetCipherType
 - gdcm::CryptographicMessageSyntax, 253
- SetColor
 - gdcm::Printer, 577
- SetColorLevel
 - vtkImageColorViewer, 863
- SetColorWindow
 - vtkImageColorViewer, 863
- SetColumns
 - gdcm::Bitmap, 210
 - gdcm::Overlay, 524
- SetCommand
 - gdcm::network::PresentationDataValue, 573
- SetComponents
 - gdcm::PersonName, 539
- SetCompressIconImage
 - gdcm::ImageChangeTransferSyntax, 411
- SetComputeZSpacing
 - gdcm::IPPSorter, 448
- SetCoordinateStartValue
 - gdcm::Curve, 271
- SetCoordinateStepValue
 - gdcm::Curve, 272
- SetCryptographicMessageSyntax
 - gdcm::Anonymizer, 152
- SetCurve
 - gdcm::Curve, 272
 - vtkGDCMImageReader, 836
- SetCurveDataDescriptor
 - gdcm::Curve, 272

- SetCurveDescription
 - gdcm::Curve, [272](#)
- SetData
 - gdcm::DataEvent, [284](#)
- SetDataElement
 - gdcm::Bitmap, [210](#)
- SetDataSet
 - gdcm::File, [360](#)
 - gdcm::network::PresentationDataValue, [574](#)
- SetDataSetTransferSyntax
 - gdcm::FileMetaInformation, [372](#)
- SetDataValueRepresentation
 - gdcm::Curve, [272](#)
- SetDebug
 - gdcm::Trace, [726](#)
- SetDebugStream
 - gdcm::Trace, [726](#)
- SetDefaultTransferSyntax
 - gdcm::PresentationContextGenerator, [570](#)
- SetDerivationCodeSequenceCodeValue
 - gdcm::FileDerivation, [365](#)
- SetDerivationDescription
 - gdcm::FileDerivation, [365](#)
- SetDescription
 - gdcm::CSAHeaderDictEntry, [265](#)
 - gdcm::ModuleEntry, [505](#)
 - gdcm::Overlay, [524](#)
- SetDescriptor
 - gdcm::DICOMDIRGenerator, [302](#)
- SetDictName
 - gdcm::DictConverter, [307](#)
- SetDicts
 - gdcm::PythonFilter, [585](#)
 - gdcm::StringFilter, [683](#)
- SetDimension
 - gdcm::Bitmap, [210](#)
- SetDimensions
 - gdcm::Bitmap, [211](#)
 - gdcm::Curve, [272](#)
 - gdcm::ImageCodec, [417](#)
- SetDimensionsValue
 - gdcm::ImageHelper, [424](#)
- SetDirectionCosines
 - gdcm::Image, [399](#)
 - vtkGDCMImageWriter, [841](#)
- SetDirectionCosinesFromImageOrientationPatient
 - vtkGDCMImageWriter, [841](#)
- SetDirectionCosinesTolerance
 - gdcm::IPPSorter, [449](#)
- SetDirectionCosinesValue
 - gdcm::ImageHelper, [424](#)
- SetDirectory
 - gdcm::network::ULWritingCallback, [804](#)
 - gdcm::SerieHelper, [644](#)
- SetDisplayId
 - vtkImageColorViewer, [863](#)
- SetDomain
 - gdcm::BoxRegion, [218](#)
- SetDropDuplicatePositions
 - gdcm::IPPSorter, [449](#)
- SetElement
 - gdcm::Tag, [717](#)
- SetElementHandler
 - gdcm::Parser, [529](#)
- SetElementTag
 - gdcm::Tag, [717](#)
- SetElementXX
 - gdcm::DictEntry, [309](#)
- SetError
 - gdcm::Trace, [726](#)
- SetErrorStream
 - gdcm::Trace, [726](#)
- SetEvent
 - gdcm::network::ULEvent, [801](#)
- SetFile
 - gdcm::Anonymizer, [152](#)
 - gdcm::DICOMDIRGenerator, [302](#)
 - gdcm::FileDerivation, [365](#)
 - gdcm::FileExplicitFilter, [367](#)
 - gdcm::IconImageFilter, [393](#)
 - gdcm::Printer, [577](#)
 - gdcm::PythonFilter, [585](#)
 - gdcm::Reader, [603](#)
 - gdcm::SplitMosaicFilter, [668](#)
 - gdcm::StreamImageWriter, [676](#)
 - gdcm::StringFilter, [683](#)
 - gdcm::Validate, [814](#)
 - gdcm::Writer, [887](#)
- SetFileName
 - gdcm::Reader, [603](#)
 - gdcm::StreamImageReader, [672](#)
 - gdcm::StreamImageWriter, [676](#)
 - gdcm::Writer, [888](#)
 - vtkGDCMThreadedImageReader2, [858](#)
- SetFileNames
 - vtkGDCMImageReader, [836](#)
 - vtkGDCMImageWriter, [841](#)
 - vtkGDCMThreadedImageReader2, [858](#)
- SetFilePattern
 - vtkGDCMImageReader, [836](#)
- SetFilePrefix
 - vtkGDCMImageReader, [836](#)
- SetFilename
 - gdcm::TableReader, [711](#)
- SetFileNames
 - gdcm::DICOMDIRGenerator, [302](#)
- SetFiles
 - gdcm::FileSet, [378](#)

- SetFiniteVolume
 - gdcm::Surface, [693](#)
- SetForce
 - gdcm::ImageChangeTransferSyntax, [412](#)
 - gdcm::ImageFragmentSplitter, [421](#)
- SetForcePixelSpacing
 - gdcm::ImageHelper, [424](#)
- SetForceRescaleInterceptSlope
 - gdcm::ImageHelper, [424](#)
- SetFragmentSizeMax
 - gdcm::ImageFragmentSplitter, [421](#)
- SetFrameOrigin
 - gdcm::Overlay, [524](#)
- SetFromDataElement
 - gdcm::Attribute, [169](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [174](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [181](#)
 - gdcm::Element, [328](#)
 - gdcm::Element< TVR, VM::VM1_n >, [332](#)
- SetFromDataSet
 - gdcm::Attribute, [169](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, [174](#)
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [181](#)
 - gdcm::MediaStorage, [492](#)
- SetFromFile
 - gdcm::MediaStorage, [492](#)
- SetFromHeader
 - gdcm::MediaStorage, [492](#)
- SetFromModality
 - gdcm::MediaStorage, [492](#)
- SetFromSourceImageSequence
 - gdcm::MediaStorage, [492](#)
- SetFromString
 - gdcm::DirectionCosines, [318](#)
- SetFromUID
 - gdcm::UIDs, [756](#)
- SetGreenLUT
 - gdcm::LookupTable, [480](#)
- SetGroup
 - gdcm::Curve, [272](#)
 - gdcm::Overlay, [524](#)
 - gdcm::Tag, [717](#)
- SetGroupXX
 - gdcm::DictEntry, [309](#)
- SetHeader
 - gdcm::File, [360](#)
- SetHighBit
 - gdcm::PixelFormat, [548](#)
- SetHostname
 - gdcm::ServiceClassUser, [650](#)
- SetIE
 - gdcm::IODEntry, [444](#)
- SetIconImage
 - gdcm::Pixmap, [552](#)
- SetImage
 - gdcm::PixmapWriter, [561](#)
 - gdcm::SplitMosaicFilter, [668](#)
- SetImplementationClassUID
 - gdcm::FileMetaInformation, [372](#)
- SetImplementationVersionName
 - gdcm::FileMetaInformation, [372](#)
- SetInput
 - gdcm::BitmapToBitmapFilter, [215](#)
 - gdcm::ImageConverter, [419](#)
 - vtkImageColorViewer, [863](#)
- SetInputConnection
 - vtkImageColorViewer, [863](#)
- SetInputFileName
 - gdcm::DictConverter, [307](#)
 - gdcm::FileAnonymizer, [363](#)
- SetIntercept
 - gdcm::Image, [399](#)
 - gdcm::Rescaler, [608](#)
- SetKey
 - gdcm::CSAElement, [257](#)
- SetKeyword
 - gdcm::DictEntry, [309](#)
- SetLUT
 - gdcm::Bitmap, [211](#)
 - gdcm::ImageCodec, [417](#)
 - gdcm::LookupTable, [480](#)
 - gdcm::SegmentedPaletteColorLookupTable, [626](#)
- SetLastElement
 - gdcm::ParseException, [526](#)
- SetLastFragment
 - gdcm::network::PresentationDataValue, [574](#)
- SetLength
 - gdcm::ByteValue, [225](#)
 - gdcm::Element< TVR, VM::VM1_2 >, [329](#)
 - gdcm::Element< TVR, VM::VM1_n >, [332](#)
 - gdcm::Element< TVR, VM::VM2_2n >, [334](#)
 - gdcm::Element< TVR, VM::VM2_n >, [335](#)
 - gdcm::Element< TVR, VM::VM3_3n >, [337](#)
 - gdcm::Element< TVR, VM::VM3_n >, [338](#)
 - gdcm::RLECodec, [612](#)
 - gdcm::SequenceOfFragments, [635](#)
 - gdcm::SequenceOfItems, [641](#)
 - gdcm::Value, [816](#)
- SetLengthToUndefined
 - gdcm::SequenceOfItems, [641](#)
- SetLoadMode
 - gdcm::SerieHelper, [644](#)
- SetLookupTable
 - vtkImageMapToColors16, [868](#)

- SetLossless
 - gdcm::JPEGCodec, [467](#)
 - gdcm::JPEGLSCodec, [471](#)
- SetLossyError
 - gdcm::JPEGLSCodec, [471](#)
- SetLossyFlag
 - gdcm::Bitmap, [211](#)
 - gdcm::ImageCodec, [417](#)
- SetManifold
 - gdcm::Surface, [693](#)
- SetMaxPDULength
 - gdcm::network::ULConnectionInfo, [797](#)
- SetMaxPDUSize
 - gdcm::network::ULConnection, [795](#)
- SetMaximumLength
 - gdcm::network::MaximumLengthSub, [484](#)
- SetMaximumPointDistance
 - gdcm::Surface, [693](#)
- SetMeanPointDistance
 - gdcm::Surface, [693](#)
- SetMedicalImageProperties
 - vtkGDCMImageReader, [836](#)
 - vtkGDCMImageWriter, [841](#)
 - vtkGDCMPolyDataWriter, [850](#)
- SetMergeModeToAbstractSyntax
 - gdcm::PresentationContextGenerator, [570](#)
- SetMergeModeToTransferSyntax
 - gdcm::PresentationContextGenerator, [570](#)
- SetMeshPrimitive
 - gdcm::Surface, [693](#)
- SetMessageHeader
 - gdcm::network::PresentationDataValue, [574](#)
- SetMinMaxForPixelType
 - gdcm::Rescaler, [608](#)
- SetName
 - gdcm::CSAElement, [257](#)
 - gdcm::CSAHeaderDictEntry, [265](#)
 - gdcm::DictEntry, [309](#)
 - gdcm::LODEntry, [444](#)
 - gdcm::Macro, [482](#)
 - gdcm::Module, [502](#)
 - gdcm::ModuleEntry, [505](#)
 - gdcm::network::AbstractSyntax, [145](#)
 - gdcm::network::ApplicationContext, [155](#)
 - gdcm::network::TransferSyntaxSub, [731](#)
 - gdcm::PDBElement, [533](#)
- SetNameFromUID
 - gdcm::network::AbstractSyntax, [145](#)
 - gdcm::network::TransferSyntaxSub, [731](#)
- SetNeedByteSwap
 - gdcm::Bitmap, [211](#)
 - gdcm::ImageCodec, [417](#)
- SetNeedOverlayCleanup
 - gdcm::ImageCodec, [417](#)
- SetNestedDataSet
 - gdcm::Item, [453](#)
- SetNoOfItems
 - gdcm::CSAElement, [257](#)
- SetNoSwap
 - gdcm::Element, [328](#)
 - gdcm::Element< TVR, VM::VM1_n >, [332](#)
- SetNumberOfCurves
 - gdcm::Pixmap, [552](#)
- SetNumberOfDimensions
 - gdcm::Bitmap, [211](#)
 - gdcm::ImageCodec, [417](#)
- SetNumberOfFilenames
 - gdcm::FilenameGenerator, [377](#)
- SetNumberOfFrames
 - gdcm::Overlay, [524](#)
- SetNumberOfInputPorts
 - vtkGDCMPolyDataWriter, [850](#)
- SetNumberOfItems
 - gdcm::SequenceOfItems, [641](#)
- SetNumberOfOverlays
 - gdcm::Pixmap, [552](#)
- SetNumberOfPoints
 - gdcm::Curve, [272](#)
- SetNumberOfResolutions
 - gdcm::JPEG2000Codec, [461](#)
- SetNumberOfSegments
 - gdcm::SegmentWriter, [631](#)
- SetNumberOfSurfacePoints
 - gdcm::Surface, [693](#)
- SetNumberOfSurfaces
 - gdcm::SurfaceWriter, [700](#)
- SetNumberOfTableValues
 - vtkLookupTable16, [878](#)
- SetNumberOfValues
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, [181](#)
- SetNumberOfVectors
 - gdcm::Surface, [693](#)
- SetObliquityThresholdCosineValue
 - gdcm::Orientation, [518](#)
- SetOffScreenRendering
 - vtkImageColorViewer, [863](#)
- SetOrigin
 - gdcm::Image, [399](#)
 - gdcm::Overlay, [524](#)
- SetOriginValue
 - gdcm::ImageHelper, [424](#)
- SetOutputDimensions
 - gdcm::IconImageGenerator, [395](#)
- SetOutputFileName
 - gdcm::DictConverter, [307](#)
 - gdcm::FileAnonymizer, [363](#)
- SetOutputFormatToLuminance

- vtkImageMapToColors16, [868](#)
- SetOutputFormatToLuminanceAlpha
 - vtkImageMapToColors16, [868](#)
- SetOutputFormatToRGB
 - vtkImageMapToColors16, [868](#)
- SetOutputFormatToRGBA
 - vtkImageMapToColors16, [868](#)
- SetOutputType
 - gdcm::DictConverter, [307](#)
- SetOutsideValuePixel
 - gdcm::IconImageGenerator, [395](#)
- SetOverlay
 - gdcm::Overlay, [524](#)
- SetOverlayVisibility
 - vtkImageColorViewer, [863](#)
- SetOwner
 - gdcm::PrivateTag, [580](#)
- SetPDU
 - gdcm::network::ULEvent, [801](#)
- SetParentId
 - vtkImageColorViewer, [863](#)
- SetPattern
 - gdcm::FilenameGenerator, [377](#)
- SetPermissions
 - gdcm::System, [707](#)
- SetPhotometricInterpretation
 - gdcm::Bitmap, [211](#)
 - gdcm::ImageChangePhotometricInterpretation, [405](#)
 - gdcm::ImageCodec, [417](#)
- SetPixelFormat
 - gdcm::Bitmap, [211](#)
 - gdcm::ImageCodec, [417](#)
 - gdcm::JPEGCodec, [467](#)
 - gdcm::Rescaler, [608](#)
- SetPixelMinMax
 - gdcm::IconImageGenerator, [395](#)
- SetPixelRepresentation
 - gdcm::PixelFormat, [548](#)
- SetPixmap
 - gdcm::IconImageGenerator, [395](#)
 - gdcm::PixmapWriter, [561](#)
- SetPlanarConfiguration
 - gdcm::Bitmap, [211](#)
 - gdcm::ImageChangePlanarConfiguration, [408](#)
 - gdcm::ImageCodec, [417](#)
- SetPointCoordinatesData
 - gdcm::Surface, [693](#)
- SetPointPositionAccuracy
 - gdcm::Surface, [693](#)
- SetPointsBoundingBoxCoordinates
 - gdcm::Surface, [693](#)
- SetPort
 - gdcm::ServiceClassUser, [650](#)
- SetPortSCP
 - gdcm::ServiceClassUser, [650](#)
- SetPosition
 - vtkImageColorViewer, [864](#)
- SetPreamble
 - gdcm::FileMetaInformation, [372](#)
- SetPrefix
 - gdcm::FilenameGenerator, [377](#)
- SetPresentationContextId
 - gdcm::network::PresentationContextAC, [568](#)
 - gdcm::network::PresentationContextRQ, [572](#)
 - gdcm::network::PresentationDataValue, [574](#)
 - gdcm::PresentationContext, [567](#)
- SetPresentationContexts
 - gdcm::network::ULConnection, [795](#)
 - gdcm::ServiceClassUser, [650](#)
- SetPrimitiveData
 - gdcm::MeshPrimitive, [499](#)
- SetPrimitiveType
 - gdcm::MeshPrimitive, [499](#)
- SetPrimitivesData
 - gdcm::MeshPrimitive, [499](#)
- SetPrivateCreator
 - gdcm::Tag, [718](#)
- SetProcessingAlgorithm
 - gdcm::Surface, [693](#)
- SetProgress
 - gdcm::ProgressEvent, [582](#)
- SetPropertyCategory
 - gdcm::Segment, [624](#)
- SetPropertyType
 - gdcm::Segment, [624](#)
- SetPurposeOfReferenceCodeSequenceCodeValue
 - gdcm::FileDerivation, [365](#)
- SetQuality
 - gdcm::JPEG2000Codec, [461](#)
 - gdcm::JPEGCodec, [467](#)
- SetRTStructSetProperties
 - vtkGDCMPolyDataWriter, [850](#)
- SetRate
 - gdcm::JPEG2000Codec, [461](#)
- SetReason
 - gdcm::network::AAAbortPDU, [134](#)
 - gdcm::network::PresentationContextAC, [568](#)
- SetRecommendedDisplayCIELabValue
 - gdcm::Surface, [693](#)
- SetRecommendedDisplayGrayscaleValue
 - gdcm::Surface, [693](#)
- SetRecommendedPresentationOpacity
 - gdcm::Surface, [693](#)
- SetRecommendedPresentationType
 - gdcm::Surface, [693](#)
- SetRecomputeItemLength
 - gdcm::FileExplicitFilter, [367](#)
- SetRecomputeSequenceLength

- gdcm::FileExplicitFilter, 367
- SetRedLUT
 - gdcm::LookupTable, 480
- SetRef
 - gdcm::IODEntry, 444
- SetRegion
 - gdcm::ImageRegionReader, 431
- SetRenderWindow
 - vtkImageColorViewer, 864
- SetRenderer
 - vtkImageColorViewer, 864
- SetRescaleInterceptSlopeValue
 - gdcm::ImageHelper, 424
- SetRetired
 - gdcm::DictEntry, 309
- SetReversible
 - gdcm::JPEG2000Codec, 461
- SetRoot
 - gdcm::UIDGenerator, 737
- SetRootDirectory
 - gdcm::DICOMDIRGenerator, 302
- SetRows
 - gdcm::Bitmap, 211
 - gdcm::Overlay, 524
- SetSamplesPerPixel
 - gdcm::PixelFormat, 548
- SetScalarType
 - gdcm::PixelFormat, 548
- SetSearchParameter
 - gdcm::BaseRootQuery, 199
- SetSegmentAlgorithmName
 - gdcm::Segment, 624
- SetSegmentAlgorithmType
 - gdcm::Segment, 624
- SetSegmentDescription
 - gdcm::Segment, 624
- SetSegmentLabel
 - gdcm::Segment, 624
- SetSegmentNumber
 - gdcm::Segment, 624
- SetSegments
 - gdcm::SegmentWriter, 631
- SetSize
 - vtkImageColorViewer, 864
- SetSlice
 - vtkImageColorViewer, 864
- SetSliceOrientation
 - vtkImageColorViewer, 864
- SetSliceOrientationToXY
 - vtkImageColorViewer, 864
- SetSliceOrientationToXZ
 - vtkImageColorViewer, 864
- SetSliceOrientationToYZ
 - vtkImageColorViewer, 864
- SetSlope
 - gdcm::Image, 399
 - gdcm::Rescaler, 609
- SetSortFunction
 - gdcm::Sorter, 664
- SetSource
 - gdcm::network::AAAbortPDU, 135
- SetSourceApplicationEntityTitle
 - gdcm::FileMetaInformation, 372
- SetSpacing
 - gdcm::Image, 399
- SetSpacingValue
 - gdcm::ImageHelper, 424
- SetState
 - gdcm::network::ULConnection, 795
- SetStream
 - gdcm::Reader, 604
 - gdcm::StreamImageReader, 673
 - gdcm::StreamImageWriter, 676
 - gdcm::Trace, 726
 - gdcm::Writer, 888
- SetStreamToFile
 - gdcm::Trace, 726
- SetStyle
 - gdcm::Printer, 577
- SetSurfaceComments
 - gdcm::Surface, 693
- SetSurfaceCount
 - gdcm::Segment, 624
- SetSurfaceNumber
 - gdcm::Surface, 693
- SetSurfaceProcessing
 - gdcm::Surface, 693
- SetSurfaceProcessingDescription
 - gdcm::Surface, 693
- SetSurfaceProcessingRatio
 - gdcm::Surface, 693
- SetSyngoDT
 - gdcm::CSAElement, 257
- SetTag
 - gdcm::AnonymizeEvent, 147
 - gdcm::DataElement, 279
- SetTargetPixelType
 - gdcm::Rescaler, 609
- SetTileSize
 - gdcm::JPEG2000Codec, 461
- SetTimeout
 - gdcm::network::ARTIMTimer, 161
 - gdcm::ServiceClassUser, 650
- SetToUndefined
 - gdcm::VL, 820
- SetTransferSyntax
 - gdcm::Bitmap, 212
 - gdcm::ImageChangeTransferSyntax, 412

- gdcm::network::PresentationContextAC, 568
- SetTuple
 - gdcm::network::RoleSelectionSub, 613
 - gdcm::network::ServiceClassApplicationInformation, 645
 - gdcm::network::SOPClassExtendedNegociationSub, 660
- SetType
 - gdcm::ModuleEntry, 505
 - gdcm::Overlay, 524
- SetTypeOfData
 - gdcm::Curve, 272
- SetUsage
 - gdcm::IODEntry, 444
- SetUseSeriesDetails
 - gdcm::SerieHelper, 644
- SetUseTargetPixelType
 - gdcm::Rescaler, 609
- SetUseVRUN
 - gdcm::FileExplicitFilter, 367
- SetUserCodec
 - gdcm::ImageChangeTransferSyntax, 412
- SetUserData
 - gdcm::Parser, 529
- SetUserInformation
 - gdcm::network::AAAssociateRQPDU, 142
- SetVL
 - gdcm::DataElement, 280
- SetVLToUndefined
 - gdcm::DataElement, 280
- SetVM
 - gdcm::CSAElement, 257
 - gdcm::CSAHeaderDictEntry, 265
 - gdcm::DictEntry, 309
- SetVR
 - gdcm::CSAElement, 257
 - gdcm::CSAHeaderDictEntry, 265
 - gdcm::DataElement, 280
 - gdcm::DictEntry, 309
- SetValue
 - gdcm::Attribute, 169
 - gdcm::Attribute< Group, Element, TVR, VM::VM1 >, 174
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 181
 - gdcm::CSAElement, 257
 - gdcm::DataElement, 279
 - gdcm::Element, 328
 - gdcm::Element< TVR, VM::VM1_n >, 332
 - gdcm::PDBElement, 533
- SetValues
 - gdcm::Attribute, 169
 - gdcm::Attribute< Group, Element, TVR, VM::VM1_n >, 181
- SetVectorAccuracy
 - gdcm::Surface, 693
- SetVectorCoordinateData
 - gdcm::Surface, 693
- SetVectorDimensionality
 - gdcm::Surface, 693
- SetWarning
 - gdcm::Trace, 726
- SetWarningStream
 - gdcm::Trace, 726
- SetWindowId
 - vtkImageColorViewer, 864
- SetWriteDataSetOnly
 - gdcm::Writer, 888
- SetZSpacingTolerance
 - gdcm::IPPSorter, 449
- setAttribute
 - gdcm::terminal, 131
- setbgcolor
 - gdcm::terminal, 131
- setfgcolor
 - gdcm::terminal, 131
- setmode
 - gdcm::terminal, 131
- SetupInteractor
 - vtkImageColorViewer, 864
- Shift
 - vtkGDCMImageReader, 838
- ShiftEnd
 - gdcm::ByteBuffer, 219
- ShowAbort
 - gdcm::SimpleSubjectWatcher, 656
- ShowAnonymization
 - gdcm::SimpleSubjectWatcher, 656
- ShowData
 - gdcm::SimpleSubjectWatcher, 656
- ShowDataSet
 - gdcm::SimpleSubjectWatcher, 656
- ShowIteration
 - gdcm::SimpleSubjectWatcher, 656
- ShowProgress
 - gdcm::SimpleSubjectWatcher, 656
- SimpleMemberCommand
 - gdcm::SimpleMemberCommand, 654
- SimpleSubjectWatcher
 - gdcm::SimpleSubjectWatcher, 656
- SingleSerieUIDFileSetHT
 - gdcm::SerieHelper, 644
- SingleSerieUIDFileSetmap
 - gdcm::SerieHelper, 643
- Size
 - gdcm::CodeString, 241
 - gdcm::DataSet, 291
 - gdcm::GroupDict, 391

- gdcm::network::AAAbortPDU, [135](#)
- gdcm::network::AAAssociateACPDU, [137](#)
- gdcm::network::AAAssociateRJPDU, [139](#)
- gdcm::network::AAAssociateRQPDU, [143](#)
- gdcm::network::AbstractSyntax, [145](#)
- gdcm::network::ApplicationContext, [155](#)
- gdcm::network::AReleaseRPPDU, [158](#)
- gdcm::network::AReleaseRQPDU, [160](#)
- gdcm::network::AsynchronousOperationsWindow-Sub, [163](#)
- gdcm::network::BasePDU, [196](#)
- gdcm::network::ImplementationClassUIDSub, [436](#)
- gdcm::network::ImplementationVersionNameSub, [437](#)
- gdcm::network::MaximumLengthSub, [484](#)
- gdcm::network::PDataTFPDU, [531](#)
- gdcm::network::PresentationContextAC, [568](#)
- gdcm::network::PresentationContextRQ, [572](#)
- gdcm::network::PresentationDataValue, [574](#)
- gdcm::network::RoleSelectionSub, [613](#)
- gdcm::network::ServiceClassApplicationInformation, [645](#)
- gdcm::network::SOPClassExtendedNegociationSub, [660](#)
- gdcm::network::TransferSyntaxSub, [731](#)
- gdcm::network::UserInformation, [812](#)
- size_type
 - gdcm::CodeString, [240](#)
 - gdcm::LO, [475](#)
 - gdcm::String, [681](#)
- SizeType
 - gdcm::DataSet, [287](#)
 - gdcm::FilenameGenerator, [376](#)
 - gdcm::IOD, [442](#)
 - gdcm::NestedModuleEntries, [513](#)
 - gdcm::network::AAAssociateACPDU, [137](#)
 - gdcm::network::AAAssociateRQPDU, [141](#)
 - gdcm::network::PDataTFPDU, [531](#)
 - gdcm::network::PresentationContextRQ, [571](#)
 - gdcm::PresentationContext, [566](#)
 - gdcm::PresentationContextGenerator, [569](#)
 - gdcm::SequenceOfFragments, [634](#)
 - gdcm::SequenceOfItems, [639](#)
- Slice
 - vtkImageColorViewer, [865](#)
- SliceOrientation
 - vtkImageColorViewer, [865](#)
- SmartPointer
 - gdcm::Object, [516](#)
 - gdcm::SmartPointer, [658](#)
- Sort
 - gdcm::IPPSorter, [449](#)
 - gdcm::Sorter, [664](#)
- SortFunc
 - gdcm::Sorter, [665](#)
- SortFunction
 - gdcm::Sorter, [663](#)
- Sorter
 - gdcm::Sorter, [664](#)
- SpacialFiducialsStorage
 - gdcm::MediaStorage, [489](#)
- SpacialRegistrationStorage
 - gdcm::MediaStorage, [489](#)
- Spacing
 - gdcm::Spacing, [666](#)
- SpacingType
 - gdcm::Spacing, [666](#)
- SpatialFiducialsStorage
 - gdcm::UIDs, [746](#)
- SpatialRegistrationStorage
 - gdcm::UIDs, [746](#)
- Spectroscopy
 - gdcm::Spectroscopy, [667](#)
- Split
 - gdcm::ImageFragmentSplitter, [421](#)
 - gdcm::SplitMosaicFilter, [668](#)
- SplitExtent
 - vtkGDCMThreadedImageReader2, [858](#)
- SplitMosaicFilter
 - gdcm::SplitMosaicFilter, [668](#)
- Squeeze
 - gdcm::ApplicationEntity, [156](#)
- StableSort
 - gdcm::Sorter, [664](#)
- StandaloneCurveStorage
 - gdcm::MediaStorage, [489](#)
- StandaloneCurveStorageRetired
 - gdcm::UIDs, [745](#)
- StandaloneModalityLUTStorage
 - gdcm::MediaStorage, [489](#)
- StandaloneModalityLUTStorageRetired
 - gdcm::UIDs, [746](#)
- StandaloneOverlayStorage
 - gdcm::MediaStorage, [489](#)
- StandaloneOverlayStorageRetired
 - gdcm::UIDs, [745](#)
- StandalonePETCurveStorageRetired
 - gdcm::UIDs, [747](#)
- StandaloneVOILUTStorage
 - gdcm::MediaStorage, [489](#)
- StandaloneVOILUTStorageRetired
 - gdcm::UIDs, [746](#)
- Start
 - gdcm::network::ARTIMTimer, [161](#)
- StartAssociation
 - gdcm::ServiceClassUser, [650](#)
- StartElement
 - gdcm::TableReader, [711](#)

- gdcm::XMLDictReader, 890
- gdcm::XMLPrivateDictReader, 892
- StartElementHandler
 - gdcm::Parser, 528
- StartFilter
 - gdcm::SimpleSubjectWatcher, 656
- StereometricRelationshipStorage
 - gdcm::UIDs, 746
- Stop
 - gdcm::network::ARTIMTimer, 161
- StopAssociation
 - gdcm::ServiceClassUser, 650
- StopProtocol
 - gdcm::network::ULConnection, 795
- StorageCommitmentPullModelSOPClassRetired
 - gdcm::UIDs, 744
- StorageCommitmentPullModelSOPInstanceRetired
 - gdcm::UIDs, 744
- StorageCommitmentPushModelSOPClass
 - gdcm::UIDs, 744
- StorageCommitmentPushModelSOPInstance
 - gdcm::UIDs, 744
- StorageServiceClass
 - gdcm::UIDs, 744
- StoredPrintStorageSOPClassRetired
 - gdcm::UIDs, 745
- StrCaseCmp
 - gdcm::System, 707
- StrNCaseCmp
 - gdcm::System, 707
- StrTokR
 - gdcm::System, 707
- Stream
 - gdcm::Writer, 888
- StreamImageReader
 - gdcm::Reader, 604
 - gdcm::StreamImageReader, 671
- StreamImageWriter
 - gdcm::StreamImageWriter, 676
 - gdcm::Writer, 888
- String
 - gdcm::String, 681
- StringFilter
 - gdcm::StringFilter, 683
- StructureSetDate
 - vtkRTStructSetProperties, 883
- StructureSetLabel
 - vtkRTStructSetProperties, 883
- StructureSetName
 - vtkRTStructSetProperties, 883
- StructureSetTime
 - vtkRTStructSetProperties, 883
- Study
 - gdcm::Study, 685
- StudyComponentManagementSOPClass
 - gdcm::MediaStorage, 489
- StudyComponentManagementSOPClassRetired
 - gdcm::UIDs, 744
- StudyRootQueryRetrieveInformationModelFIND
 - gdcm::UIDs, 747
- StudyRootQueryRetrieveInformationModelGET
 - gdcm::UIDs, 747
- StudyRootQueryRetrieveInformationModelMOVE
 - gdcm::UIDs, 747
- StudyInstanceUID
 - vtkRTStructSetProperties, 883
- Subject
 - gdcm::Subject, 686
- SubstanceAdministrationLoggingSOPClass
 - gdcm::UIDs, 744
- SubstanceAdministrationLoggingSOPInstance
 - gdcm::UIDs, 744
- SubstanceApprovalQuerySOPClass
 - gdcm::UIDs, 748
- Superclass
 - gdcm::AnonymizeEvent, 147
 - gdcm::DataEvent, 283
 - gdcm::DataSetEvent, 292
 - gdcm::LO, 475
 - gdcm::ProgressEvent, 582
- Surface
 - gdcm::Surface, 690
- SurfaceSegmentationStorage
 - gdcm::MediaStorage, 490
 - gdcm::UIDs, 749
- SurfaceCount
 - gdcm::Segment, 624
- SurfaceReader
 - gdcm::SurfaceReader, 698
- SurfaceVector
 - gdcm::Segment, 623
- SurfaceWriter
 - gdcm::SurfaceWriter, 700
- Surfaces
 - gdcm::Segment, 624
- Swap
 - gdcm::ByteSwap, 220
 - gdcm::SwapperDoOp, 702
 - gdcm::SwapperNoOp, 703
- SwapArray
 - gdcm::SwapperDoOp, 702
 - gdcm::SwapperNoOp, 703
- SwapCode
 - gdcm::SwapCode, 702
- SwapCodeType
 - gdcm::SwapCode, 701
- SwapFromSwapCodeIntoSystem
 - gdcm::ByteSwap, 220

- SwapRange
 - gdcm::ByteSwap, [220](#)
- SwapRangeFromSwapCodeIntoSystem
 - gdcm::ByteSwap, [220](#)
- SyngoDTField
 - gdcm::CSAElement, [258](#)
- SyntaxError
 - gdcm::Parser, [528](#)
- SystemIsBigEndian
 - gdcm::ByteSwap, [220](#)
- SystemIsLittleEndian
 - gdcm::ByteSwap, [220](#)
- T1
 - gdcm::Type, [734](#)
- T1C
 - gdcm::Type, [734](#)
- T2
 - gdcm::Type, [734](#)
- T2C
 - gdcm::Type, [734](#)
- T3
 - gdcm::Type, [734](#)
- TM
 - gdcm::VR, [827](#)
- TRIANGLE
 - gdcm::MeshPrimitive, [498](#)
- TRIANGLE_FAN
 - gdcm::MeshPrimitive, [498](#)
- TRIANGLE_STRIP
 - gdcm::MeshPrimitive, [498](#)
- TS_END
 - gdcm::TransferSyntax, [729](#)
- TConstMemberFunctionPointer
 - gdcm::MemberCommand, [494](#)
- TMComp
 - gdcm, [118](#)
- TMemberFunctionPointer
 - gdcm::MemberCommand, [495](#)
 - gdcm::SimpleMemberCommand, [654](#)
- TS
 - gdcm::Bitmap, [213](#)
- TSName
 - gdcm::UIDs, [742](#)
- TSType
 - gdcm::TransferSyntax, [729](#)
 - gdcm::UIDs, [749](#)
- TYPETOENCODING
 - gdcm, [124](#)
 - gdcmVR.h, [1150](#)
- TYPETOLENGTH
 - gdcmVM.h, [1148](#)
- Table
 - gdcm::Table, [708](#)
- Table16
 - vtkLookupTable16, [878](#)
- TableEntry
 - gdcm::TableEntry, [709](#)
- TableReader
 - gdcm::TableReader, [710](#)
- TableRow
 - gdcm::network::TableRow, [712](#)
- Tag
 - gdcm::Tag, [714](#)
- tag
 - gdcm::Tag, [718](#)
- TagMismatchError
 - gdcm::Parser, [528](#)
- TagField
 - gdcm::DataElement, [281](#)
- TagPath
 - gdcm::TagPath, [719](#)
- TagToValue
 - gdcm::Scanner, [617](#)
- TagToValueValueType
 - gdcm::Scanner, [617](#)
- tags
 - gdcm::Tag, [718](#)
- TalairachBrainAtlasFrameofReference
 - gdcm::UIDs, [743](#)
- TestAbortOff
 - gdcm::SimpleSubjectWatcher, [656](#)
- TestAbortOn
 - gdcm::SimpleSubjectWatcher, [656](#)
- TestPBKDF2
 - gdcm::ASN1, [162](#)
- Testing
 - gdcm::Testing, [721](#)
- TestsList.txt, [1154](#)
- TextSRStorageTrialRetired
 - gdcm::UIDs, [746](#)
- ThreadedExecute
 - vtkImageRGBToYBR, [874](#)
 - vtkImageYBRToRGB, [876](#)
- ThreadedRequestData
 - vtkGDCMThreadedImageReader2, [858](#)
 - vtkImageMapToColors16, [868](#)
 - vtkImageMapToWindowLevelColors2, [870](#)
- to_string
 - gdcm, [124](#)
- ToPyObject
 - gdcm::PythonFilter, [585](#)
- ToString
 - gdcm::StringFilter, [684](#)
- ToStringPair
 - gdcm::StringFilter, [684](#)
- ToUnixSlashes
 - gdcm::Filename, [374](#)

- ToWindowsSlashes
 - gdcm::Filename, [374](#)
- ToshibaPrivateDataStorage
 - gdcm::MediaStorage, [489](#)
- Trace
 - gdcm::Trace, [725](#)
- TransferSyntax
 - gdcm::TransferSyntax, [729](#)
- TransferSyntaxArrayType
 - gdcm::PresentationContext, [566](#)
- TransferSyntaxStringsType
 - gdcm::UIDs, [742](#)
- TransferSyntaxSub
 - gdcm::network::TransferSyntaxSub, [731](#)
- Transition
 - gdcm::network::Transition, [732](#), [733](#)
- transitions
 - gdcm::network::TableRow, [712](#)
- Trim
 - gdcm::String, [682](#)
- TrimInternal
 - gdcm::CodeString, [241](#)
- Truncate
 - gdcm::String, [682](#)
- TryJPEG2000Codec
 - gdcm::Bitmap, [212](#)
 - gdcm::ImageChangeTransferSyntax, [412](#)
- TryJPEG2000Codec2
 - gdcm::Bitmap, [212](#)
- TryJPEGCodec
 - gdcm::Bitmap, [212](#)
 - gdcm::ImageChangeTransferSyntax, [412](#)
- TryJPEGCodec2
 - gdcm::Bitmap, [212](#)
- TryJPEGLSCodec
 - gdcm::Bitmap, [212](#)
 - gdcm::ImageChangeTransferSyntax, [412](#)
- TryKAKADUCodec
 - gdcm::Bitmap, [212](#)
- TryPVRGCodec
 - gdcm::Bitmap, [212](#)
- TryRAWCodec
 - gdcm::Bitmap, [212](#)
 - gdcm::ImageChangeTransferSyntax, [412](#)
- TryRLECodec
 - gdcm::Bitmap, [212](#)
 - gdcm::ImageChangeTransferSyntax, [412](#)
- Type
 - gdcm::Element, [327](#)
 - gdcm::Element< TVR, VM::VM1_n >, [331](#)
 - gdcm::Type, [735](#)
 - gdcm::VL, [819](#)
- TypeType
 - gdcm::Type, [734](#)
- UI
 - gdcm::VR, [827](#)
- UINT12
 - gdcm::PixelFormat, [546](#)
- UINT16
 - gdcm::PixelFormat, [546](#)
- UINT32
 - gdcm::PixelFormat, [546](#)
- UINT8
 - gdcm::PixelFormat, [546](#)
- UL
 - gdcm::VR, [827](#)
- UN
 - gdcm::VR, [827](#)
- UNKNOWN
 - gdcm::PhotometricInterpretation, [543](#)
- UNKNOWN
 - gdcm::CSAHeader, [260](#)
 - gdcm::LookupTable, [478](#)
 - gdcm::Orientation, [518](#)
 - gdcm::PixelFormat, [546](#)
 - gdcm::Spacing, [666](#)
 - gdcm::Surface, [690](#)
 - gdcm::Type, [734](#)
- URI
 - gdcm::MediaStorage, [490](#)
- US
 - gdcm::VR, [827](#)
- US_SS
 - gdcm::VR, [827](#)
- US_SS_OW
 - gdcm::VR, [827](#)
- UT
 - gdcm::VR, [827](#)
- UIComp
 - gdcm, [118](#)
- UIDGenerator
 - gdcm::UIDGenerator, [736](#)
- ULAction
 - gdcm::network::ULAction, [758](#)
- ULBasicCallback
 - gdcm::network::ULBasicCallback, [792](#)
- ULConnection
 - gdcm::network::ULConnection, [794](#)
- ULConnectionCallback
 - gdcm::network::ULConnectionCallback, [796](#)
- ULConnectionInfo
 - gdcm::network::ULConnectionInfo, [797](#)
- ULConnectionManager
 - gdcm::network::ULConnectionManager, [800](#)
- ULEvent
 - gdcm::network::ULEvent, [801](#)
- ULTransitionTable
 - gdcm::network::ULTransitionTable, [802](#)

ULWritingCallback
 gdcmm::network::ULWritingCallback, [803](#)
UTComp
 gdcmm, [118](#)
uid_1_2_840_10008_15_0_3_1
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_10
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_11
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_12
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_13
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_14
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_15
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_16
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_17
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_18
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_19
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_2
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_20
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_21
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_22
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_23
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_24
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_25
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_26
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_27
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_28
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_29
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_3
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_30
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_31
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_3_4
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_5
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_6
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_7
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_8
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_3_9
 gdcmm::UIDs, [754](#)
uid_1_2_840_10008_15_0_4_1
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_4_2
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_4_3
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_4_4
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_4_5
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_4_6
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_4_7
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_15_0_4_8
 gdcmm::UIDs, [755](#)
uid_1_2_840_10008_1_1
 gdcmm::UIDs, [749](#)
uid_1_2_840_10008_1_2
 gdcmm::UIDs, [749](#)
uid_1_2_840_10008_1_20_1
 gdcmm::UIDs, [750](#)
uid_1_2_840_10008_1_20_1_1
 gdcmm::UIDs, [750](#)
uid_1_2_840_10008_1_20_2
 gdcmm::UIDs, [750](#)
uid_1_2_840_10008_1_20_2_1
 gdcmm::UIDs, [750](#)
uid_1_2_840_10008_1_2_1
 gdcmm::UIDs, [749](#)
uid_1_2_840_10008_1_2_1_99
 gdcmm::UIDs, [749](#)
uid_1_2_840_10008_1_2_2
 gdcmm::UIDs, [749](#)
uid_1_2_840_10008_1_2_4_100
 gdcmm::UIDs, [750](#)
uid_1_2_840_10008_1_2_4_50
 gdcmm::UIDs, [749](#)
uid_1_2_840_10008_1_2_4_51
 gdcmm::UIDs, [749](#)
uid_1_2_840_10008_1_2_4_52
 gdcmm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_53
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_54
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_55
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_56
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_57
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_58
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_59
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_60
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_61
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_62
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_63
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_64
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_65
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_66
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_70
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_80
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_81
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_90
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_91
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_92
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_93
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_94
gdcm::UIDs, [749](#)

uid_1_2_840_10008_1_2_4_95
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_2_5
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_2_6_1
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_2_6_2
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_3_10
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_40
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_40_1
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_42
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_42_1
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_1
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_10
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_11
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_12
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_13
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_14
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_15
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_16
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_17
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_18
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_2
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_3
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_4
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_5
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_6
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_7
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_8
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_1_9
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_2_1
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_4_2_2
gdcm::UIDs, [750](#)

uid_1_2_840_10008_1_9
gdcm::UIDs, [750](#)

uid_1_2_840_10008_2_16_4
gdcm::UIDs, [750](#)

uid_1_2_840_10008_2_6_1
gdcm::UIDs, [750](#)

uid_1_2_840_10008_3_1_1_1
gdcml::UIDs, [750](#)

uid_1_2_840_10008_3_1_2_1_1
gdcml::UIDs, [750](#)

uid_1_2_840_10008_3_1_2_1_4
gdcml::UIDs, [750](#)

uid_1_2_840_10008_3_1_2_2_1
gdcml::UIDs, [750](#)

uid_1_2_840_10008_3_1_2_3_1
gdcml::UIDs, [750](#)

uid_1_2_840_10008_3_1_2_3_2
gdcml::UIDs, [751](#)

uid_1_2_840_10008_3_1_2_3_3
gdcml::UIDs, [751](#)

uid_1_2_840_10008_3_1_2_3_4
gdcml::UIDs, [751](#)

uid_1_2_840_10008_3_1_2_3_5
gdcml::UIDs, [751](#)

uid_1_2_840_10008_3_1_2_5_1
gdcml::UIDs, [751](#)

uid_1_2_840_10008_3_1_2_5_4
gdcml::UIDs, [751](#)

uid_1_2_840_10008_3_1_2_5_5
gdcml::UIDs, [751](#)

uid_1_2_840_10008_3_1_2_6_1
gdcml::UIDs, [751](#)

uid_1_2_840_10008_4_2
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_1
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_14
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_15
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_16
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_16_376
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_17
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_17_376
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_18
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_18_1
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_2
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_22
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_23
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_24
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_24_1
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_25
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_26
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_27
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_29
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_30
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_31
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_32
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_33
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_4
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_4_1
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_4_2
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_9
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_1_9_1
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_4_1_1_1
gdcml::UIDs, [751](#)

uid_1_2_840_10008_5_1_4_1_1_10
gdcml::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_104_1
gdcml::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_104_2
gdcml::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_11
gdcml::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_11_1
gdcml::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_11_2
gdcml::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_11_3
gdcml::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_11_4
gdcml::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_128
gdcml::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_129
gdcml::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_12_1
gdcml::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_12_1_1
gdcml::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_12_2
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_12_2_1
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_12_3
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_13_1_1
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_13_1_2
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_13_1_3
 gdcml::UIDs, 755
 uid_1_2_840_10008_5_1_4_1_1_1_1
 gdcml::UIDs, 751
 uid_1_2_840_10008_5_1_4_1_1_1_1_1
 gdcml::UIDs, 751
 uid_1_2_840_10008_5_1_4_1_1_1_2
 gdcml::UIDs, 751
 uid_1_2_840_10008_5_1_4_1_1_1_2_1
 gdcml::UIDs, 751
 uid_1_2_840_10008_5_1_4_1_1_1_3
 gdcml::UIDs, 751
 uid_1_2_840_10008_5_1_4_1_1_1_3_1
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_2
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_20
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_2_1
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_3
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_3_1
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_4
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_481_1
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_481_2
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_481_3
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_481_4
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_481_5
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_481_6
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_481_7
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_481_8
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_481_9
 gdcml::UIDs, 753

uid_1_2_840_10008_5_1_4_1_1_4_1
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_4_2
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_5
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_6
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_66
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_66_1
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_66_2
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_66_3
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_66_4
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_66_5
 gdcml::UIDs, 755
 uid_1_2_840_10008_5_1_4_1_1_67
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_6_1
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_6_2
 gdcml::UIDs, 755
 uid_1_2_840_10008_5_1_4_1_1_7
 gdcml::UIDs, 752
 uid_1_2_840_10008_5_1_4_1_1_77_1
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_1
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_1_1
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_2
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_2_1
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_3
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_4
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_4_1
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_1
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_2
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_3
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_5_4
 gdcml::UIDs, 753
 uid_1_2_840_10008_5_1_4_1_1_77_1_6
 gdcml::UIDs, 755

uid_1_2_840_10008_5_1_4_1_1_77_2
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_7_1
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_7_2
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_7_3
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_7_4
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_8
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_88_1
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_11
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_2
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_22
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_3
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_33
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_4
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_40
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_50
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_59
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_65
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_88_67
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_1_9
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_9_1
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_9_1_1
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_9_1_2
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_9_1_3
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_9_2_1
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_9_3_1
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_1_9_4_1
gdcm::UIDs, [752](#)

uid_1_2_840_10008_5_1_4_1_2_1_1
gdcm::UIDs, [753](#)

uid_1_2_840_10008_5_1_4_1_2_1_2
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_1_2_1_3
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_1_2_2_1
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_1_2_2_2
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_1_2_2_3
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_1_2_3_1
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_1_2_3_2
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_1_2_3_3
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_31
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_32
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_32_1
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_32_2
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_32_3
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_33
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_1
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_2
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_3
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_4
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_4_1
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_4_2
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_4_3
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_4_4
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_34_5
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_37_1
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_37_2
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_37_3
gdcm::UIDs, [754](#)

uid_1_2_840_10008_5_1_4_38_1
gdcm::UIDs, [754](#)

- uid_1_2_840_10008_5_1_4_38_2
 - gdcm::UIDs, [754](#)
- uid_1_2_840_10008_5_1_4_38_3
 - gdcm::UIDs, [754](#)
- uid_1_2_840_10008_5_1_4_41
 - gdcm::UIDs, [754](#)
- uid_1_2_840_10008_5_1_4_42
 - gdcm::UIDs, [754](#)
- UltrasoundImageStorage
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- UltrasoundImageStorageRetired
 - gdcm::MediaStorage, [488](#)
 - gdcm::UIDs, [745](#)
- UltrasoundMultiFramedImageStorage
 - gdcm::MediaStorage, [488](#)
- UltrasoundMultiFramedImageStorageRetired
 - gdcm::MediaStorage, [488](#)
- UltrasoundMultiframeImageStorage
 - gdcm::UIDs, [745](#)
- UltrasoundMultiframeImageStorageRetired
 - gdcm::UIDs, [745](#)
- UnInstallPipeline
 - vtkImageColorViewer, [865](#)
- UnRegister
 - gdcm::Object, [516](#)
- UndefinedEntityError
 - gdcm::Parser, [528](#)
- underline
 - gdcm::terminal, [131](#)
- UnexpectedStateError
 - gdcm::Parser, [528](#)
- UnifiedProcedureStepEventSOPClass
 - gdcm::UIDs, [747](#)
- UnifiedProcedureStepPullSOPClass
 - gdcm::UIDs, [747](#)
- UnifiedProcedureStepPushSOPClass
 - gdcm::UIDs, [747](#)
- UnifiedProcedureStepWatchSOPClass
 - gdcm::UIDs, [747](#)
- UnifiedWorklistandProcedureStepSOPInstance
 - gdcm::UIDs, [747](#)
- UnifiedWorklistandProcedureStepServiceClass
 - gdcm::UIDs, [747](#)
- Unknown
 - gdcm::SwapCode, [701](#)
 - gdcm::TransferSyntax, [729](#)
- Unpack
 - gdcm::Unpacker12Bits, [808](#)
- Update
 - gdcm::Curve, [272](#)
 - gdcm::Overlay, [525](#)
- UpdateDisplayExtent
 - vtkImageColorViewer, [865](#)
- UpdateOrientation
 - vtkImageColorViewer, [865](#)
- UpdatePosition
 - gdcm::ByteBuffer, [219](#)
- Usage
 - gdcm::Usage, [810](#)
- UsageType
 - gdcm::Usage, [810](#)
- UseDictAlways
 - gdcm::PythonFilter, [585](#)
 - gdcm::StringFilter, [684](#)
- UserOption
 - gdcm::Usage, [810](#)
- UserInfoInformation
 - gdcm::network::UserInformation, [812](#)
- UserOrdering
 - gdcm::SerieHelper, [644](#)
- V
 - gdcm::Validate, [814](#)
- VERBOSE_STYLE
 - gdcm::Printer, [576](#)
- VERTEX
 - gdcm::MeshPrimitive, [498](#)
- VIEWType_END
 - gdcm::Surface, [690](#)
- VL16
 - gdcm::VR, [827](#)
- VL32
 - gdcm::VR, [827](#)
- VLEndoscopicImageStorage
 - gdcm::MediaStorage, [490](#)
 - gdcm::UIDs, [746](#)
- VLImageStorageTrialRetired
 - gdcm::UIDs, [746](#)
- VLMicroscopicImageStorage
 - gdcm::UIDs, [746](#)
- VLMultiframeImageStorageTrialRetired
 - gdcm::UIDs, [746](#)
- VLPhotographicImageStorage
 - gdcm::MediaStorage, [490](#)
 - gdcm::UIDs, [746](#)
- VLSlideCoordinatesMicroscopicImageStorage
 - gdcm::UIDs, [746](#)
- VLWholeSlideMicroscopyImageStorage
 - gdcm::MediaStorage, [490](#)
 - gdcm::UIDs, [749](#)
- VM0
 - gdcm::VM, [822](#)
- VM1
 - gdcm::VM, [822](#)
- VM10
 - gdcm::VM, [822](#)
- VM12

- gdcM::VM, [822](#)
- VM16
 - gdcM::VM, [822](#)
- VM18
 - gdcM::VM, [822](#)
- VM1_2
 - gdcM::VM, [823](#)
- VM1_3
 - gdcM::VM, [823](#)
- VM1_32
 - gdcM::VM, [823](#)
- VM1_4
 - gdcM::VM, [823](#)
- VM1_5
 - gdcM::VM, [823](#)
- VM1_8
 - gdcM::VM, [823](#)
- VM1_99
 - gdcM::VM, [823](#)
- VM1_n
 - gdcM::VM, [823](#)
- VM2
 - gdcM::VM, [822](#)
- VM24
 - gdcM::VM, [822](#)
- VM256
 - gdcM::VM, [823](#)
- VM28
 - gdcM::VM, [822](#)
- VM2_2n
 - gdcM::VM, [823](#)
- VM2_n
 - gdcM::VM, [823](#)
- VM3
 - gdcM::VM, [822](#)
- VM30_30n
 - gdcM::VM, [823](#)
- VM32
 - gdcM::VM, [822](#)
- VM35
 - gdcM::VM, [822](#)
- VM3_3n
 - gdcM::VM, [823](#)
- VM3_4
 - gdcM::VM, [823](#)
- VM3_n
 - gdcM::VM, [823](#)
- VM4
 - gdcM::VM, [822](#)
- VM47_47n
 - gdcM::VM, [823](#)
- VM4_4n
 - gdcM::VM, [823](#)
- VM5
 - gdcM::VM, [822](#)
- VM6
 - gdcM::VM, [822](#)
- VM6_6n
 - gdcM::VM, [823](#)
- VM7_7n
 - gdcM::VM, [823](#)
- VM8
 - gdcM::VM, [822](#)
- VM9
 - gdcM::VM, [822](#)
- VM99
 - gdcM::VM, [823](#)
- VM_END
 - gdcM::VM, [823](#)
- VMType
 - gdcM::Attribute, [165](#)
 - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, [172](#)
- VOILUTBoxSOPClass
 - gdcM::UIDs, [745](#)
- VR_END
 - gdcM::VR, [827](#)
- VR_VM1
 - gdcM::VR, [827](#)
- VRALL
 - gdcM::VR, [827](#)
- VRASCII
 - gdcM::VR, [827](#)
- VRBINARY
 - gdcM::VR, [827](#)
- VT100
 - gdcM::terminal, [131](#)
- VIEWType
 - gdcM::Surface, [690](#)
- VL
 - gdcM::VL, [819](#)
- VM
 - gdcM::VM, [823](#)
- VMType
 - gdcM::VM, [822](#)
- VR
 - gdcM::VR, [827](#)
- VRBINARY
 - gdcM, [124](#)
- VRField
 - gdcM::CSAElement, [258](#)
 - gdcM::DataElement, [281](#)
- VRType
 - gdcM::VR, [826](#)
- VRTypeTemplateCase
 - gdcMVR.h, [1150](#)
- VTK_CMYK
 - vtkGDCMImageReader.h, [1156](#)

- VTK_LEGACY
 - vtkImageColorViewer, [865](#)
- VTK_LOOKUP_TABLE
 - vtkGDCMImageReader.h, [1156](#)
- VTK_YBR
 - vtkGDCMImageReader.h, [1156](#)
- Valid
 - gdcm::Preamble, [565](#)
- Validate
 - gdcm::PixelFormat, [549](#)
 - gdcm::Validate, [814](#)
- ValidateQuery
 - gdcm::BaseRootQuery, [200](#)
 - gdcm::FindPatientRootQuery, [382](#)
 - gdcm::FindStudyRootQuery, [384](#)
 - gdcm::MovePatientRootQuery, [508](#)
 - gdcm::MoveStudyRootQuery, [511](#)
- Validation
 - gdcm::Validate, [814](#)
- Value
 - gdcm::Value, [815](#)
- value
 - gdcm::SerieHelper::Rule, [614](#)
 - gdcm::STATIC_ASSERTION_FAILURE< true >, [670](#)
- value_type
 - gdcm::CodeString, [241](#)
 - gdcm::LO, [475](#)
 - gdcm::String, [681](#)
- ValueField
 - gdcm::DataElement, [281](#)
 - gdcm::PDBelement, [533](#)
- ValueLengthField
 - gdcm::DataElement, [281](#)
- ValueMultiplicityField
 - gdcm::CSAElement, [258](#)
- ValuePtr
 - gdcm::DataElement, [275](#)
- ValueType
 - gdcm::Scanner, [618](#)
- VerificationSOPClass
 - gdcm::UIDs, [742](#)
- Verify
 - gdcm::Defs, [297](#), [298](#)
 - gdcm::Macro, [482](#)
 - gdcm::Module, [502](#)
- Version
 - gdcm::Version, [817](#)
- Video
 - gdcm::MediaStorage, [490](#)
- VideoEndoscopicImageStorage
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [746](#)
- VideoMicroscopicImageStorage
 - gdcm::UIDs, [746](#)
- VideoPhotographicImageStorage
 - gdcm::UIDs, [746](#)
- vtkGDCMImageWriter
 - JPEG2000_COMPRESSION, [841](#)
 - JPEG_COMPRESSION, [841](#)
 - JPEGLS_COMPRESSION, [841](#)
 - NO_COMPRESSION, [841](#)
 - RLE_COMPRESSION, [841](#)
- vtkImageColorViewer
 - SLICE_ORIENTATION_XY, [862](#)
 - SLICE_ORIENTATION_XZ, [862](#)
 - SLICE_ORIENTATION_YZ, [862](#)
- vtkBooleanMacro
 - vtkGDCMImageReader, [836](#), [837](#)
 - vtkGDCMImageWriter, [842](#)
 - vtkGDCMThreadedImageReader, [855](#)
 - vtkGDCMThreadedImageReader2, [858](#)
 - vtkImageColorViewer, [865](#)
 - vtkImageMapToColors16, [868](#)
- vtkGDCMImageReader, [833](#)
 - ~vtkGDCMImageReader, [835](#)
 - ApplyInverseVideo, [838](#)
 - ApplyLookupTable, [838](#)
 - ApplyPlanarConfiguration, [838](#)
 - ApplyShiftScale, [838](#)
 - ApplyYBRToRGB, [838](#)
 - CanReadFile, [835](#)
 - Curve, [838](#)
 - DirectionCosines, [838](#)
 - ExecuteData, [835](#)
 - ExecuteInformation, [835](#)
 - FileNames, [838](#)
 - FillMedicalImageInformation, [835](#)
 - ForceRescale, [838](#)
 - GetDescriptiveName, [836](#)
 - GetFileExtensions, [836](#)
 - GetIconImage, [836](#)
 - GetOverlay, [836](#)
 - IconDataScalarType, [838](#)
 - IconImageDataExtent, [838](#)
 - IconNumberOfScalarComponents, [838](#)
 - ImageFormat, [838](#)
 - ImageOrientationPatient, [838](#)
 - ImagePositionPatient, [838](#)
 - LoadIconImage, [838](#)
 - LoadOverlays, [838](#)
 - LoadSingleFile, [836](#)
 - LossyFlag, [838](#)
 - MedicalImageProperties, [838](#)
 - New, [836](#)
 - NumberOfIconImages, [838](#)
 - NumberOfOverlays, [838](#)
 - PlanarConfiguration, [838](#)

- PrintSelf, [836](#)
- RequestDataCompat, [836](#)
- RequestInformationCompat, [836](#)
- Scale, [838](#)
- SetCurve, [836](#)
- SetFileNames, [836](#)
- SetFilePattern, [836](#)
- SetFilePrefix, [836](#)
- SetMedicalImageProperties, [836](#)
- Shift, [838](#)
- vtkBooleanMacro, [836](#), [837](#)
- vtkGDCMImageReader, [835](#)
- vtkGetMacro, [837](#)
- vtkGetObjectMacro, [837](#)
- vtkGetStringMacro, [837](#)
- vtkGetVector3Macro, [837](#)
- vtkGetVector6Macro, [837](#)
- vtkSetMacro, [837](#)
- vtkSetVector6Macro, [837](#)
- vtkTypeRevisionMacro, [837](#)
- vtkGDCMImageReader, [835](#)
- vtkGDCMMedicalImageProperties, [845](#)
- vtkGDCMImageReader.h, [1155](#)
 - VTK_CMYK, [1156](#)
 - VTK_YBR, [1156](#)
- vtkGDCMImageWriter, [839](#)
 - ~vtkGDCMImageWriter, [841](#)
 - CompressionTypes, [841](#)
 - GetDescriptiveName, [841](#)
 - GetFileExtensions, [841](#)
 - GetFileName, [841](#)
 - New, [841](#)
 - PrintSelf, [841](#)
 - SetDirectionCosines, [841](#)
 - SetDirectionCosinesFromImageOrientationPatient, [841](#)
 - SetFileNames, [841](#)
 - SetMedicalImageProperties, [841](#)
 - vtkBooleanMacro, [842](#)
 - vtkGDCMImageWriter, [841](#)
 - vtkGetMacro, [842](#)
 - vtkGetObjectMacro, [842](#)
 - vtkGetStringMacro, [842](#)
 - vtkSetMacro, [842](#)
 - vtkSetStringMacro, [842](#)
 - vtkTypeRevisionMacro, [843](#)
 - vtkGDCMImageWriter, [841](#)
 - vtkGDCMMedicalImageProperties, [845](#)
 - Write, [843](#)
 - WriteGDCMData, [843](#)
 - WriteSlice, [843](#)
- vtkGDCMImageWriter.h, [1156](#)
- vtkGDCMMedicalImageProperties, [843](#)
 - ~vtkGDCMMedicalImageProperties, [844](#)
- Clear, [844](#)
- GetFile, [845](#)
- New, [845](#)
- PrintSelf, [845](#)
- PushBackFile, [845](#)
- vtkGDCMImageReader, [845](#)
- vtkGDCMImageWriter, [845](#)
- vtkGDCMMedicalImageProperties, [844](#)
- vtkTypeRevisionMacro, [845](#)
- vtkGDCMMedicalImageProperties, [844](#)
- vtkGDCMMedicalImageProperties.h, [1156](#)
- vtkGDCMPolyDataReader, [845](#)
 - ~vtkGDCMPolyDataReader, [847](#)
 - FileName, [848](#)
 - FillMedicalImageInformation, [847](#)
 - MedicalImageProperties, [848](#)
 - New, [847](#)
 - PrintSelf, [847](#)
 - RTStructSetProperties, [848](#)
 - RequestData, [847](#)
 - RequestData_HemodynamicWaveformStorage, [847](#)
 - RequestData_RTStructureSetStorage, [847](#)
 - RequestInformation, [847](#)
 - RequestInformation_HemodynamicWaveformStorage, [847](#)
 - RequestInformation_RTStructureSetStorage, [847](#)
- vtkGDCMPolyDataReader, [847](#)
- vtkGetObjectMacro, [847](#)
- vtkGetStringMacro, [847](#)
- vtkSetStringMacro, [848](#)
- vtkTypeRevisionMacro, [848](#)
- vtkGDCMPolyDataReader, [847](#)
- vtkGDCMPolyDataReader.h, [1157](#)
- vtkGDCMPolyDataWriter, [848](#)
 - ~vtkGDCMPolyDataWriter, [850](#)
 - InitializeRTStructSet, [850](#)
 - MedicalImageProperties, [851](#)
 - New, [850](#)
 - PrintSelf, [850](#)
 - RTStructSetProperties, [851](#)
 - SetMedicalImageProperties, [850](#)
 - SetNumberOfInputPorts, [850](#)
 - SetRTStructSetProperties, [850](#)
 - vtkGDCMPolyDataWriter, [850](#)
 - vtkTypeRevisionMacro, [850](#)
 - vtkGDCMPolyDataWriter, [850](#)
 - WriteData, [851](#)
 - WriteRTSTRUCTData, [851](#)
 - WriteRTSTRUCTInfo, [851](#)
- vtkGDCMPolyDataWriter.h, [1158](#)
- vtkGDCMTesting, [851](#)
 - ~vtkGDCMTesting, [853](#)
 - GetGDCMDataRoot, [853](#)
 - GetMD5MetaImage, [853](#)

- GetMHDMD5FromFile, [853](#)
- GetNumberOfMD5MetalImages, [853](#)
- GetRAWMD5FromFile, [853](#)
- GetVTKDataRoot, [853](#)
- MD5MetalImagesType, [853](#)
- New, [853](#)
- PrintSelf, [853](#)
- vtkGDCMTesting, [853](#)
- vtkTypeRevisionMacro, [853](#)
- vtkGDCMTesting, [853](#)
- vtkGDCMTesting.h, [1158](#)
- vtkGDCMThreadedImageReader, [853](#)
 - ~vtkGDCMThreadedImageReader, [855](#)
 - ExecuteData, [855](#)
 - ExecuteInformation, [855](#)
 - New, [855](#)
 - PrintSelf, [855](#)
 - ReadFiles, [855](#)
 - RequestDataCompat, [855](#)
 - vtkBooleanMacro, [855](#)
 - vtkGDCMThreadedImageReader, [855](#)
 - vtkGetMacro, [855](#)
 - vtkSetMacro, [855](#)
 - vtkTypeRevisionMacro, [855](#)
 - vtkGDCMThreadedImageReader, [855](#)
- vtkGDCMThreadedImageReader.h, [1159](#)
- vtkGDCMThreadedImageReader2, [856](#)
 - ~vtkGDCMThreadedImageReader2, [857](#)
 - GetFileName, [857](#)
 - New, [857](#)
 - PrintSelf, [857](#)
 - RequestInformation, [858](#)
 - SetFileName, [858](#)
 - SetFileNames, [858](#)
 - SplitExtent, [858](#)
 - ThreadedRequestData, [858](#)
 - vtkBooleanMacro, [858](#)
 - vtkGDCMThreadedImageReader2, [857](#)
 - vtkGetMacro, [858](#)
 - vtkGetObjectMacro, [858](#)
 - vtkGetVector3Macro, [858](#)
 - vtkGetVector6Macro, [858](#)
 - vtkSetMacro, [858](#), [859](#)
 - vtkSetVector3Macro, [859](#)
 - vtkSetVector6Macro, [859](#)
 - vtkTypeRevisionMacro, [859](#)
 - vtkGDCMThreadedImageReader2, [857](#)
- vtkGDCMThreadedImageReader2.h, [1160](#)
- vtkGetMacro
 - vtkGDCMImageReader, [837](#)
 - vtkGDCMImageWriter, [842](#)
 - vtkGDCMThreadedImageReader, [855](#)
 - vtkGDCMThreadedImageReader2, [858](#)
 - vtkImageColorViewer, [865](#)
- vtkImageMapToColors16, [868](#)
- vtkImageMapToWindowLevelColors2, [870](#)
- vtkGetObjectMacro
 - vtkGDCMImageReader, [837](#)
 - vtkGDCMImageWriter, [842](#)
 - vtkGDCMPolyDataReader, [847](#)
 - vtkGDCMThreadedImageReader2, [858](#)
 - vtkImageColorViewer, [865](#)
 - vtkImageMapToColors16, [868](#)
- vtkGetStringMacro
 - vtkGDCMImageReader, [837](#)
 - vtkGDCMImageWriter, [842](#)
 - vtkGDCMPolyDataReader, [847](#)
 - vtkRTStructSetProperties, [882](#)
- vtkGetVector3Macro
 - vtkGDCMImageReader, [837](#)
 - vtkGDCMThreadedImageReader2, [858](#)
- vtkGetVector6Macro
 - vtkGDCMImageReader, [837](#)
 - vtkGDCMThreadedImageReader2, [858](#)
- vtkImageColorViewer, [859](#)
 - ~vtkImageColorViewer, [862](#)
 - AddInput, [862](#)
 - AddInputConnection, [862](#)
 - FirstRender, [865](#)
 - GetColorLevel, [862](#)
 - GetColorWindow, [862](#)
 - GetInput, [862](#)
 - GetOffScreenRendering, [862](#)
 - GetOverlayVisibility, [862](#)
 - GetPosition, [862](#)
 - GetSize, [863](#)
 - GetSliceMax, [863](#)
 - GetSliceMin, [863](#)
 - GetSliceRange, [863](#)
 - GetWindowName, [863](#)
 - ImageActor, [865](#)
 - InstallPipeline, [863](#)
 - Interactor, [865](#)
 - InteractorStyle, [865](#)
 - New, [863](#)
 - OverlayImageActor, [865](#)
 - PrintSelf, [863](#)
 - Render, [863](#)
 - RenderWindow, [865](#)
 - Renderer, [865](#)
 - SetColorLevel, [863](#)
 - SetColorWindow, [863](#)
 - SetDisplayId, [863](#)
 - SetInput, [863](#)
 - SetInputConnection, [863](#)
 - SetOffScreenRendering, [863](#)
 - SetOverlayVisibility, [863](#)
 - SetParentId, [863](#)

- SetPosition, [864](#)
- SetRenderWindow, [864](#)
- SetRenderer, [864](#)
- SetSize, [864](#)
- SetSlice, [864](#)
- SetSliceOrientation, [864](#)
- SetSliceOrientationToXY, [864](#)
- SetSliceOrientationToXZ, [864](#)
- SetSliceOrientationToYZ, [864](#)
- SetWindowId, [864](#)
- SetupInteractor, [864](#)
- Slice, [865](#)
- SliceOrientation, [865](#)
- UnInstallPipeline, [865](#)
- UpdateDisplayExtent, [865](#)
- UpdateOrientation, [865](#)
- VTK_LEGACY, [865](#)
- vtkBooleanMacro, [865](#)
- vtkGetMacro, [865](#)
- vtkGetObjectMacro, [865](#)
- vtkImageColorViewer, [862](#)
- vtkTypeRevisionMacro, [865](#)
- vtkImageColorViewer, [862](#)
- WindowLevel, [866](#)
- vtkImageColorViewer.h, [1160](#)
- vtkImageMapToColors16, [866](#)
 - ~vtkImageMapToColors16, [867](#)
 - ActiveComponent, [868](#)
 - DataWasPassed, [868](#)
 - GetMTime, [867](#)
 - LookupTable, [868](#)
 - New, [867](#)
 - OutputFormat, [868](#)
 - PassAlphaToOutput, [868](#)
 - PrintSelf, [867](#)
 - RequestData, [868](#)
 - RequestInformation, [868](#)
 - SetLookupTable, [868](#)
 - SetOutputFormatToLuminance, [868](#)
 - SetOutputFormatToLuminanceAlpha, [868](#)
 - SetOutputFormatToRGB, [868](#)
 - SetOutputFormatToRGBA, [868](#)
 - ThreadedRequestData, [868](#)
 - vtkBooleanMacro, [868](#)
 - vtkGetMacro, [868](#)
 - vtkGetObjectMacro, [868](#)
 - vtkImageMapToColors16, [867](#)
 - vtkSetMacro, [868](#)
 - vtkTypeRevisionMacro, [868](#)
 - vtkImageMapToColors16, [867](#)
- vtkImageMapToColors16.h, [1161](#)
- vtkImageMapToWindowLevelColors2, [869](#)
 - ~vtkImageMapToWindowLevelColors2, [870](#)
 - Level, [871](#)
 - New, [870](#)
 - PrintSelf, [870](#)
 - RequestData, [870](#)
 - RequestInformation, [870](#)
 - ThreadedRequestData, [870](#)
 - vtkGetMacro, [870](#)
 - vtkImageMapToWindowLevelColors2, [870](#)
 - vtkSetMacro, [870](#), [871](#)
 - vtkTypeRevisionMacro, [871](#)
 - vtkImageMapToWindowLevelColors2, [870](#)
 - Window, [871](#)
- vtkImageMapToWindowLevelColors2.h, [1161](#)
- vtkImagePlanarComponentsToComponents, [871](#)
 - ~vtkImagePlanarComponentsToComponents, [872](#)
 - New, [872](#)
 - PrintSelf, [872](#)
 - RequestData, [872](#)
 - vtkImagePlanarComponentsToComponents, [872](#)
 - vtkTypeRevisionMacro, [873](#)
 - vtkImagePlanarComponentsToComponents, [872](#)
- vtkImagePlanarComponentsToComponents.h, [1162](#)
- vtkImageRGBToYBR, [873](#)
 - ~vtkImageRGBToYBR, [874](#)
 - New, [874](#)
 - PrintSelf, [874](#)
 - ThreadedExecute, [874](#)
 - vtkImageRGBToYBR, [874](#)
 - vtkTypeRevisionMacro, [874](#)
 - vtkImageRGBToYBR, [874](#)
- vtkImageRGBToYBR.h, [1162](#)
- vtkImageYBRToRGB, [874](#)
 - ~vtkImageYBRToRGB, [876](#)
 - New, [876](#)
 - PrintSelf, [876](#)
 - ThreadedExecute, [876](#)
 - vtkImageYBRToRGB, [876](#)
 - vtkTypeRevisionMacro, [876](#)
 - vtkImageYBRToRGB, [876](#)
- vtkImageYBRToRGB.h, [1163](#)
- vtkLookupTable16, [876](#)
 - ~vtkLookupTable16, [877](#)
 - Build, [877](#)
 - GetPointer, [878](#)
 - MapScalarsThroughTable2, [878](#)
 - New, [878](#)
 - PrintSelf, [878](#)
 - SetNumberOfTableValues, [878](#)
 - Table16, [878](#)
 - vtkLookupTable16, [877](#)
 - vtkTypeRevisionMacro, [878](#)
 - vtkLookupTable16, [877](#)
 - WritePointer, [878](#)
- vtkLookupTable16.h, [1163](#)
- vtkRTStructSetProperties, [878](#)

- ~vtkRTStructSetProperties, 880
- AddContourReferencedFrameOfReference, 880
- AddReferencedFrameOfReference, 881
- AddStructureSetROI, 881
- AddStructureSetROIObservation, 881
- Clear, 881
- DeepCopy, 881
- GetContourReferencedFrameOfReferenceClassUID, 881
- GetContourReferencedFrameOfReferenceInstanceUID, 881
- GetNumberOfContourReferencedFrameOfReferences, 881
- GetNumberOfReferencedFrameOfReferences, 881
- GetNumberOfStructureSetROIs, 881
- GetReferencedFrameOfReferenceClassUID, 881
- GetReferencedFrameOfReferenceInstanceUID, 881
- GetStructureSetObservationNumber, 881
- GetStructureSetROIDescription, 881
- GetStructureSetROIGenerationAlgorithm, 881
- GetStructureSetROIName, 881
- GetStructureSetROINumber, 881
- GetStructureSetROIObservationLabel, 881
- GetStructureSetROIRefFrameRefUID, 881
- GetStructureSetRTROIInterpretedType, 881
- Internals, 882
- New, 881
- PrintSelf, 882
- ReferenceFrameOfReferenceUID, 882
- ReferenceSeriesInstanceUID, 882
- SOPInstanceUID, 883
- SeriesInstanceUID, 883
- StructureSetDate, 883
- StructureSetLabel, 883
- StructureSetName, 883
- StructureSetTime, 883
- StudyInstanceUID, 883
- vtkGetStringMacro, 882
- vtkRTStructSetProperties, 880
- vtkSetStringMacro, 882
- vtkTypeRevisionMacro, 882
- vtkRTStructSetProperties, 880
- vtkRTStructSetProperties.h, 1164
- vtkSetMacro
 - vtkGDCMImageReader, 837
 - vtkGDCMImageWriter, 842
 - vtkGDCMThreadedImageReader, 855
 - vtkGDCMThreadedImageReader2, 858, 859
 - vtkImageMapToColors16, 868
 - vtkImageMapToWindowLevelColors2, 870, 871
- vtkSetStringMacro
 - vtkGDCMImageWriter, 842
 - vtkGDCMPolyDataReader, 848
 - vtkRTStructSetProperties, 882
- vtkSetVector3Macro
 - vtkGDCMThreadedImageReader2, 859
- vtkSetVector6Macro
 - vtkGDCMImageReader, 837
 - vtkGDCMThreadedImageReader2, 859
- vtkTypeRevisionMacro
 - vtkGDCMImageReader, 837
 - vtkGDCMImageWriter, 843
 - vtkGDCMMedicalImageProperties, 845
 - vtkGDCMPolyDataReader, 848
 - vtkGDCMPolyDataWriter, 850
 - vtkGDCMTesting, 853
 - vtkGDCMThreadedImageReader, 855
 - vtkGDCMThreadedImageReader2, 859
 - vtkImageColorViewer, 865
 - vtkImageMapToColors16, 868
 - vtkImageMapToWindowLevelColors2, 871
 - vtkImagePlanarComponentsToComponents, 873
 - vtkImageRGBToYBR, 874
 - vtkImageYBRToRGB, 876
 - vtkLookupTable16, 878
 - vtkRTStructSetProperties, 882
- WIREFRAME
 - gdcm::Surface, 690
- WarningOff
 - gdcm::Trace, 726
- WarningOn
 - gdcm::Trace, 727
- Waveform
 - gdcm::Waveform, 883
 - gdcm::MediaStorage, 490
- WaveformStorageTrialRetired
 - gdcm::UIDs, 745
- what
 - gdcm::Exception, 351
- white
 - gdcm::terminal, 131
- Window
 - vtkImageMapToWindowLevelColors2, 871
- WindowLevel
 - vtkImageColorViewer, 866
- Write
 - gdcm::ByteValue, 225
 - gdcm::CommandDataSet, 245
 - gdcm::CSAHeader, 262
 - gdcm::DataElement, 280
 - gdcm::DataSet, 291
 - gdcm::Element, 328
 - gdcm::Element< TVR, VM::VM1_n >, 332
 - gdcm::EncodingImplementation< VR::VRASCII >, 345
 - gdcm::EncodingImplementation< VR::VRBINARY >, 346

- gdcm::ExplicitDataElement, [354](#)
- gdcm::File, [360](#)
- gdcm::FileAnonymizer, [363](#)
- gdcm::FileMetaInformation, [372](#)
- gdcm::Fragment, [386](#)
- gdcm::ImageWriter, [435](#)
- gdcm::ImplicitDataElement, [440](#)
- gdcm::Item, [453](#)
- gdcm::network::AAAbortPDU, [135](#)
- gdcm::network::AAssociateACPDU, [137](#)
- gdcm::network::AAssociateRJPDU, [139](#)
- gdcm::network::AAssociateRQPDU, [143](#)
- gdcm::network::AbstractSyntax, [145](#)
- gdcm::network::ApplicationContext, [155](#)
- gdcm::network::AReleaseRPPDU, [158](#)
- gdcm::network::AReleaseRQPDU, [160](#)
- gdcm::network::AsynchronousOperationsWindow-Sub, [163](#)
- gdcm::network::BasePDU, [196](#)
- gdcm::network::ImplementationClassUIDSub, [436](#)
- gdcm::network::ImplementationUIDSub, [437](#)
- gdcm::network::ImplementationVersionNameSub, [437](#)
- gdcm::network::MaximumLengthSub, [484](#)
- gdcm::network::PDataTFPDU, [531](#)
- gdcm::network::PresentationContextAC, [568](#)
- gdcm::network::PresentationContextRQ, [572](#)
- gdcm::network::PresentationDataValue, [574](#)
- gdcm::network::RoleSelectionSub, [613](#)
- gdcm::network::ServiceClassApplicationInformation, [646](#)
- gdcm::network::SOPClassExtendedNegociationSub, [660](#)
- gdcm::network::TransferSyntaxSub, [732](#)
- gdcm::network::UserInformation, [813](#)
- gdcm::PGXCodec, [542](#)
- gdcm::PixmapWriter, [561](#)
- gdcm::PNMCodec, [563](#)
- gdcm::Preamble, [565](#)
- gdcm::SegmentWriter, [631](#)
- gdcm::SequenceOfFragments, [636](#)
- gdcm::SequenceOfItems, [641](#)
- gdcm::StreamImageWriter, [677](#)
- gdcm::SurfaceWriter, [700](#)
- gdcm::Tag, [718](#)
- gdcm::ValueIO, [816](#)
- gdcm::VL, [820](#)
- gdcm::VR, [828](#)
- gdcm::VRVLSize< 0 >, [832](#)
- gdcm::VRVLSize< 1 >, [832](#)
- gdcm::Writer, [888](#)
- vtkGDCMImageWriter, [843](#)
- Write16
 - gdcm::VL, [820](#)
- WriteASCII
 - gdcm::Element< TVR, VM::VM1_n >, [332](#)
- WriteBuffer
 - gdcm::ByteValue, [225](#)
 - gdcm::SequenceOfFragments, [636](#)
- WriteBufferAsRGBA
 - gdcm::LookupTable, [480](#)
- WriteData
 - vtkGDCMPolyDataWriter, [851](#)
- WriteFooter
 - gdcm::DictConverter, [307](#)
- WriteGDCMData
 - vtkGDCMImageWriter, [843](#)
- WriteHeader
 - gdcm::DictConverter, [307](#)
- WriteHelpFile
 - gdcm::BaseRootQuery, [200](#)
- WriteImageInformation
 - gdcm::StreamImageWriter, [677](#)
- WriteImageSubregionRAW
 - gdcm::StreamImageWriter, [677](#)
- WritePointer
 - vtkLookupTable16, [878](#)
- WriteQuery
 - gdcm::BaseRootQuery, [200](#)
- WriteRTSTRUCTData
 - vtkGDCMPolyDataWriter, [851](#)
- WriteRTSTRUCTInfo
 - vtkGDCMPolyDataWriter, [851](#)
- WriteRawHeader
 - gdcm::StreamImageWriter, [677](#)
- WriteSlice
 - vtkGDCMImageWriter, [843](#)
- Writer
 - gdcm::Writer, [887](#)
- XML
 - gdcm::Printer, [576](#)
- XMLEncoding
 - gdcm::UIDs, [743](#)
- XRay3DAngiographicImageStorage
 - gdcm::MediaStorage, [490](#)
 - gdcm::UIDs, [746](#)
- XRay3DCraniofacialImageStorage
 - gdcm::UIDs, [746](#)
- XRayAngiographicBiPlaneImageStorageRetired
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [746](#)
- XRayAngiographicImageStorage
 - gdcm::MediaStorage, [489](#)
 - gdcm::UIDs, [746](#)
- XRayRadiationDoseSR
 - gdcm::MediaStorage, [490](#)
- XRayRadiationDoseSRStorage

- gdcM::UIDs, [747](#)
- XRayRadiofluoroscopicImageStorage
 - gdcM::UIDs, [746](#)
- XRayRadiofluoroscopicImageStorage
 - gdcM::MediaStorage, [489](#)
- XMLDictReader
 - gdcM::XMLDictReader, [890](#)
- XMLPrivateDictReader
 - gdcM::XMLPrivateDictReader, [892](#)
- YBR_FULL
 - gdcM::PhotometricInterpretation, [543](#)
- YBR_FULL_422
 - gdcM::PhotometricInterpretation, [543](#)
- YBR_ICT
 - gdcM::PhotometricInterpretation, [543](#)
- YBR_PARTIAL_420
 - gdcM::PhotometricInterpretation, [543](#)
- YBR_PARTIAL_422
 - gdcM::PhotometricInterpretation, [543](#)
- YBR_RCT
 - gdcM::PhotometricInterpretation, [543](#)
- YES
 - gdcM::Surface, [690](#)
- YBR2RGB
 - gdcM::ImageChangePhotometricInterpretation, [405](#)
- yellow
 - gdcM::terminal, [131](#)
- ZEROED_OUT
 - gdcM::CSAHeader, [260](#)
- ZSpacing
 - gdcM::IPPSorter, [450](#)
- ZTolerance
 - gdcM::IPPSorter, [450](#)